Rich Interfaces for Reading News on the Web

Earl J. Wagner ewagner@u.northwestern.edu
Jiahui Liu j-liu2@northwestern.edu
Larry Birnbaum birnbaum@cs.northwestern.edu
Kenneth D. Forbus forbus@northwestern.edu

Northwestern University, Evanston IL USA

ABSTRACT
Using content-specific models to guide information retrieval and extraction can provide richer interfaces to end-users for both understanding the context of news events and navigating related news articles. In this paper we discuss a system, **Brussell**, that uses semantic models to organize retrieval and extraction results, generating both storylines explaining how news event situations unfold and also biographical sketches of the situation participants. We generalize these models to introduce a new category of knowledge representation, an *explanatory structure*, that can scale up to include information from hundreds of documents, yet still provide model-based UI support to end-users. An informal survey of business news suggests the broad prevalence of news event situations indicating Brussell's potential utility, while an evaluation quantifies its performance in finding kidnapping situations.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Authors Keywords
Design, Human Factors, Explanatory Structures

INTRODUCTION
People read the news to learn about what is happening in the world. In addition to reading traditional newspapers, people access news articles on the Web through desktop computers, notebooks and even mobile phones. They arrive at articles through news aggregators such as portals and collaborative recommendation sites, or even just emails from friends. But what happens when they want to find out more about the things discussed in an article? In contrast to all of these developments, little has changed in how people explore the context of the news they read. As one reader observed in a recent ethnographic study of young news readers conducted for the Associated Press, "if you want background, it's up to you." [3]

IUI'09, February 8–11, 2009, Sanibel Island, Florida, USA.

The Need for Background to News
News articles commonly discuss events in detail and relevant information about the people and organizations involved in the events. But a reader may not have heard about one of the individuals introduced in the article and want to see a biographical sketch. Alternately, she may see an article describe an organization as having the "second highest revenues in its industry" and wonder what those revenues are exactly, and for what goods and services. Or given that an event just happened, she may want to know what events happened before that led to its occurrence. In focusing on details that are new or have changed, however, articles often leave out contextual information like this. As another reader told AP, "news [today] is not the full story, but more like a preview–it's kind of annoying sometimes. I don't like to get bits and pieces of information." Rather than being a shortcoming of the news format, however, we see this as an opportunity for software to offer a richer user experience for navigating the context of news.

This problem of exploring the context of information appears more broadly, as people browse the Web not only to search for specific facts, but also as part of "`building a picture' of an organization, topic or person." [18] Even more than when just browsing the Web, however, the need for a "big picture" view is particularly acute when reading news. Another reader explained to AP that he "does not want to be fed bits. I want to know all the details at once." However, the nature and specific kinds of big picture views that might provide information gatherers with "all the details at once", and how software might be constructed to support their elaboration, has not received nearly as much attention as search more narrowly construed.

The Contexts of Situations and Participants in the News
Consider the case of a person reading about a rescue operation that freed a kidnapped Colombian politician. A typical news article covering this event provides details of the rescue and some information about how long she had been held. It mentions the kidnappers and some information about the rescue and status of other rescued

hostages. Toward the end, it refers to the final negotiations preceding the rescue.

Although it mentions some previous events in the kidnapping, the article does not provide a high-level view of how it unfolded over time. Its discussion of the participants in the situation assumes that the reader is already familiar with them. To learn more the events and participants, the reader must manually find relevant news articles or informative pages on the Web. He must identify identifying terms including entity names and event keywords. Then he must cut-and-paste them into a search engine and sort through search results to find relevant pages and assemble an account of what happened. These steps make for an inconvenient process familiar to anyone who reads news on the Web.

Context from Structured Presentations of Information

In looking for more information about a person or organization, he may arrive at a Wikipedia page with "trading card"-style infoboxes listing essential information. These infoboxes answer clusters of questions about entities. For example, for a person they provide information on:

- Who is this person?
- What positions has this person held?
- What groups has this person been associated with?

Although the same information may be found scattered among many Web pages, it is useful to see it gathered all in one place. This serves at least two important purposes. First, it provides a "gestalt" allowing him to easily take in all of the information to know what it means and how it is related. Second, it allows him to notice any details that could be helpful in making sense of the situation. A reader can't simply ask a search tool to "show me what's most relevant or interesting in making sense of this situation", but in availing himself of a structured presentation of related information in a conventional form, he can more easily orient himself.

Thus one possibility for better support for understanding the context of news articles is in providing easier access to biographical sketches of event participants. However, the context of news articles involves not just the named entities, but also the events and, importantly, the causal relationships among the events. In reading about the kidnapping, the reader may want to know more about the events that preceded it, in other words, all of the events that make up "the kidnapping" of the individual. We say these events make up the kidnapping *news situation*, where by news situation we mean its limited sequence of causally-related events covered in the news.

