How stories evolve: News graphs as a way to represent context and evolution of news stories

ABSTRACT

Providing the history and context(s) of a news article that emerges in the middle of an evolving news story–sometimes multiple news stories–is a complex task. The complexity of the task is compounded by the fact that different users are interested in different contexts of the article, and it is impossible to guess what a particular user is most interested in. In this paper, we introduce ESTHETE, a system that provides rich context(s) (through what we call *personalized flexible context extraction*), by preprocessing and storing articles in a structured representation (directed graphs) that makes it easy for the user to explore different contexts. We formally define what constitutes a "good" news story and give an algorithm to efficiently compute coherent and contextfull news stories from the news graph. The advantage of this approach is that the incremental computational expense in incorporating new articles as they are published is minimal. We describe the design and features of ESTHETE, and present the results of a comprehensive user study highlighting the usefulness of our system. Our system is available at: `http://konfrap.com/esthete`.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

News Corpus Visualization, Flexible Personalized Context-rich News graph, News Summarization, System for News browsing, Interface Design

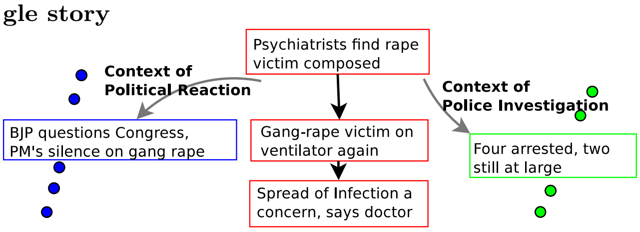
1. INTRODUCTION

Online news websites are now a common way of consuming news. These websites are very helpful in publishing breaking news and help users keep up to date with the latest developments. However, the proliferation of these websites simply means that we are drowning in more news than we know what to do with. Specific news stories that we would like to follow over days or weeks may be buried among more current or recent news. Navigating through a maze of articles looking for ones of interest is a non-trivial task.

Current news websites do not do much to help users trace the origin of these stories, nor how they overlap with each other. Most of them still "look" like news papers with a bunch of articles laid out on the web-page, with a few additional features in the form of "Related News" and "Recommended News" links. A reader trying to contextualize a particular article can follow some of these links but the context provided by these features is shallow and tracing the development of the story with its various aspects is not much easier than it was in the days of paper-based news media. The news corpus is available in electronic form, carefully archived by all news organizations. There have been some recent efforts by the research community that attempt to extract individual strands of a stories development from the corpus [10] or present interacting strands in a mesh-like structure [11]. Still others use graph-like abstractions, just like we do, to approach the limited problem of detecting a news item that represents the development of a news story [12, 13]. What remains missing is the structural framework required to process it into a form that would make it possible for a user to elaborate the many contexts behind a single news story, some stretching years into the past, some just a few days.

To make this concrete let us consider the unfortunate gang-rape incident that took place in Delhi in December 2012. Since the day this crime was first reported by the media, it grew into a complex news story, talking about related rape cases of the past, the reactions of various sections of the society (politicians, activists, human rights organizations, etc), the health of the victim up to her tragic death, the investigation carried out by the law enforcement authorities and the judicial proceedings against the offenders once they were caught. Different users may want to focus on different aspects of this story. For eg., consider a graph of news articles around this story shown in Figure 1. We see three of the many possible contexts that can be followed and utilized by a user to better understand the overall picture. We allow the user to express her intent to follow a particular context (either "the reactions of Politicians" or "the victim's condition" or "the Police investigations") and understand the story with that context. This results in a view that is personalized for a particular user, since she is free to choose the context(s) that help her follow the story. We call this notion *Personalized Flexible Context Extraction.* This notion is an extremely natural approach that a news consumer, whether an expert commentator or a lay person, might want to employ as a crucial subtask in the process of news consumption. We believe that it is possible to realize this notion given the development in information extraction techniques that has taken place in the last few years. However this realization requires the specification and implementation of a framework for (pre)processing and structuring the news corpus in such a way that it can easily serve information needs that cannot always be forseen when the news emerges.

Figure 1: How multiple contexts are added to a single story



Our attempt in this paper is to provide such a framework and show that it can be used in a flexible way by a user to fulfill his or her need to contextualize what he or she is reading in the news. Building on the work of [4] that models news corpora as a time-evolving graph by linking related news articles, we provide a query mechanism that organizes this structure by topics of interest and important actors, and presents the development of a story on a timeline. The user can now interact with the system by filtering articles based on topics, actors, time, or a combination of all three by iteratively querying on these fields. The process of querying generates different views (subgraphs) of the main news graph back-end, avoiding the need of creating a new graph structure for every query. For example, Figure 7 shows the timeline of the recent rape case incidents reported from India.

An important aspect of our work that represents a completely different approach from the prominent attempts at news corpus processing and extraction in the literature is that we take an offline approach i.e. we view the news corpus as an evolving entity that is streaming past us. We store it in a structured way such that it can be browsed and, importantly, *different kinds of contexts can be extracted from it by users with different approaches to the same news stories.* This represents a break in thinking from those approaches that take the entire news corpus as their input and either produce an output which *is* the story [10, 11] or produce some output based on a pre-existing notion of what is the story [12, 13]. We allow the user to build the story. Apart from providing lower flexibility to the user, in our opinion, these other approaches are fundamentally not-scalable since every time an information need arises a set of expensive computations has to be done. Our approach can be thought of as a data-structuring approach: We preprocess the data to make news browsing and user-led context discovery an easier and computationally feasible process.

In summary, our contributions are as follows:

- We build on a graph-based framework ([4]) to represent articles in a news corpus and make it query-able.
- We propose metrics to capture utility of a news article to understand a news story in terms of the coherent context provided by it.
- We propose an efficient graph-based algorithm to find useful contexts around a news story that help to get the bigger picture.
- We provide an interactive interface to users which is easy to use, and helps in tracking the evolution of the story and zooming to specific parts of interest.
- We report on the results of a thorough evaluation of our system and show that our system compares well with existing ones.

