Putting news in context, automatically

Larry Birnbaum, Miriam Boon, Scott Bradley, and Jennifer Wilson
Northwestern University
Intelligent Information Laboratory
Ford EDC, 2133 Sheridan Rd., Evanston, IL 60201 USA
l-birnbaum@northwestern.edu

ABSTRACT
Intelligent information technologies can be designed to automatically and immediately provide both journalists and ordinary newsreaders with a broad range of the contextual information they need in order to understand news stories. Our work on specific systems has inspired the creation of a general architecture and platform for developing applications capable of automatically identifying, selecting, and presenting relevant contextual information. These systems can interact directly with news consumers through mechanisms such as browser extensions.

Categories and Subject Descriptors
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Algorithms, Design, Human Factors

Keywords
Computational journalism; contextual search; intelligent information systems

1. INTRODUCTION
A persistent problem in understanding news—and information more generally—lies in understanding the broader context of the particular news story or other informational content you're reading. What is the background for this story? What is distinctive about it? Who and what are the people, places, and organizations involved? What is their history? How do the events in this story, or their scale, compare with other, similar events in the past? What are the possible implications of this story? How does it look from a variety of different perspectives? And so on.

For more than a decade, we have been developing contextual and intelligent information technologies aimed at addressing this problem by *automatically* providing both journalists and ordinary newsreaders answers to the kinds of questions outlined above. More recently, we have begun collecting, organizing, and abstracting the functions, techniques, and overall architecture that previous projects have used in order to create a general platform and toolkit for such contextual information systems—the Northwestern InfoLab ***News Context Project***. Our ultimate vision is to create an open source framework and toolkit with methods for automatically identifying, selecting, and presenting the broad range of contextual information that users need in order to gain a more nuanced understanding of news stories and other information. These applications can interact directly with news consumers through mechanisms such as browser extensions, or mediated by publishers via content management systems (CMSs).

2. MOTIVATION
Journalism's role in society, arguably, is to uncover and communicate the truth [12]. Providing adequate context is widely understood to be a necessary aspect of this process [6, 16, 18, 20]—and yet, at the same time, in practical settings the importance of context is rarely acknowledged except in terms of the need for analysis in addition to factual reporting. One survey of 242 journalistic codes of ethics found that only 12 per cent included some variation on "Provide commentary, background, and criticism," which could be interpreted to encompass context.

Part of the explanation for this disconnect may lie in the practical constraints under which journalism has historically operated. The amount of context a journalist can provide, particularly in daily news, has been subject to strict constraints based on column space or broadcast time. Online, however, these bottlenecks are eliminated. Yet today, while many online codes of ethics for bloggers include attribution and linking to sources, they still do not explicitly mention other forms of context [1, 4, 7, 13, 19].

Of course, although the space available for contextualizing information online is now scalable (within reason), the skilled labor necessary to produce that content is not. However, this is a constraint that computer science is uniquely suited to address. We believe that many useful forms of context can be provided automatically. That is the goal of the ***News Context Project***.

3. BACKGROUND AND EXAMPLES
Many researchers have created systems that provide context in a variety of forms, such as social [14] and automatic [8] annotation, social media, relation to personal context [17], etc. Others have created technologies that were not used to provide context for readers, but could easily have been used for that purpose [5]. Over the past decade and more, we, and others affiliated with our lab, have developed a basic approach to the automatic retrieval of information potentially relevant to a given content item [2, 3, 17]. As a tangible example of a system constructed along these lines for use in relation to news articles, consider *Tell Me More* [9–11]. *Tell Me More* provides newsreaders and journalists with additional information not contained in the article they are currently reading, but that *is* available in other news articles

covering the same situation. The system (see Figure 1) works first by identifying a cluster of stories covering the same topic—either by formulating a query based on frequent terms and named entities contained in the initial story, and then searching online news repositories for related stories based on this query, or by identifying previously aggregated clusters of stories containing the initial story within such repositories. It then searches within these stories to find additional information *not* contained within the original story, and presents the paragraphs containing this information (with the information itself highlighted) in a window adjacent to the original story.

Figure 1. Tell Me More

In general, determining what constitutes new information is an extremely difficult problem. *Tell Me More* therefore uses a number of simple and easily computed syntactic proxies to identify such new information. These are, first, to search for new *data*—i.e., numbers—that aren't described in the original story (attempting to weed out numbers that serve a rhetorical or structural role in the found story, rather than an informational one). The second is to search for new *people*, *organizations*, and *places* not mentioned in the original story—i.e., named entities. The third is to search for new *quotes* not found in the original story (taking into account truncation and, to some extent, paraphrase). These categories of new information obviously aren't exhaustive—but, critically, they are both arguably useful and computationally feasible.