For example, the dismissal of a lawsuit, if it occurs, will follow the filing of that lawsuit, and both are part of a particular lawsuit-type news situation. The individual events constituting a situation are situation events, or just events, and distinct situations of the same type, "lawsuit", do not necessarily involve the same events, just as different lawsuits may have different outcomes. Within a situation type, the events may have ordering relations, and the occurrence of one event can prevent another; the settlement of a suit will not occur if it has already been dismissed.

A reader's expectations for how a situation will unfold includes relationships like these and, as such, they contribute to an event's situational context. This context gives rise to another cluster of questions, including:

- What happened in this situation?
- How did it start? How did it end?
- Who are the participants involved?
- What other similar and related situations have these participants been involved in?
- What happened in the other, related situations referenced in this article?

Just as Wikipedia's biographical sketch infobox provides the essential information of an entity through a structured presentation, a storyline for the situation in terms of events can help in answering questions like these. This storyline view could organize the milestone events that make up the kidnapping from the original abduction to the current release. It could also list the participants and their roles and link to biographical sketches.

VIEWING A SITUATION IN THE NEWS WITH BRUSSELL

Let's return to the case of learning more about the rescue event. Suppose the person were using Brussell, a research system we've developed that provides direct software support for accessing informative views of the overall kidnapping situation and its participants. Then, rather than interacting with the article at the textual level by selecting keywords to search with, he could simply right click on a phrase describing the event (see Figure 1). We call a phrase like this one a *situation reference*.

This reveals a context-menu with questions specific to the situation being referenced. To find out more about it, the user selects "What happened in this kidnapping?" from the context-menu, which loads in the browser a storyline for the kidnapping including milestone events (see Figure 2).

With the storyline view, the user can see that the overall kidnapping situation began with the individual's abduction in February of 2002 and continued more recently with the release of a videotape of the hostage and an appeal for the hostage's release both occurring in late 2007. Clicking on the release event updates the toolbar to show date and location information for the event and loads an article about the release (see Figure 3). From this article's lead he can see that planning for the rescue began several months ago.

Referring to the timeline, he realizes that that was shortly after the last appeal for release.

The article lead also mentions the kidnapper group and he'd like to find out more about it. To do so, he clicks on its name in the toolbar, which loads its biographical sketch view (see Figure 4). In addition to details about the group, this includes all of the situation events it has been involved in, and images from articles about those situations. Within this view, details and images link to the article from which they were extracted, enabling the user to verify them and learn more.

We expect that readers will access Brussell's big picture views in two kinds of circumstances: one, when reading an article primarily about a situation event and wanting to know more, as in the example. In other cases, an article largely about one event refers to another in a single sentence. For example, an article about Microsoft's offer for Yahoo states, "Yahoo recently acquired Zimbra", and the user may want to find out how and why that occurred to better understand the context of the offer in the article.

Although the example shows how structured presentations can be helpful, we don't expect that users would use Brussell to view the situation context of events in every article they read. Actual usage would depend, of course, upon whether the reader is simply skimming the news or doing in-depth reading. For example during a session of reading several articles over the course of an hour, a user might want to view the situations for many of these, perhaps one out of every four or five.

PRESENTING INFORMATION THROUGH EXPLANATORY STRUCTURES

In the example we saw two kinds of structured presentations. The first, a situation storyline view, resembles an ordinary timeline with a sequence of events oriented in time. The second, a biographical sketch view, presents essential details of a situation participant, the merged storylines of all of the situations involving the participant, and images of their participation in the situations. With the example suggesting these big picture views can be helpful, it is important to ask, where do they come from?

In fact, content-specific information presentations like these are automatically generated from models through information retrieval and information extraction. Systems taking a similar approach include vertical search engines. ZoomInfo presents resumes of individuals generated employment information that it automatically extracts from pages on the Web. [21] CiteSeer provides a "product page" for computer-science publications freely available on the

Figure 1. Asking about a situation referenced in the article.

Figure 2. Viewing milestone events for the selected situation.

Figure 3. Viewing an article on the selected situation event.