Organization: The rest of the paper is organized as follows. We first summarize the related work that has addressed the problem of news corpus mining, categorizing it under various threads (Section 2). We then formally state our definition of a personalized flexible context extracting news browsing tool, describing the basic features of such a tool and our approach to building such a tool based on the news graphs of [4] (Section 3). In particular, we introduce the notion and importance of *context* of this news graph, and describe how these contexts can be extracted. Next, we give a detailed block-by-block description of our system (Section 4). We then present results from various evaluation experiments conducted on our news browsing system (Section 5). Finally, we conclude with some observations and future scope (Section 6).

2. PAST WORK

Our works builds upon several areas of research, in particular the graphical representation of news articles, the identification and tracking of topics, time-based story and event evolution, multi document summarization, and information visualisation.

Graph representation of Articles. Choudhary et al.[4] proposed (which was at that time) a novel method of organizing a news corpus into a directed graph, by mining and tracking the *transformations* that the *actors* in the corpus undergo. These transformations, defined for one or two actors, can be one of the following:

- **Birth:** An actor appearing for the first time
- **Cease:** An actor appearing for the last time
- **Merge:** Two actors interacting for the first time
- **Split:** Two actors interacting for the last time
- **Continue:** An actor remaining popular in a period

Their idea can be summarized as follows: 1) Mine transformations in the article set 2) Find a weighted-edge covering of all transformations. For a more detailed description, we refer the reader to the paper. The work does not discuss how such a news graph may be mined to extract useful information, rather proposes this graph as a means of visualizing the news. However, such graphs easily blow up as the complexity of the news story increases.

prehensive summary from an input sequence of articles.

Figure 2: A news graph of a crime incident from Bengal

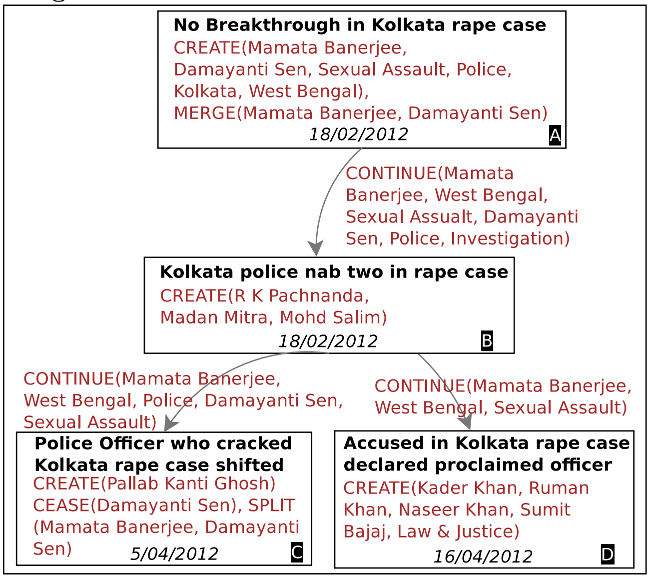


Figure 2 shows an example graph that gets created using this technique for a set of 4 articles covering a crime story. Nodes A and B talk about the incident being reported, reactions from various sections of the society and the progress of investigation. Then, there is a fork in the story, with node C talking about the fate of the investigating officer Damayanti Sen, and node D talks covers the judicial probe of the incident. The nodes C and D cover different different aspects of the same story.

Topic Detection and Tracking. Topic Detection & Tracking (TDT) is a well studied problem [7, 1, 5]. In TDT, a topic is defined as a collection of important words by considering documents to be bags-of-words. In "Topic detection", one determines clusters of articles that discuss the same topic, while in "Topic tracking", one detects stories that discuss a previously known topic [2].

Story Evolution. Kim et al.[6] propose a method for detecting and tracking latent topics that appear within news articles, and visualize how the news corpus evolves from one topic set to another using topic similarity metrics. Subašić et al.[13] show how an event emerges, changes and disappears by dividing each event into several time stages and forming a network of co-occurence of terms based on frequency and time relevance. Nallapati et al.[9, 8] present a system to discover subtopics within a news topic and construct a graph showing their inter-dependencies. These systems are best-suited for simple linear news stories and may not work well for complex stories. In an extension to their previous work [10], which finds the most coherent chain of articles that cover the story connecting 2 input articles, Shahaf et al.[11] construct a roadmap of chains of stories, which may have many branches of smaller stories.

Summarization. Barzilay et al.[3] talk about multidocument news summarization to generate a coherent and comprehensive summary from an input sequence of articles.

Visualization. Vydiswaran et al.[15] design a system to explore news by querying for topics, time range and locations. However, they process articles per user query. In addition, there is no notion of tracking news evolution on a timeline.

As far as we know, our work is the first attempt towards using a rich graph structure built out of the raw articles, that is amenable to repetitive querying without any significant cost of post-processing. Having such a structure, allows us to run efficient algorithms on the news article stream to mine and track the underlying news stories and to provide a personalised context to the story tailored to the user's query. Further, by visualizing these stories on a timeline, helps the user follow the progression of the story through the different actors and topics.

3. CONTEXTFULL NEWS BROWSING

In this section, we first define a notion of contextuality in a news browsing system, and motivate the need for the system to serve multiple flexible and personalizable contexts to a user as per her requirements and/or preferences. We reason why we think this is a necessary feature of a news browsing system. Next, we show how rich contexts can be served to a user on the fly based on her intent from the graph structure created from the input news corpus, taking specific examples.

3.1 Formal definition of Context

Let us attempt to define clearly what we mean by context.

> Given a news article or articles (from here on, referred to as our result set S), *context* is that set of stories preceding, co-occuring or following that article or articles which enhance our understanding of the events or ideas described in that article or articles.