As another example of such a system, consider *LocalSavvy* [15], shown in Figure 2 presenting an Iranian news source commenting on a visit to Iran by Vladimir Putin.

Given an initial news story as a seed, this system identifies and presents news stories on the same issues or events, sourced from publication venues associated with the various stakeholders in that story. For example, starting from a story about events in Pakistan from a U.S.-based source such as *CNN* or the *New York Times*, the system will find and present stories from Pakistani sources (e.g., *The Dawn* of Lahore, Pakistan), as well as, e.g., Indian sources. *LocalSavvy* utilizes an architecture very similar to that of *Tell Me More*. First, it identifies frequent terms and named entities in the story; this analysis is then used to form queries. The system also analyzes the story to determine stakeholders, whether directly named or associated by location or other information. This analysis is used to determine publication venues that might offer perspective on the situation from the viewpoints of those stakeholders. The queries are then run against these sources, and the results collated and presented to the user.

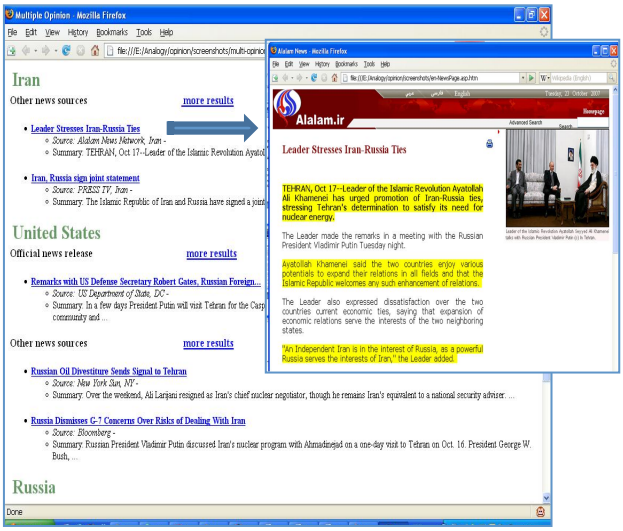

Figure 2. LocalSavvy

4. AN ARCHITECTURE FOR CONTEXTUAL SYSTEMS

Based on our experiences in building these and many similar systems, we have developed an architectural framework for such contextual information systems, appropriately abstracted. This framework comprises, first, a platform that instantiates the following key phases of processing, as depicted in Figure 3:

1. **Content Analysis:** Analyze the content of the item and/or of metadata associated with the item. In the case of textual content or metadata, this involves the use of text analytics such as term histograms, named entity recognition, the use of heuristics based on document structure, etc.
2. **Query Formation:** Based on an assessment of the user's information needs, and the nature of the available information resources, use the results of the previous analysis as a basis for constructing queries that are likely to retrieve content that might address those information needs. This involves selecting and weighting components of the analysis, transforming them in a variety of ways, and adding additional constraints, and may strongly depend on the information sources that will be queried.
3. **Source Determination:** Based on the assessment of the user's information needs and/or the content of the item, and the available information sources, determine which are appropriate to query.
4. **Query Management:** Aim the resulting queries at the available information resources as appropriate, and collect and organize the results as they are returned.
5. **Result Ranking and Extraction:** Filter and rank the query results, and/or extract relevant portions of these results, based

on criteria derived from the user's information needs and the initial content.

6. **Presentation:** Present the selected results or extracted portions to the user in an appropriate format and relation to the original content.

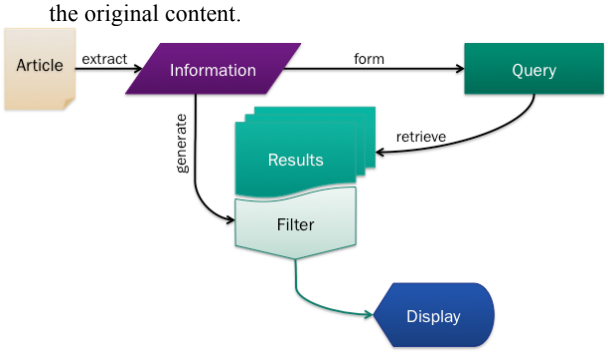

Figure 3. Basic architecture of Context applications

The toolkit's methods are drawn from primarily in-house projects, but also include a number of Python libraries commonly used for related tasks. Together these include standard text analytic tools for tasks such as detecting bigrams and trigrams, computing term histograms, filtering by stop lists, recognizing named entities, etc., for phase 1; tools for using the results of these analytic methods, and other input, to construct queries of appropriate length and complexity for phase 2; tools for eliminating duplicate results, ranking results by comparing them with the original document, or finding appropriate portions of results in phase 5; and so on. These tools make it significantly easier for an experienced developer to construct a new instance of the architecture that applies to a different kind of content, implements a different notion of contextual relevance, uses different information sources, or presents its results differently.