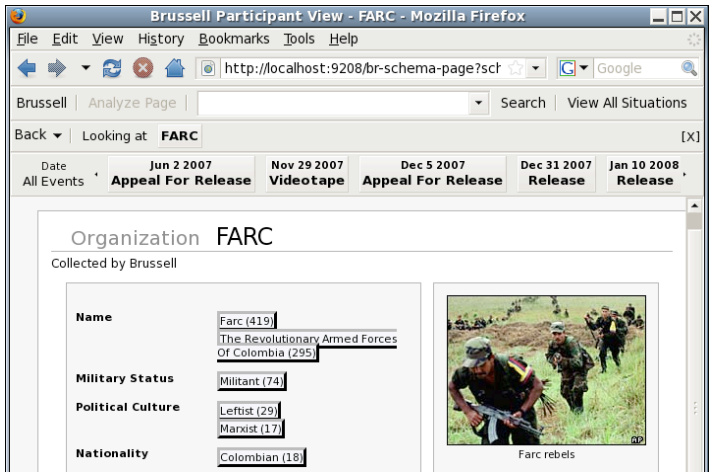

Figure 4. Viewing the biographical sketch for a participant.

Web by extracting their abstracts and authorship and citation information. [5]

In this paper we detail the contribution of the Brussell system in going a step farther than these websites by not only generating content-specific views, but also enabling users to access these views within their web-browsing task context. The situation and biographical sketch models are created and presented using similar mechanisms and we call them both *explanatory structures*. An explanatory structure, or ES, is a content-specific template featuring semantic constraints that can guide information retrieval and extraction to provide a conventional information presentation linked to the user's task. In addition to the kidnapping situation type and organization biographical sketch, Brussell supports situation explanatory structures for legal trials and corporate acquisitions, and biographical sketches for persons and groups of people.

Having seen an example of the kind of direct support for situations and participants these views can provide, we next turn to the features of explanatory structures and how they drive Brussell's functioning. Then we focus on Brussell's situation models and establish that it is reasonable to expect them to be common in the news and thus content-specific support for interacting with situations is warranted. We also quantify Brussell's performance in extracting kidnapping situations from news articles. We conclude by looking at background work and future directions.

TO SERVE EXPLANATORY STRUCTURES

Beginning with the properties of explanatory structures, we see how Brussell creates ES instances to provide to users. We then see how they impose functional constraints on Brussell's architecture and drive its operation.

Properties of Explanatory Structures

Centered on an aspect of a focal entity or entities

An explanatory structure is about a specific *thing*, whether a conventional named-entity such as a person, organization, or product, an intellectual product such as a legal trial, legislative act, or research project or, in the case of Brussell's situations, a sequence of events centered on a specific participant or set of participants. In the example above, the focal entity is the kidnapped individual.

Explanatory structures do not exhaustively collect all information about the entity, however, but rather present a specific and well-defined *aspect* such as all of a person's research publications, or the events in a situation, or the essential biographical details of an individual or group.

Conventional genres of content-specific information presentations

Explanatory structures act as familiar big picture views supporting easy orientation by presenting information as a gestalt. The biographical sketch is an often-used format for presenting essential information about a person or organization. A situation storyline appears as a timeline, with all of associated expectations of linearity, ordering, and the relevance and notability of selected events.

The slots of an explanatory structure differ from search results in working together to support understanding of a topic. The results of a search are unrelated and, in essence disjunctive. Either one result is what the user is looking for or, if not, then perhaps another one is. In contrast, the information in an explanatory structure is conjunctive; the whole is greater than the sum of its parts and each element contributes to the overall meaning. By contextualizing the information it contains, the explanatory structure itself also contributes to the meaning if its elements, by indicating what happened before and after entries in resumes and timelines, for example.

Support rich interaction within the user's task context

Explanatory structures are designed to be easily accessed from the user's current task context, including the browser as in the example. To support this access, the ES includes indicators to automatically recognize relevant references within documents the user is reading, without requiring the user to select or search for them individually.

Knowing the affordances in advance makes it possible to provide richer interaction such as inspecting situation and participant references directly and selecting a choice from a semantic menu. These techniques can even subtly provide relevant information new to the user. For example, the identity of the kidnapper appears in the context-menu, even though it may not be in article.

Finally, they organize at a high level the entities in user's current task and allow for easy navigation and traversal among relevant documents.

Authored knowledge structure types with semantic constraints and typed fields

Explanatory structures consist of a frame structure with slots and values that fill the slots. Each slot is constrained to hold values of a certain type and quantity. Brussell's biographical sketch ES type limits certain kinds of information to be extracted when reading entity references. It includes a person's age, nationality and employer, and an organization's industry, for example. Similarly, a situation ES specifies the roles that participants may play and imposes type restrictions on these roles. For example, an organization can't be kidnapped, although a person or group of people can be.

Some slots may not be filled and thus not presented. Other slots may not be revealed because their existence conflicts with shown information, as determined by semantic constraints associated with the slots. For example, if a kidnapped individual has been released, incorrect information that he was killed would not be shown. The

situation model specifies the possible milestone events and semantic constraints holding among them including their ordering and which events are mutually exclusive

Meta-information drives finding and creating new instances, and extending existing instances

To find and extend instances of explanatory structures, Brussell uses indicators and extractors associated with the ES and its slots, respectively. The ES type specifies keywords used to retrieve relevant documents. In the case of a legal trial, this includes "trial" and "*suit".