Context for a user using our system, comprises of all the additional articles, topics, actors, events, etc which help to interpret a particular the story in a broader sense. Adding context serves to make it easier for users to better understand a chain of news articles around an event. For eg., consider a crime story where the prime suspect is X. To fully understand the involvement of X in the story, one may be required to go through articles which would have first reported the crime (and not mentioned X). These articles serve as context. Hence, one way to think of context-serving articles are *articles which are coherent with articles containing X and have still more information to provide.* We use our precomputed back-end news graph to mine context-adding articles.

3.2 Metrics to quantify context of an article

Given this intuition, we now describe 3 metrics that together capture the utility of showing an article b alongside the result set S. Utility here is captured in terms of the amount of context (which should be both coherent and relevant) that is given by an article to S.

We first define the notion of strength Ω (that is used in 2 of the metrics) of an entity[1] λ at time τ, which captures this entity's popularity at τ as a function of the number of articles featuring λ published in the time window $[\tau - P, \tau + F]$.

$$\Omega(\lambda, \tau) = \frac{\sum_{a \in \mathcal{A}[\tau - P, \tau + F]} \mathbb{1}(\lambda \in \mathcal{E}(a))}{|\mathcal{A}[\tau - P, \tau + F]|} \qquad (1)$$

P and F are past and future time windows that we keep at 50 days each, and $\mathcal{A}[\tau - P, \tau + F]$ is the set of all articles published in this time window. $\mathcal{E}(a)$ is the set of all entities featuring in article a.

Now, we define the following 3 metrics.

1. $coh_a(b)$: Coherence of article b with article a
2. $cont_a(b)$: Context added by article b to article a
3. $r(a, b)$: Overall article similarity between a and b

Coherence

Given an article a, we define $coh_a(b)$ as the *coherence* between a and some other article b. Intuitively, coherence captures the continuity of the story when the two articles are read in succession. Clearly, coherence between 2 articles depends on the amount of their entity overlap. Moreover, common entities with high strength should contribute less to coherence. This idea is similar in spirit to that of "Inverse document frequency" in Tf-Idf. Words (entities) common across many documents (articles) are less characteristic in describing a particular document (story). More prominent entities will by definition occur in many articles, and can potentially tie many unrelated stories together.

To substantiate, a globally popular entity such as "Soccer" should contribute less to the coherence score of a story than particular instances of soccer stories (having less Ω) like "EPL 2012" and "UEFA Cup 2012". So, we quantify coherence as

$$coh_a(b) = \frac{\sum_{\lambda \in \mathcal{E}(a) \cap \mathcal{E}(b)} \left(\frac{\Omega(\lambda, \tau_a)}{\Omega^*}\right)^{-1}}{|\mathcal{E}(a)|} \qquad (2)$$

where $\Omega^* = min_{\lambda \in \mathcal{E}(a) \cap \mathcal{E}(b)} \Omega(\lambda, \tau_a)$ is the minimum actor strength and τ_a is the time at which article a is published. Ω^* is used to normalize $coh_a(b)$ between 0 and 1.

Context

Along the same lines, we define $cont_A(b)$ as the *context* given by an article b to an article set (news story) A. There are two different kinds of context that a user may be interested in (exemplified on a Fuel price hike story; see Figure 3):

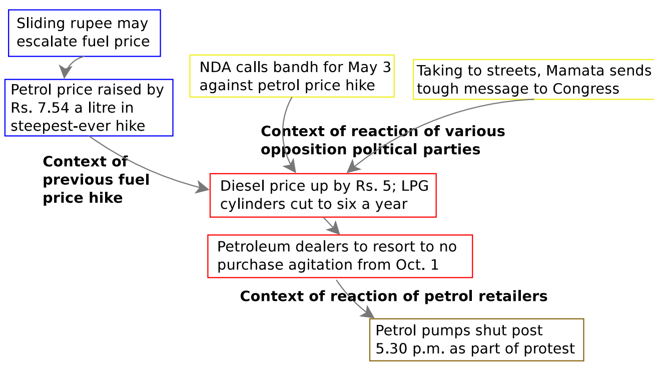
1. Context provided by new entities not featuring in the story of A. For instance, context provided by the reactions of different political parties to a story on fuel price hike (shown in yellow)
2. Context of the stories around the same entities in a different setting. For instance, context provided by the story of fuel price hike of May 2012 to the one in October 2012 (shown in blue)

[1]Entity can refer to an actor or a topic

k	**Headlines of high-context articles**
0	*Footage shows Kim Jong-un in military drill (08/01/12), North Korea plans nuclear test (24/01/13), North Korea says its in state of war with South Korea (30/03/13)*
0.5	*Army chief removal seen as purge (16/07/12), North Korea's new leadership is turning toward reform: Cumings (01/03/12)*
1	*Eternal leader Kims body to be enshrined (12/01/12), Emerging Stocks Sink as Kim Jong Il Death Sows Stability Concern (30/12/11)*

Table 1: Context-adding articles for different k

Figure 3: The fuel price hike story (September 2012) and its contexts



It is not fair for a system to assume preference for a particular kind of context in general, and a user depending on her background and the task at hand, may prefer one kind over another. Hence, we drive this feature through a user-adjustable parameter k. We quantify context as

$$cont_A(b) = \frac{\sum_{\lambda \in \mathcal{E}(b)} \Omega(\lambda, \tau_b) * \left(\frac{\sum_{a \in A} \mathbb{1}(\lambda \in \mathcal{E}(a))}{|A|}\right)^k}{|\mathcal{E}(b)|} \qquad (3)$$

where τ_b is the time at which b is published, $\frac{\sum_{a \in A} \mathbb{1}(\lambda \in \mathcal{E}(a))}{|A|}$ captures the strength of entity λ in set A and $k \in [0, 1]$ is the parameter which influences which of the 2 kinds of context is more prominent. A lower k dampens the weight given to the strength of common entities between A and b, and hence favours context around the popular entities (high Ω). Conversely, a higher k forces articles having a good overlap of occuring entities with set A to result in a higher context score. In table 1, we show the different articles that score high on context on 3 different values of k. The base story A (*North Koreans mourn Kim*[2]) is an article from 2012 about the late Kim Jong-il[3]. For $k = 1$, the context-adding articles closely belong to the topics around the death of the leader. The other extreme ($k = 0$) helps discover broader stories of the fallout of the leader's death, for example on the economy, foreign policy, etc.