5. EXAMPLE IMPLEMENTATIONS

We have used the toolkit to refine and deploy a number of news context systems of the sort outlined above.

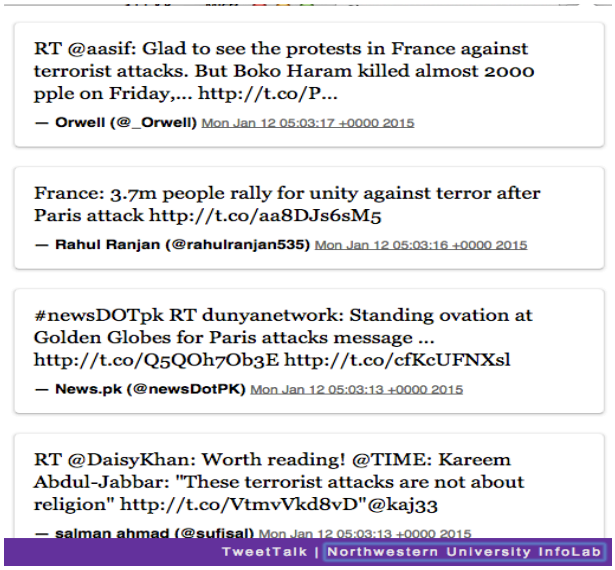

Figure 4. TweetTalk

One of these systems, called *TweetTalk*, searches social media—specifically, Twitter—to automatically find individual and personal comments relevant to a news story or other document you're reading online. Implemented as a browser extension with some server-side components, this system again uses the basic architecture described above. First, it analyzes the current page to find common terms and named entities, and then uses a subset of these terms and entities to search for "Top" tweets, as determined by Twitter, that are relevant to the story (phases 1 through 4). It then attempts to filter out tweets from news organizations—in order to favor more personal reactions—and ranks the remaining tweets in the order determined by Twitter itself (phase 5). The resulting tweets are then displayed in a pop-up window next to the original article (phase 6). Figure 4 shows the system's results given an article from the *New York Times* (January 11, 2015) entitled "Huge Show of Solidarity in Paris against Terrorism," about the march in Paris following the terrorist attack on the magazine Charlie Hebdo and a kosher supermarket there.

Another example system, built using exactly the same platform and component libraries, and also using Twitter as an information source, is called *Stakeholder Tweetback*. This system again analyzes the current page to find common terms and named entities, and uses a subset of these terms and entities to form a query (phases 1 and 2). In addition, it also uses named entity detection to determine the people and organizations involved in the story, identifies their verified Twitter accounts, and then searches Twitter for postings from these accounts relevant to the query (phases 3 and 4). It then presents these tweets to the user in a pop-up window next to the original article (phase 6), organized by the Twitter handles of the stakeholders.

Yet another example of a system we have built using this platform is *readdit*, which automatically identifies reddit discussion threads relevant to a story you're reading online. This system again uses the same basic architecture described above, but instead of searching Twitter, it uses the resulting queries to search reddit. An example of the results based on a BBC story entitled "Paris attacks: Twelve suspects held overnight" (January 16, 2015) can be seen in Figure 5.

6. USER-CONFIGURABLE NEWS CONTEXT PLATFORM

The platform and component libraries described briefly above have made it significantly faster and easier to build new contextual information systems of the sort presented here. However, even with this framework, the construction of these systems remains a job for reasonably experienced developers. We believe the next step in the evolution of these systems is the development of configurable platforms that will make it possible for non-programmers, including ultimately end-users, to construct these sorts of contextual information systems for themselves. In configuring such platforms, users would specify the nature of the contextual relations that mattered to them, and the type of information that would provide the context they seek. These specifications would then determine the appropriate component tools to be used in each phase of the overall process, and how they would be parameterized in order to identify, select, and present contextual information that fulfilled those relations.