Brussell uses text pattern recognizers associated with the ES and slots to find references to situations and participants and extract information to populate the slots. It repurposes these recognizers to find references in the current web page.

Record provenance of information as evidence

Finally, as part of the process of retrieving pages and extracting information, Brussell records the sources of information as well as the information itself. This allows the user to inspect the evidence supporting the information he sees. If a detail seems unexpected to the user, or if it appears that the page might provides further interesting details, the user can access the page directly to learn more.

How Explanatory Structures Drive Brussell's Operation

We now turn to the question of how to support these features of explanatory structures. Several important challenges arise. First is the question of where the ES types come from. A system must possess a pre-existing library of ES types, and each must be elaborated sufficiently such that they can be instantiated and managed with little or no supervision. Creating these types automatically and populating them with extractors remains an area of future work and we discuss this more later. Second is the issue of where the ES instances come from. In order to provide anticipatory support within the user's task, Brussell runs automatically and, in reading documents, it knows when to create a new ES instance and when to merge new information with an existing instance. Third, the system must effectively reconcile erroneous and conflicting information. Finally, the system must employ techniques to limit the distraction from any incorrect information that remains.

Knowing when to instantiate and when to merge

In reading through source material, and finding references to situations and entities, when does the system create a new ES instance? Brussell distinguish instances based on the focal participant, specified by the "profile" of the ES. For situations, this is the identity of the kidnapped individual, or the combination of the plaintiff and defendant in a legal trial. An unsolved issue with this approach is that multiple situations with the same profile are merged into the same instance, as with multiple lawsuits between feuding companies.

Intelligently handle errors and reconcile conflicting information

A well-known problem with building and manipulating explicitly represented models is that of handling errors and resolving conflicting information. A source may simply provide incorrect information. To some extent this can be partly ameliorated by pre-selecting the sources of information but, for example, often a breaking-news article features incorrect information that is later amended. Or information in an article may be correct, but presented idiosyncratically and, as a result, extracted incorrectly.

We regard conflicts as the main impediment to scaling knowledge representations to the Web. It is reasonable to expect that correct information will be stated more often than incorrect information, however, especially over time as consensus develops over the details of events that originally may have been hazy. So Brussell implements a voting algorithm to resolve error due either to incorrect article information or faulty extraction. After filtering out duplicate articles and sentences from the input pages it reads, it treats every textual appearance of a fact or reference as a vote and simply counts the number of textual references to an event, event fact or biographical detail. No votes are weighed more than others.

Voting is used to resolve conflicts among structure values as well as text values for slots:

- At the top-most level, to select which events actually occur within a situation
- For facts about events including dates, locations and monetary amounts
- Concerning biographical information about situation participants such as names, nationalities, person occupations and group sizes

Brussell uses type-specific techniques for reconciling differing structures and, further, it uses vague accounts as support for specific information. For example, in determining the date of an event, "last month" may be counted as a vote for "April 20th," but not vice versa. Similarly, the description that a kidnapper was "a group of militants" supports "Al Qaeda in Iraq" over "US troops".

Saving textual supports for extracted information serves an additional purpose: to justify how conflicting information has been reconciled.

Hide incorrect information

Since the system is extracting information with minimal supervision, it needs strategies that select correct information and eliminate, or at least hide incorrect information. It is acceptable, and even desirable, for a user

to be able to explicitly request an alternate account or "minority report", however.

It is assumed that Brussell will instantiate situations and participants promiscuously so, for example, it doesn't show all references in a page, and instead reveals them only when the user moves the mouse over one. This works well when an entire invalid situation or participant can be hidden, but if participant is involved in a valid situation or a single event in a situation is misread, incorrect information can "piggy-back" onto a valid participant or situation, e.g. "kidnapping of President George Bush" or mentions of a spurious negotiations event in an otherwise correct situation. Further, the problem of negated and hypothetical situation events mentioned in the news remains unresolved.

Brussell's Architecture

Brussell consists of a Firefox browser plugin and server software, which may both run on the same computer. When the user loads a new page in the browser, the browser software retrieves any cached entity and situation references for the page. If the server hasn't already analyzed the page, it renders the button with the label "Analyze Page". A user can view references in news pages, as in the example, or can request the analysis of any web pages, such as blog posts, by clicking on the button.