Overall article similarity

[2]http://www.thehindu.com/news/international/north-koreans-mourn-kim/article4206294.ece
[3]http://en.wikipedia.org/wiki/Kim_Jong-il

The overall article similarity $r : \mathcal{A} \times \mathcal{A} \rightarrow [0,1]$ between 2 articles where $\mathcal{A}$ is the set of all articles is defined as:

$$r(a,b) = sim(a,b) * Jacc(\mathcal{E}(a), \mathcal{E}(b)) * e^{-\alpha|\tau_a - \tau_b|} \quad (4)$$

Here, *sim* is cosine similarity between articles represented as vectors in the TfIdf space, *Jacc* is the jaccard similarity between the entity sets of articles a and b, and the final term is to factor in the time elapsed (in days) between their publishing dates (τ_a & τ_b). The value of α was kept at 1. This score is pre-computed for each pair of articles.

Next, we normalize each of these metrics to [0, 1] by dividing with the maximum score respective metric that is observed in the window defined by $[\tau - P, \tau + F]$. This step is needed so we can compare these scores across different news stories. Our net score to bring out utility of presenting an article b is the product of the 3 metrics.

3.3 Context-extraction algorithm

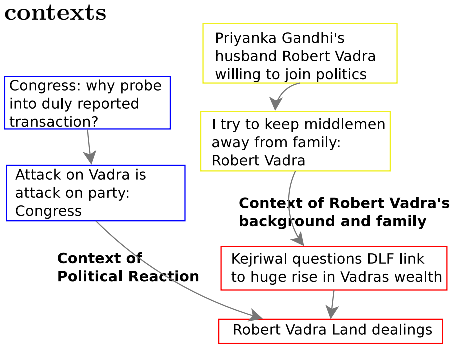
The setting is as follows: A user issues a certain filter query (like "**Corruption** AND **Manmohan Singh**"). The result article set S is computed by applying this filter on the news graph backend. Now, we want to identify articles related to S via a path in the graph, which would be helpful to the user to understand the story and the bigger picture. Our algorithm can now be described as follows: We attempt to get "most useful" neighbourhood of S. The algorithm is a walk on the news graph starting from articles in S, going as far as till we get the desired number of stories to present to the user. This is similar to a user clicking "Next" at the bottom of a search results page to get more results.

GetNeighbourhood
input: Input article set S, MaxStories θ
output: Neighbourhood N
$Q \leftarrow \{\}$ $\triangleright$ Empty Priority Queue
for all node $n \in S$ **do**
  Enqueue(Q, $(n, 1)$)
end for
while Is_Empty(Q) == **False** and $|N| \leq \theta$ **do**
  (v, w) = Dequeue(Q)
  $N \leftarrow N \cup \{v\}$
  for all node $n \in Neighbours(v)$ **do**
    $w_1 \leftarrow coh_v(n)$
    $w_2 \leftarrow cont_S(n)$
    $w_3 \leftarrow r(v, n)$
    $w_{eff} \leftarrow w_1 * w_2 * w_3 * w$
    Enqueue_Or_Update_Node(Q, (n, w_{eff}))
  end for
end while
return N

Here, *Neighbours* returns all of a node's neighbours in the graph. Q is an object of priority queue ADT[4] which was implemented as a heap. *Enqueue* adds a node according to its weight to the heap. *Enqueue_Or_Update_Node* has the semantics that it updates the weight and position of node n in Q if found, else adds the node to Q. *Dequeue* returns the node with the highest weight in the heap. We call $GetNeighbourhood(S, \theta)$ where S is the result set of a user's query and $\theta = 30$ for our experiments. The choice of θ is driven by the front-end interface, and we found 30 to work well. The intuition behind this algorithm is that we want to identify those nodes from the neighbourhood of the result set S, which are both coherent and contextual to S, and do so in an efficient manner (variant of BFS on the graph).

[4]Abstract Datatype

Figure 4: Robert Vadra corruption story and its contexts



3.4 Some examples of context extracted

In this section, we present sample news graphs generated on various stories and show how context is useful for each of them.