Providing appropriate specifications for contextual relations, and for information that can satisfy those relations, requires developing a language for such specifications. In other words, users must be presented with an intuitive representation of the possible "dimensions" of contextual information in relationship to a given content item. Does the contextual information provide

historical background on the initial content? Personal experiences relating to that content? Alternative viewpoints? A basis for comparison? And so on.

Figure 5. readdit, and a reddit thread it finds

The choice and setting of such intuitively understandable dimensions of information relations must, in turn, be mapped to the choice of specific components, and appropriate parameterizations of those components, within each phase of the overall process, in order to produce information bearing the appropriate contextual relationship with the original content item. Our current research is aimed at developing such a specification language, appropriate mappings, and interfaces making it possible for non-programmers to determine and specify the context they need, and thereby configure contextual information systems that meet those needs.

This approach carries another benefit as well: transparency. Systems that automatically provide contextual information—like any systems that provide us with content, such as search engines or recommendation systems—inevitably make editorial judgments. That is, they determine the information we will see (and what we won't see), the form in which we see it, and the order in which we see it—all judgments that have historically been made by human editors.

Human editors presumably make these decisions based on their editorial values. News organizations and other publishers embody these values, which are instilled through discussion and by example, and consumers typically have some sense of these values based on a publisher's brand. Does the publisher value timeliness? Accuracy? Balance? Titillation? From *Bloomberg*, to *The New Yorker*, to *TMZ*, we generally understand the editorial values underlying the choice of information being presented to us.

When computational systems carry out editorial functions, that's not always the case. The editorial values that inform their decisions are often opaque—not only to readers, but, to some extent, even to the engineers who build them. These systems are optimizing something—most likely revenue—but what factors are they juggling, in human terms, to do that, and why do they strike the particular balance they do?

In order to promote transparency, contextual information systems need to be seen as *editorial* algorithms, and therefore specified in terms of editorial factors that are understandable to humans—first of all, to the people who develop them, but ultimately to publishers and readers as well. In other words, these algorithms should be making decisions in terms of factors such as accuracy, importance, balance, color, data, etc.—editorial attributes of stories and information that people understand. These kinds of factors, and the trade-offs among them, would make up a specification language for contextual relations and information attributes as described above. Their presentation to newsreaders in an understandable form would make it possible for these readers to grasp, and ultimately control, the editorial decisions that determine the information they see. A hypothetical interface design for visualizing some of these factors and trade-offs can be seen in Figure 6.

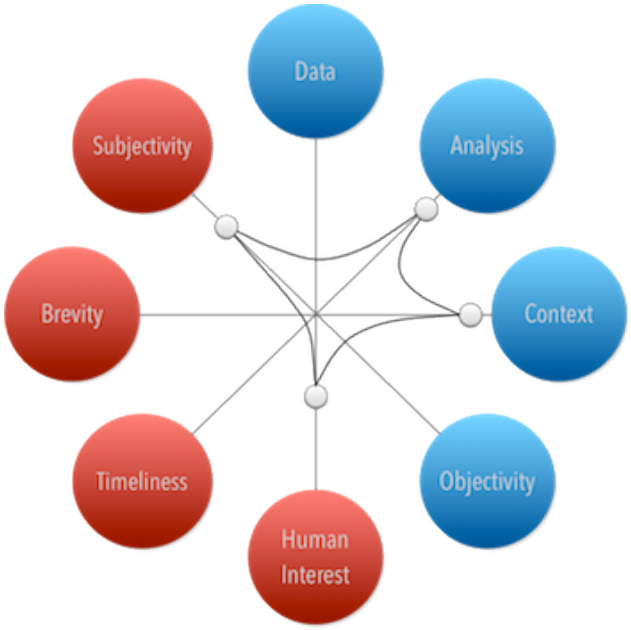

Figure 6. A hypothetical interface for information attributes of stories and context.

Ultimately, editorial models expressed in these terms should be (building on the concept of first-class objects in Computer

Science) *first-class media objects*: They should be explicit, understandable, readable, writable, and ultimately publishable. In this way, people will be able to develop editorial models for context—specifications of kinds of contextual information they deem useful—that can be published, compared, and then adopted for use by news and information consumers on the basis of the editorial values they express.

7. CONCLUSION

We have described an architecture for contextual news information systems (and examples of such systems), instantiated in a platform for constructing such systems along with a component library of useful tools for carrying out the key phases of processing defined by this platform. This platform and set of components make it significantly easier for a developer to build new instances of the architecture that use different definitions of context or search different information sources for relevant information.