The back-end system requires manually created situation model types (inspired by scripts) and currently supports kidnappings, legal trials and corporate acquisitions each of which has multiple possible outcomes and on the order of 8-12 possible events. The system runs daily to retrieve news articles from several English-language news websites via RSS feeds and store them in a Lucene index. [2] After retrieving new articles, it then queries the index to collect the new articles with keywords associated with the situation types it supports and reads through them to create and extend situation instances. These instances include a single reference up to several hundred if they are well publicized. Using an index of saved news articles rather than searching the Web directly allows Brussell to show the source of extracted information even if the article is removed from the news website.

Brussell uses GATE [8], a standard open-source information extraction system to extract situation information including event references, dates and locations, and entity details such as person names and occupations or organization names and nationalities. Extracting this information allows references such as "the British journalist abducted last year" to be resolved to a particular kidnapping.

THE PREVALENCE OF NEWS SITUATIONS IN BUSINESS NEWS

In the example we saw the support Brussell can provide in reading about kidnapping situations. It's not necessarily obvious that these sorts of stereotypical news situations are prevalent, however. Determining how often they appear would establish an upper bound on the coverage of a system. Obviously, the system would not be useful if it could only provide a richer interface for interacting with a tiny fraction of the news articles a user would read. On the other hand, the system could be useful if it potentially provides support for interacting with many news articles, for some if not many domains. To determine whether this is the case, we performed an informal investigation into the frequency of situation references in news. We selected a particular domain in which we expected them to be particularly common and thus the tool to be especially useful, business news.

Experiment Setup

We randomly selected 100 English-language business news articles published on the Web from April 2005 to August 2008. Articles were retrieved from nine prominent English-language news sources: ABC News, BBC News, Los Angeles Times, The New York Times, San Francisco Chronicle, San Jose Mercury News, USA Today, The Washington Post, and Yahoo! News (which features news from the AP, Reuters and other wire services). Since the hierarchical organization of news sites is often reflected in the URLs of their articles, to determine whether an article is within a business section or otherwise likely to be about business news, we looked for the word "business" in its URL. A person read the text in the article's title, lead and content and manually annotated any situation references.

To be considered a situation reference, we required that the phrase include at least one event-related verb and one or more named entities. So "Roche offered $85 per share for Genentech" would be annotated as an "acquisition offer" reference. Because Brussell performs a simple form of situation-based anaphora-resolution to merge vague references such as "Roche's offer" and "the bid for Genentech", these would also be included in the assumption that disambiguating references appear elsewhere in the article. A reference featuring no or minimal identification of a participant such as "the bid for the company" would not, however.

In addition, we limited our focus on situations that would be "interesting" to the user. That is, we focused on cases in which the user would conceivably want to learn more about a situation by seeing its storyline view. Consider a quote in an article by an individual "Mark Corallo, a spokesman for Coventry". Even though a person's employment at an organization is conceivably a situation that includes events such as the person's hiring, possible promotions, and eventual departure, we would not expect the employment of a spokesperson by a company to be something the user would be interested in learning more about. Further, it is not likely that this person's employment would be considered "newsworthy" and covered in detail by further news articles, with the result that no situation could be

created for it. So the employment of an individual who was simply quoted was not annotated.

Frequent fluctuations in quantities such as stock prices, federal reserve rates, interest rates, inflation, approval ratings and survey results can not be accommodated within Brussell's situation models so these were also excluded. Quarterly corporate earnings results and forecasts were also excluded because though they could conceivably be considered situational, with its current architecture Brussell cannot currently distinguish among them.

Experiment Results

One graduate student annotated the situation references in the 100 business news articles finding that 58% had at least one situation. 42% had none and consisted of articles about, for example, earnings forecasts and reports and "lifestyle" issues such as the best cities for recent college graduates and how to live more environmentally-consciously.

Looking at just the articles with references, the histogram in Figure 5 shows how many articles have different quantities of references with the mode being 3 references in 15% of the articles. Articles with references had a mean of 4.1 references and a median of 3 references.

The results for references broken down by the most common situation type appear in Figure 6. Events in three situation types appeared in more than 10% of the articles: 55 employment event references appeared in 19%, 51 corporate acquisition events in 16% and 33 product lifecycle events in 15%.

The most common employment event was a "hire" event, which appeared in most articles referencing employment transitions with 20 references in 11 documents. A typical hire event reference was "Mike Burbach, who became editorial editor of The Pioneer Press three weeks ago..." A typical example of an "offer" reference, the most common acquisition event with 8 references in 4 documents, is "Warner last made a formal approach earlier this year, a 2.1 billion pound offer..." Finally, the most common product

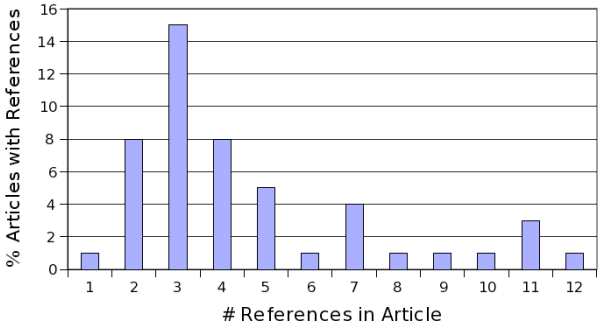

Figure 5: Histogram with number of references per article in business news articles.