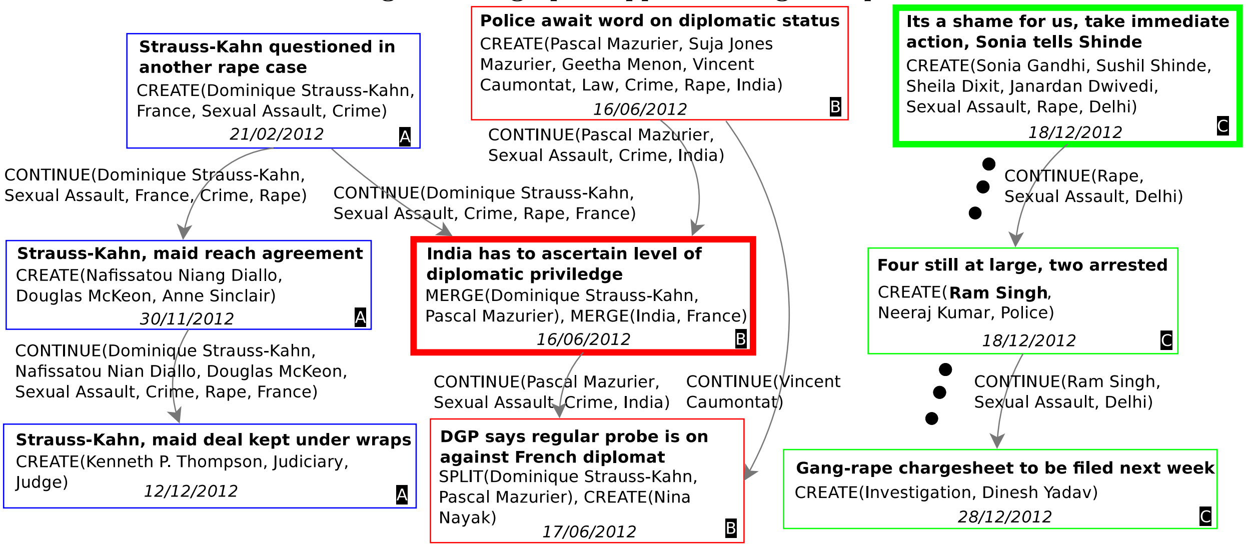
Consider Figure 4 showing the graph of the story around Robert Vadra's alleged corruption scandal reported in the news. As the story progresses in time, the articles start to assume that the reader has enough knowledge of who Robert Vadra is, what is he known for, how has this controversy spiralled to include the top political parties of India, etc. Hence, these articles become increasingly inaccessible to a user with less prior context. However, using the graph, we can provide this context at any stage. Moreover, while following the story, the reader can look for how have the political parties reacted to this story as another context.

Consider Figure 5 where 3 distinct stories are presented. The nodes marked A(in blue) cover a case involving Dominique Strauss-Kahn (DSK), those marked B(in red) cover of a case surrounding Pascal Mazurier(PM), and those marked C(in green) talk of the internationally reported Gang-rape case that happened in Delhi(DGR).[5] Two key things can be noted here. Even with a fair overlap of topics (Sexual Assault, Rape), the DSK and DGR subgraphs are not directly linked to each other. On the other hand, the PM and DSK stories have a direct link in between them. This points to the fact, that the graph generation algorithm is successfully able to determine how close or far stories (subgraphs) are to each other.

A key strength of our approach is that we allow the user to iteratively query the underlying corpus on multiple parameters and never have to recreate the graph. Having created the graph once, we just filter out only the relevant nodes clusters and visualize them on a timeline as events.

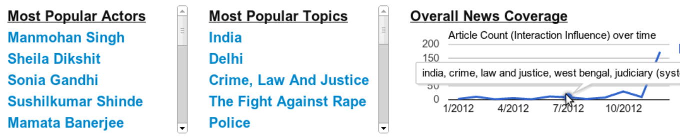
[5]In no way do we want to use this unfortunate incident or others discussed in the paper in any way except to exemplify the technicalities of our work.

Figure 5: A graph snippet showing multiple stories



4. ESTHETE: A SYSTEM FOR CONTEXT-FULL NEWS BROWSING

Figure 6: Analytics which help the user better understand the stories



In this section we first describe the key features of our system. Next, we give a block design of our system, followed by a detailed explanation of each component. The reader can experiment with the system at `http://konfrap.com/esthete`.

4.1 Key features of ESTHETE

4.1.1 *Queryability*

Our system is queryable by users, where a query entered by a user is in the form of filter on the actors and/or topics talked about in the news corpus. Hence, the users need only to pick a suitable filter to study the intended story in desired detail. For eg., an example query could be "**Corruption** AND **Robert Vadra** AND **Haryana**" which yields the set of all articles S around the corruption scandal involving Robert Vadra (an Indian businessman) in Haryana (a state in India).

4.1.2 *Returning context-adding articles*

Our system returns additional articles which add the the appropriate kind of context desired by the user. This also helps the user see the bigger picture. We are able to find these articles given the user's filter query (which generates the set S), using algorithm $GetNeighbourhood$, discussed in Section 3.2. The news graph not only tells us the articles related to a particular story (neighbourhood of a subgraph), but also *qualifies* how these articles are related, by way of the nature of the transformations that the edges cover.

4.1.3 *Faceted Searchability*

Our system is a guided environment where a user is presented with popular actors, topics, events, time periods, etc automatically from the news corpus, which guide her search to discover and explore more popular stories first. These also serve to label different stories and guide the user towards a more refined query around specific substories. Figure 6 shows a snapshot of this feature.

4.1.4 *Structured storage of an article stream*

The rate at which new articles are published is growing exponentially. If a system has to do well at taking in a live input stream of news articles, it can afford to process this set only once, and then store this set in a structure which is efficient to maintain and query later. Since our back-end is a graph, we can efficiently query looking only at the graph metadata. Moreover, if the articles are coming from a third-party, they need not be stored at our server (once we build the graph), we can query them on-demand.

4.1.5 *On-demand detail*

The system should respect the user's final say in the detail at which different stories are visualized. Should the system just give a 10-line summary of the event captured by 5 articles? Or should it focus in to the 5 articles, creating 2 sub-stories within them? This decision should be left to the user as a preference.

4.1.6 *Summarizing News stories*

We summarize news articles constituting a particular news story to make it easy for a user to the gist of the story. This helps the user to quickly decide whether the story interests her or not.