We have recently made the code for this platform available on GitHub as an open-source toolkit. Like all open source projects, this is by its nature a work in progress. Although the current methods provide the tools needed to develop interesting systems—such as *TweetTalk*, *Stakeholder Tweetback*, and *readdit*—we continue to expand the space of feasible and potentially useful news context systems enabled by the approach. The contextual information systems described above, and others like them, can be found at infolab.northwestern.edu/context/, which provides browser extensions and web interfaces for trying them out.

ACKNOWLEDGMENTS

We thank the National Science Foundation, the John S. and James L. Knight Foundation, and Google for their support. Miriam Boon has also been supported by a Cognitive Science Fellowship from Northwestern University, and by the University's program in Technology and Social Behavior.

REFERENCES

[1] A Bloggers' Code of Ethics - CyberJournalist.net - Online News Association Ethics and Credibility: 2003. *http://www.cyberjournalist.net/news/000215.php*. Accessed: 2008-11-04.

[2] Budzik, J. and Hammond, K.J. 2000. User interactions with everyday applications as context for just-in-time information access. *Proceedings of the 5th international conference on intelligent user interfaces* (2000), 44–51.

[3] Budzik, J., Hammond, K.J. and Birnbaum, L. 2001. Information access in context. *Knowledge-based systems*. 14, 1 (2001), 37–53.

[4] Cenite, M., Detenber, B.H., Koh, A.W., Lim, A.L. and Soon, N.E. 2009. Doing the right thing online: a survey of bloggers' ethical beliefs and practices. *New Media & Society*. 11, 4 (2009), 575–597.

[5] Diakopoulos, N., De Choudhury, M. and Naaman, M. 2012. Finding and assessing social media information sources in the context of journalism. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), 2451–2460.

[6] Friend, C. and Singer, J. 2007. *Online Journalism Ethics: Traditions and Transitions*. Routledge.

[7] HONcode: Principles - Quality and trustworthy health information: *http://www.hon.ch/HONcode/Conduct.html*. Accessed: 2015-08-13.

[8] Hullman, J., Diakopoulos, N. and Adar, E. 2013. Contextifier: automatic generation of annotated stock visualizations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), 2707–2716.

[9] Iacobelli, F., Birnbaum, L. and Hammond, K.J. 2010. Tell me more, not just more of the same. *Proceedings of the 15th international conference on Intelligent user interfaces* (2010), 81–90.

[10] Iacobelli, F., Nichols, N., Birnbaum, L. and Hammond, K. 2010. Finding new information via robust entity detection. *Proactive Assistant Agents (PAA2010) AAAI 2010 Fall Symposium* (2010).

[11] Iacobelli, F., Nichols, N., Birnbaum, L. and Hammond, K. 2012. Information Finding with Robust Entity Detection: The Case of an Online News Reader. *Human-Computer Interaction: The Agency Perspective*. Springer. 375–387.

[12] Kovach, B. and Rosenstiel, T. 2007. *The Elements of Journalism: What Newspeople Should Know and the Public Should Expect, Completely Updated and Revised*. Three Rivers Press.

[13] Kuhn, M. 2007. Interactivity and prioritizing the human: A code of blogging ethics. *Journal of Mass Media Ethics*. 22, 1 (2007), 18–36.

[14] Kulkarni, C. and Chi, E. 2013. All the News That's Fit to Read: A Study of Social Annotations for News Reading. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 2407–2416.

[15] Liu, J. and Birnbaum, L. 2008. What do they think?: aggregating local views about news events and topics. *Proceedings of the 17th international conference on World Wide Web* (2008), 1021–1022.

[16] Plato, 427? BCE-347? BCE 1998. The Allegory of the Cave. *The Republic*.

[17] Rhodes, B.J. 2000. Margin notes: Building a contextually aware associative memory. *Proceedings of the 5th international conference on Intelligent user interfaces* (2000), 219–224.

[18] Stamford, B. 2000. Curing Health and Medical Coverage. *American Journalism Review*. (Apr. 2000).

[19] The Healthcare Blogger Code of Ethics and HIPAA: *http://thesocialmedic.net/2011/02/the-healthcare-blogger-code-of-ethics-and-hipaa/*. Accessed: 2015-08-13.

[20] Ward, S.J.A. 2006. *The Invention of Journalism Ethics: The Path to Objectivity and Beyond*. McGill-Queen's University Press.