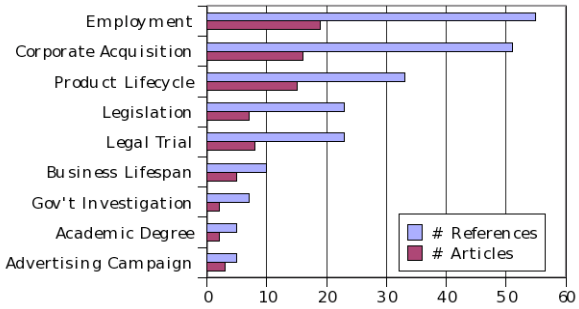

Figure 6: The prevalence of references to situation types in 100 business news articles.

lifecycle event was a "release" event reference with 8 references in 7 documents. A typical example would be "iPhone, the company's new smartphone that Jobs unveiled at its Macworld conference last week."

From this informal survey, we can see that situations are fairly common and it is reasonable to suppose that a tool like Brussell could often be used to support direct interaction with situations

EVALUATING BRUSSELL'S PERFORMANCE IN EXTRACTING SITUATIONS

Two further issues arise regarding Brussell: how well it performs in extracting situations overall and whether its performance improves as it reads more. We place greater emphasis on the second concern because we see the contribution of the system not in the quality of its extraction mechanisms per se but rather how well it can present information about prominent situations.

We can tell how well it performs overall by observing its performance in extracting the following:

- Whether a situation occurred
- The events within a situation and their dates
- Biographical details of situation participants

To determine whether the system performs better when extracting from more articles, we compare situations referenced many times with those infrequently referenced. For testing and training, we looked for definitive sources of information about situations of a particular type that consist of multiple events. Using published lists of kidnapping situations, we evaluated the performance of the system on a corpus of news articles to answer these questions.

Experiment Setup

We both trained and tested Brussell on collections of kidnappings of foreigners in Iraq since the beginning of the US invasion in March 2003. The training collection was published by the AP and included 35 kidnappings through October 2004. To test the system, we turned to a more

recent Wikipedia page listing 164 kidnappings through August 2008. [19] For example, in the section for Australians, the entry "Douglas Wood, construction engineer, kidnapped April 30, 2005, and freed June 15, 2005," is represented as a kidnapping situation consisting of two dated events, about a victim with a name, nationality, and occupation. Because Brussell cannot distinguish situations involving vaguely identified participants, 35 kidnappings of unnamed individuals (such as "an Iraqi translator") and groups of individuals (such as "two French journalists") were not used.

Brussell has an index of approximately one million articles. Nearly 70,000 of these include a kidnapping term: "kidnap*", "captur*", "abduct*", or "hostage*". To focus on the cases from the Wikipedia list, we narrowed this set to the 24,687 articles containing both the complete name of a kidnapped individual in the list and a kidnapping term.

We sought to test the system's functionality using criteria somewhat different from tradition information extraction evaluations. Because Brussell is aimed at providing a specific user experience we sought to test the functionality it would have in a "real world" context. Assuming that the news Brussell downloads and indexes is representative of national news in general, we wanted to characterize the level of support a user can expect from the combination of Brussell and a news corpus this size. A traditional evaluation of event-extraction software might involve comparing the situations Brussell extracted from the test corpus with all of the situations, situation events and facts and biographical details it could potentially have extracted.

Our argument for Brussell's contribution is not in the sophistication and thoroughness of the extraction it performs, however, and is rather based on quantifying the level of detail in situations a user can expect to access for a news corpus of this size. Rather than noting how completely the situations were described in the articles in the index, we assumed the situation was completely described and assessed the system's performance in extracting the complete situation if the individual appears in any kidnapping-related articles at all.

Experiment Results

Of the 164 Wikipedia kidnapping situations involving named individuals, 135 or 81.7% were present in the news corpus. Brussell found 101, or 74.8% of the 135 situations in the test collection. That is, Brussell found at least one situation event for 74.8% of all of the Wikipedia situations for which there was at least one article in the corpus with a complete name and kidnapping terms.

Overall, Brussell found 48.9% of the biographical details of situation participants, 62.8% of the situation events, 37.3% of the event dates and 41.0% of event locations. Because the test collection didn't specify all of the correct events and facts that Brussell could recognize, we didn't measure the precision and the number of false positives.