4.2 System Overview

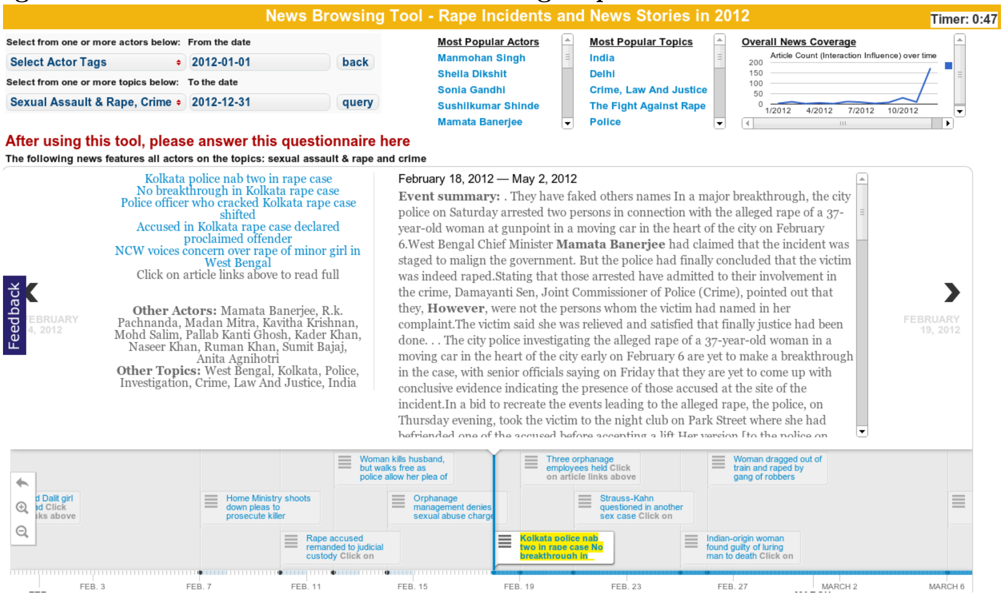


Figure 7: A screenshot of our tool visualizing Rape-related cases from India in 2012

Figure 8 presents our system architecture. It can be divided into two major components - Offline and Online. The offline component handles incoming news articles, and augments them into our underlying graph data structure representation stored in the database. The online component interfaces with the user, and based on the user's query, identifies the parts of the graph to visualize and shows the relevant stories. We will now briefly describe each aspect of our system.

News Corpus

Our news corpus is a subset of articles from the Indian national daily *The Hindu*[6], across a variety of broad categories like Crime, Economy, Government, etc. *The Hindu* was chosen because it is a popular Indian daily, and offers a convenient way to download articles along with rich meta-data like Topic tags.

Entity Extraction

We used the OpenCalais[7] API to extract all the entities appearing in the articles. In particular, we call the people featured in the articles as actors. We found the result of OpenCalais to be the most accurate among the NER tools we experimented with. For sanitizing the mined entity set (correct spelling errors, remap aliases of the same entity to a unique identifier, remove ambiguous entity tokens), we used an ontology(YAGO[8]) based approach.

Topic Detection

Along with the entities for an article, we also need the list of topics that were talked about in the article. We relied on the Hindu articles being hand tagged-at-source by rich and relevant Topic tags. These are much more expressive compared to an automated technique of topic detection.

[6]http://thehindu.com
[7]http://opencalais.com
[8]http://www.mpi-inf.mpg.de/yago-naga/yago/

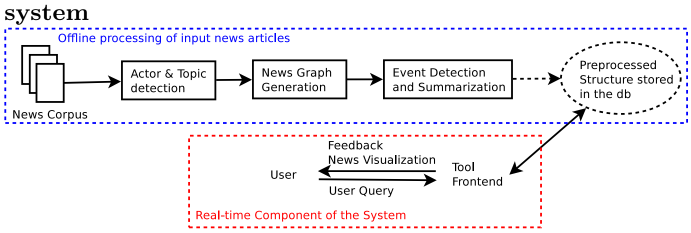


Figure 8: System diagram of our News Browsing system

News Graph Generation

We follow the same graph generation algorithm described in Choudhary et al.[4]. We extended their algorithm to augment new articles as they are published on to the graph built up to that time. The following sequence of update steps are to be followed on augmenting a new article a published at τ, into the graph structure.

1. Preprocess a to mine relevant actors and topics
2. For all articles $b \in \mathcal{A}[\tau - P, \tau]$
   - Mine and score all transformations between the articles a and b
3. Find an edge-covering of the new transformations connecting node a to nodes in $\mathcal{A}[\tau - P, \tau]$

By taking the recommended values of the parameters($P =$ 50 days), we were able to generate a news graph on 1000 nodes in around 4 minutes on a Dell Intel Centrino machine. Adding newer articles to this graph was significantly cheaper, taking only around 10 seconds depending on the article size, etc. Moreover, it only looks at articles within a time window, and hence different parts of the graph can be generated in parallel.

Event detection

The news graph created is clustered with a standard graph clustering algorithm: Markov graph clustering [14]. We used the Article similarity $r(a, b)$ as defined in 3.2 as the flow (transition probability) from a node a to b by normalizing r to get the quantity $r_n(a, b) = \frac{r(a,b)}{\sum_{b \in N(a)} r(a,b)}$. We can now formulate the flow matrix using these scores and run the Markov Clustering Algorithm. We used the standard inflation factor of 2 as our input for forming the clusters.

As we show in Section 5.2.3, events determined by way of clustering on the graph are preferred by users to a similar k-means text clustering done on raw articles in most cases.

Event summarization

To make it easier for the user to understand an event, we summarize the articles of this event to give a broad idea of the event and show this summarization alongside the articles capturing that event (Section 4.3). For this, we tried out various document summarizing tools. We finally decided to Text Compactor [9] which is based on Open Text Summarizer[10] and has a convenient API.

4.3 Frontend interface

Our interface is built on HTML using Javascript and PHP on the server-side, using TimelineJS[11] and Google Visualization API[12] libraries. Figure 7 shows a screenshot of our webtool. The interface has 3 primary parts: the Filtering & feedback pane, the News article(s) & summary pane and the Timeline pane.

Filtering & feedback pane

Our system deeply analyzes the news corpus to mine all the featured actors and topics, and makes them queryable. The user selects one or more actors and/or topics to filter news only about them. The user could also restrict on a particular time window. Relative to the filter parameters set by her, the user gets more context by way of getting a ranked list of "Popular Actors" and "Popular Topics". In addition, we also plot the number of articles published against time to study what stories were popular at what times.