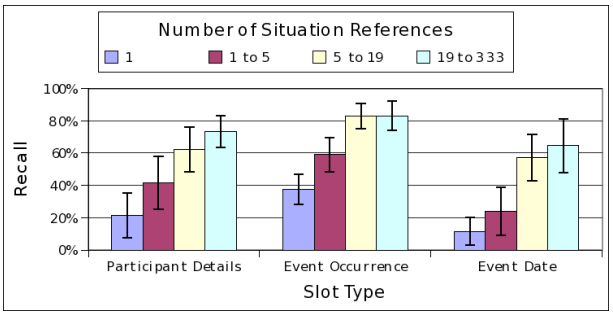

Figure 7: Reading more references to event and participant slot values significantly increases recall.

To get a better sense of how Brussell's performance varied with the number of references to a situation, we split the results into quartiles based on the number of references (see Figure 7). The mean recall of participant details for the fourth quartile is 73.5% versus 21.6% for the first quartile. The mean recall of event occurrence for the fourth quartile is 82.7% versus 37.5% for the first quartile. The mean recall of event dates for the fourth quartile is 64.7% versus 11.5% for the first quartile. For each of these slots, the recall for the fourth quartile is statistically significantly better than the first ($p < 0.001$, t-test). These results suggest that aggregating the results of extraction from multiple references can improve performance.

Limitations

It is important to mention the issue of false-positives in recognition. Brussell recognized over 10,000 spurious situations in the kidnapping articles. Although this seems extremely high—a 99% false positive rate—as we argue above, careful design of the interface for revealing references can minimize the degree to which they distract the user.

BACKGROUND WORK

Innovation on news websites

Many news sites recognize entity references within article pages and provide links to further information. Often these link to pages on their own sites. On some sites, these links are to previous articles about the same topic. On others they link to advertising for generic terms or are inserted into news article pages in optimizing for search engines.

Many news sites also provide some background through "related articles". These are either manually added or based on term-frequency but are typically not updated as new events occur, however. They save the user from having to take the step of searching for relevant articles prior to this one, but not the step of sorting through the

articles and assembling the big picture. They also usually present articles only from the same news site.

Software support for reading news

Previous research focused on extracting information from both single and multiple news articles. Some of the approaches in reading single news stories use the script conceptual formalism for story understanding, which is the basis of our approach for modeling user expectations for situations as well. Brussell's situation types and instances are simplifications of Schank and Abelson's scripts. [17]

Single News Article: Story and Event Extraction

Early work in extracting information from single stories includes systems developed by Schank and his research group at Yale. SAM uses scripts to guide a deep analysis of a news article in order to provide a summary and answer questions about the events it covers. [7] Frump, also using scripts, performed a more shallow analysis to read through news articles rapidly. [9] Like Brussell, it was connected to an online source of news, in its case the UPI newswire.

Extracting event information using templates from single news articles is the focus of work in the Message Understanding Conferences [11].

More recent work on extracting formal knowledge from news has focused on populating the Semantic Web. SemNews [12] extracts structured representations from news retrieved via RSS feeds. Unlike Brussell, however, its emphasis is generating representations in the form of RDF triples rather than presenting views to the user.

Multiple News Articles: News Summarization

Techniques in text summarization have been used to merge and reduce the information in multiple documents to present the user with a natural language summary. NewsBlaster [14] and NewsInEssence [15] cluster and summarize similar articles, while NewsJunkie [10] indicates the differences in new articles.

Multiple News Articles: Topic Detection and Tracking

Selecting and presenting all and only the news articles associated with news topics is the focus of Topic Detection and Tracking (TDT). These research systems typically represent events as term-vectors, and classify and cluster news articles using these event representations. [1]

In contrast to both TDT and news summarization systems, presenting explanatory structures requires that a system "knows what it knows" by selecting and labeling milestone events in accordance with user expectations.

There's a deeper issue at play here, however. We argue that the context of a user's news reading task has a structure, and supporting that context is not simply a matter of providing more information. In particular, there are patterns to kinds of information people want and they are reflected in the conventional presentations of information we've already seen. TDT research implicitly acknowledges this by making a semantic commitment and assigning documents to a temporal extent for an event. Without a model of events, however, TDT systems are unable to offer the rich UI that explanatory structures enable by integrating with the user's context and presenting views in accordance with the user's expectations for how a situation unfolds.

Query-free Information Retrieval

Other query-free information retrieval systems for end users include Letizia [13] and Watson [6]. These systems search the Web to find documents relevant to a user: Letizia by following the links of the currently open web page, and Watson by modeling her current task in the browser or an open Microsoft Office document.

As we noted, vertical search engines such as ZoomInfo and CiteSeer offer content-specific presentations of information. Other websites also offer popular content-specific views, such as IMDb and DBLP. Integrating these views with the user's browsing task context could provide a rich interface as Brussell does.

FUTURE WORK

The most noticeable improvement to Brussell would come with support for many more situation types. Adding a new type consists of specifying semantic constraints, retrieval keywords and extraction patterns. Authoring the patterns is the most time-consuming portion by far, though this could be automated through bootstrapping techniques. [16]

Brussell could also improve by disambiguating situations with the same focal participants, or multiple events of the same type within a situation. In some kidnappings, there are actually been multiple negotiation events and the event as presented merges information from each of them. A clustering approach may be helpful.

In some cases, further sub-structures within an explanatory structure should be recognized and extracted, as with the multiple jobs held by an individual. These would naturally link to situation views for the job transitions.

Awareness that slots are empty or unverified could trigger goal-driven, "autonomous" search and extraction. [20]

CONCLUSION

Many researchers have put forward the goal of integrating the Web with high-level semantic models to provide more goal-oriented interfaces. Some, including those working as part of the Semantic Web effort, anticipate providing this user-level functionality by having authors annotate their web pages using standardized domain-specific logical annotations. [4] In other words, this effort is aimed at providing smarter interactions with web content by constructing the web out of explicit logical representations.

Rather than bringing the Web to semantics, however, we propose bringing semantics to the Web. With Brussell, we have presented a system that enables users to interact directly with entities and situations referenced in web pages in order to navigate the context of the news webpages they read. Brussell uses standard IR and IE technologies integrated with semantic models in explanatory structures to anticipate user questions and provide high-level views that match user expectations. The prevalence of situations like the ones it supports and its performance in extraction situations both point the way to further research in rich, content-specific interfaces for reading news on the web.

ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under grant no. IIS-0325315/004. We thank our colleagues Chris Riesbeck and Francisco Iacobelli.

REFERENCES

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. 1998. Topic Detection and Tracking Pilot Study: Final Report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. San Francisco, CA, pp. 194-218, Morgan Kaufmann Publishers, Inc.
2. Apache Lucene, http://lucene.apache.org/java/docs/
3. Associated Press. 2008. A new model for news: Studying the deep structure of young-adult news consumption. http://www.ap.org/newmodel.pdf
4. Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web, In *Scientific American* 284(5):34-43 (May 2001)
5. Bollacker, K. D., Lawrence, S., and Giles, C. L. 1998. CiteSeer: an autonomous Web agent for automatic retrieval and identification of interesting publications. In *Proceedings of the Second international Conference on Autonomous Agents*.
6. Budzik, J., and Hammond, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User interfaces (IUI '00).*
7. Cullingford, R. 1981. SAM. In Schank, R. C. and Riesbeck, C.K. (Eds.), *Inside Computer Understanding*, (pp. 75--119). Hillsdale, NJ: Lawrence Erlbaum.
8. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July 2002.
9. DeJong, G. F. 1982. An Overview of the FRUMP System. In Lehnert, W. G. and Ringle, M. H. (Eds.), *Strategies for Natural Language Processing*. Hillsdale, Erlbaum, pp. 149--176.
10. Gabrilovich, E., Dumais, S., and Horvitz, E. 2004. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th International Conference on World Wide Web (WWW '04).*
11. Grishman, R. 1997. Information Extraction: Techniques and Challenges. In Pazienza, M. T. (Ed), *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology.* Lecture Notes In Computer Science, vol. 1299. Springer-Verlag, London, 10-27.
12. Java, A., Finin, T., and Nirenburg, S. SemNews: A Semantic News Framework. 2006. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06),* February 2006.
13. Lieberman, H., 1995. Letizia: An Agent That Assists Web Browsing. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.
14. McKeown,K.R.,Barzilay,R.,Evans,D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. 2002. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In *Proceedings of the Second international Conference on Human Language Technology Research* (March 2002).
15. Radev, D. R., Blair-Goldensohn, S., Zhang, Z., and Raghavan, R. S. 2001. NewsInEssence: a system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of the First International Conference on Human Language Technology Research* (San Diego, March 18 - 21,2001).
16. Riloff , E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*
17. Schank, R. C. and Abelson, R. P. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
18. Sellen, A. J., Murphy, R., and Shaw, K. L. 2002. How knowledge workers use the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02).*
19. Wikipedia. “Foreign hostages in Iraq.”. http://en.wikipedia.org/wiki/Foreign_hostages_in_Iraq
20. Wu, F. and Weld, D. S. 2007. Autonomously Semantifying Wikipedia. In *Proceedings of the Sixteenth Conference on Information and Knowledge Management (CIKM-07),* November, 2007
21. ZoomInfo.com. http://www.zoominfo.com/