News articles & summary pane

Having selected a particular story to focus on, the user can read all articles arranged sequentially so it is easy to read them in succession. On the right, we show a summary generated from the articles of this story by the methods discussed in Section 4.2. On clicking an article, the user is served the full article text. The user is also shown the full list of actors and topics discussed in this story.

Timeline pane

Having selected suitable filters, the user is presented with all the different events detected from the graph on the timeline. These appear as bubbles on the timeline, with a start and end date. The user can easily move in time, skimming through uninteresting news events. More popular events (as judged by the density of nodes & edges in the cluster), are highlighted in yellow to guide the user. On clicking a particular bubble, that event is highlighted and the corresponding story is shown in the middle pane.

[9] http://textcompactor.com/
[10] http:/libots.sourceforge.net/
[11] http://timeline.verite.co/
[12] http://developers.google.com/chart/interactive/docs/reference

5. EVALUATION

We conducted several user studies to evaluate ESTHETE, which is a system composed of a data structural graphical representation of news articles with a timeline-based visualisation.

- **Usability and Effectiveness:** For our system to be useful, not only does it have to be easy to use, but also help users understand the big picture of the news story they are interested in. Our experiments in 5.2.1 evaluate these aspects.
- **Precision:** A story presented on the timeline has several different aspects associated with it. A user can filter out the relevant aspects by selecting or deselecting topics and actors identified by the system. When a user issues a filter query, how relevant are the returned results? Note that querying is not just a keyword match, but also computes and returns context-adding articles (Section 3.2).
- **News graphs as a framework:** Finally, our third experiment establishes the usefulness of representing news articles as a graph. We compare our news graph framework to two other baselines – the articles themselves, and a clustering of articles using k-means clustering.

5.1 Setup

Our news corpus consists of articles published by "The Hindu", a popular Indian newspaper. We choose three groups of articles corresponding to "rape", "corruption" and "petrol price". We fired each of these keywords into the website's search form and downloaded all articles that were returned. Therefore, we had three article groups corresponding to the three keywords. The total no. of articles is 10,121, published between 2010-2013. Each group had specific *subsets* of articles corresponding to recent stories in the news. The first group of articles contained, in addition to articles about rape cases in various parts of India and abroad, articles about a gang-rape of a woman on a bus, which led to several weeks of protests in Delhi. Similarly, the second group of articles contained a "sub-story" about a very high-profile anti-corruption protest in Delhi, which then morphed into a political story about accusations of corruption against the son-in-law (Robert Vadra) of a powerful politician. The third group of articles contained a sub-story about petrol prices in the year 2012 which lead to political unrest.

Each article group was then organised as a news graph using the techniques explained in Section 4.2 and the following three sub stories (described in the previous paragraph), were used for experimentation i) the gang-rape incident in Delhi (S1), ii) corruption stories related to Robert Vadra (S2), and, iii) petrol price (S3).

The three user studies made use of the following setup: an article group, in its entirety was presented to the user on our interface. The users were then allowed to interact with the system using whatever filters they felt would be useful.

Story	# users	Ease of use		New Information		Big Picture	
		E	G	E	G	E	G
S1	22	3.76	4.25	4.25	3.4	4.3	3.66
S2	12	3.83	4.18	4.29	3.5	4.39	3.71
S3	6	3.75	4.33	4.4	3.5	4.4	3.66
All	40	3.78	4.24	4.28	3.44	4.34	3.67

Table 2: Average ratings for usability (out of 5); E:ESTHETE, G: Google News

Story	Questions asked
S1	*Did you learn about the reactions of different political particles to the Delhi rape incident?, Did you learn about the health condition of the victim following the crime?*
S2	*Did you learn about the interactions between Arvind Kejriwal and Robert Vadra?*
S2	*Did you learn about the reactions of AIDMK and DMK to the petrol price hike?*

Table 3: Yes-No questions asked to compare ESTHETE with Google News

The focus of each set of experiments (described below) were the three sub stories.

5.2 Experiment design and results

5.2.1 Usability and Effectiveness

Usability

In order to evaluate the usability of ESTHETE, we asked 40 users to pick a story of their choice (among S1, S2 and S3) and to study that story by interacting with ESTHETE and Google News, in any order, for 15 minutes respectively. After the users had interacted with both the systems, we asked users the following questions:

- **(Q1) Ease of use:** On a scale of 1 to 5, how would you rate both the systems on their ease of use (with 5 corresponding to "very easy")?
- **(Q2) New Information:** Given your current knowledge of the story, how would you rate both the systems on their ability to provide new information that you weren't aware of before interacting with the systems? (1 corresponds to "no new information", 5 corresponds to "more than satisfactory")
- **(Q3) Big Picture:** How would you rate both the systems on their ability to aid your understanding of the big picture of the story? (1 corresponds to "not at all", 5 corresponds to "perfectly well")

The results are tabulated in Table 2. Clearly, the users found our system to be more effective than Google News in gaining new information and understanding the bigger picture of the story, in a short time frame. This shows the ability of our system ESTHETE to bring forth and coherently link important nuggets of information. This also exhibits our system's ability to present the user with the necessary context that aided their understanding of the bigger picture of the story. The users also found the system easy to use and navigate.

Story	# responses	ESTHETE only	Google News only	Both	None
S1	36	23	2	9	2
S2	12	9	0	2	1
S3	6	3	0	3	0

Table 4: Effectiveness of ESTHETE

Story	# users	Avg. # filters	Precision
S1	10	3.4	0.89
S2	10	2.5	0.92
S3	8	2.1	0.88

Table 5: Precision of articles returned by the system using filters

Effectiveness

We also compared the effectiveness of our system, to help users understand a story in depth, with Google News. We asked 18 users to use either our system or Google News in order to understand any of the three stories listed above. The users had to pick one story of their interest and were allowed to interact with their chosen system for a session lasting 25 minutes. The user's task was to learn as much as they can about their chosen story using a system. Then, once they had completed their interaction with their chosen system (ESTHETE or Google News), they were asked to answer a series of very specific questions related to the story they studied. Some of these questions are shown in Table 3. For these questions, the users had to answer a Yes or a No based on their perception of how easy or difficult they feel it is to answer these questions using the system they just used. The users replied Yes if they felt that they could answer the question using the system they just used and vice versa.

The results are tabulated in Table 4. The table entries shows for a particular story, the number of responses for which the users responded that they can answer the underlying question using only ESTHETE, only Google News, from both and from none of these systems. The results shows that users found it easier to answer the story-related questions using our system, in comparision to Google News. This clearly demonstrates that our system is more effective than Google News in aiding the users get enough information for answering the story-specific questions. This study brings out the ability of our system to improve the user's understanding of the underlying story.

5.2.2 Precision of the system

Description. In order to measure the precision of the system, users who had intimate knowledge of the three stories were shown the news graph for the entire timeline and allowed to filter these stories using whatever keywords they felt were appropriate from the list of entities and topics extracted by our system. For example, if they filtered on "Robert Vadra" (story S2), they would retain articles which not only contained "Robert Vadra", but also articles which had a strong connection to him in relation to the corruption allegations against him. The users than judged the relevance of each of these filtered articles on a 2-point scale (either rel-

Story	No. of users	ESTHETE preferred	B1 preferred	B2 preferred
S1	7	100%	0%	0%
S2	6	83%	0%	17%
S3	7	100%	0%	0%

Table 6: Comparison of news graphs against baselines B1 and B2

evant or not relevant). On average, for each story, users used 2-4 filters (that is, they judged the relevance for 2-4 subsets of articles within the larger timeline).

Results. The results of this experiment are tabulated in Table 5. The precision values are in the range 88%-92%, indicating that our representation framework is very effective in identifying different contexts.

5.2.3 News graphs as a framework

Description. In order to determine how news graphs compare against other ways of representing news corpora, we designed the following two baselines, both of which also made use of our timeline interface. Our first baseline (B1) simply presents *all* articles (corresponding to the article groups mentioned above) on the timeline according to date of publication. For our second baseline (B2), we clustered each article group using k-means clustering, and then presented these clusters on our timeline. Therefore, visually, all three techniques looked the same. Three timelines corresponding to the three techniques were then simultaneously presented to the user and the user was asked to pick the one that they preferred in terms of understanding the various stories.

Results. The results of this experiment are tabulated in Table 6. For all 3 article groups, our representation is preferred over the other two baselines. This implies that not only is the quality of the clusters formed by our technique better than k-means, but also that the context we provide to the users is actually found to be beneficial. For the story $S2$, $B2$ was preferred by 1 user because of the time-localised characteristic of the story $S2$, due to which that user couldn't benefit from the context added by our system.

6. CONCLUSIONS

The main contribution of this paper is the formalization of a natural notion, personalized flexible context extraction, that corresponds to a need that all news consumers have felt to some extent or the other. We have outlined the structure of a system that is geared to provide a richly featured news browsing experience that revolves around the notion of context. Using the news graphs first described by Choudhary et. al. [4] we have shown how such a news browsing experience can be achieved. Our news browsing tool, ESTHETE, attempts to provide the kind of context-aware news consumption environment that makes it possible for a lay user, or an expert, to explore the various aspects of a story to whatever level of detail required, and does not tie down the user to particular structures extracted from the corpus. The fundamental difference between this work and the literature preceding it is that earlier papers took the approach of developing algorithms for processing the corpus to respond to certain classes of queries whereas we approach news browsing as a data structuring problem, preprocessing the corpus into a structured entity that contains the complex relations that a human being needs to unravel the context of a particular news story. Our user studies have shown that ESTHETE addresses the need for context without sacrficing on the need for current information and overall provides complex features in an easy-to-use interface.

7. REFERENCES

[1] J. Allan. Introduction to topic detection and tracking. In J. Allan and W. B. Croft, editors, *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 1–16. Springer US, 2002.

[2] J. Allan, editor. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] R. Barzilay, N. Elhadad, and K. R. McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1):35–55, Aug. 2002.

[4] R. Choudhary, S. Mehta, A. Bagchi, and R. Balakrishnan. Towards characterization of actor evolution and interactions in news corpora. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 422–429. Springer Berlin Heidelberg, 2008.

[5] M. Franz, T. Ward, J. S. McCarley, and W.-J. Zhu. Unsupervised and supervised clustering for topic tracking. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 310–317, New York, NY, USA, 2001. ACM.

[6] D. Kim and A. Oh. Topic chains for understanding a news corpus. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part II*, CICLing'11, pages 163–176, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple semantics in topic detection and tracking. *Information Retrieval*, 7:347–368, 2004. 10.1023/B:INRT.0000011210.12953.86.

[8] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 198–207, New York, NY, USA, 2005. ACM.

[9] R. Nallapati, A. Feng, F. Peng, and J. Allan. Event threading within news topics. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 446–453, New York, NY, USA, 2004. ACM.

[10] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 623–632, New York, NY, USA, 2010. ACM.

[11] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: generating information maps. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 899–908, New York, NY, USA, 2012. ACM.

[12] I. Subašić and B. Berendt. Web mining for understanding stories through graph visualisation. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 570–579, 2008.

[13] I. Subašić and B. Berendt. Story graphs: Tracking document set evolution using dynamic graphs. *Intell. Data Anal*, 17(1):125–147, 2013.

[14] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, Utrecht, May 2000. http://www.library.uu.nl/digiarchief/dip/diss/1895620/inhoud.htm.

[15] V. V. Vydiswaran, J. van den Eijkhof, R. Chandrasekar, A. Paradiso, and J. St. George. News sync: Enabling scenario-based news exploration. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–10, 2011.