Using Lightweight NLP and Semantic Modeling to Realize the Internet's Potential as a Corporate Radar

Alex Kass and Christopher Cowell-Shah

Accenture Technology Labs
1661 Page Mill Road
Palo Alto, California 94304
alex.kass@accenture.com, c.w.cowell-shah@accenture.com

Abstract

Many executives wish that they had a more systematic means of using technology to help them identify external events that represent evidence of potential threats or opportunities. The *Business Event Advisor* is a prototype corporate radar kit that addresses this need. It can be used to create customized solutions to monitor the external business environment in which a company operates. It exploits Internet-based information sources to help decision-makers systematically detect and interpret simple patterns of external events relevant to their business concerns. To accomplish this, it integrates text mining components and a simple inference mechanism around a semantic model of a company's business environment. Applications built with the system can produce structured descriptions of events that may be relevant to a given company, and can then aggregate those events around its model of the company's business environment and suggest what impact(s) these events might have on that company. We discuss our motivation for creating this prototype, the architecture of the system, the kinds of information it can provide, the kinds of models it requires, and possible directions for future research (including use of pattern matching to auto-generate elements of our system's models).

Overview

The vast stream of information constantly broadcast on the Internet is destined to revolutionize business intelligence (BI). Many executives wish that they had a more systematic means of using technology to help them identify, as early as possible, the external events that represent potential threats or opportunities to their companies. When you can nip threats in the bud, or seize opportunities before they grow stale, you win. When effectively harnessed, the frequently-updated, easy-to-access information available on the Internet could greatly extend the limited, mostly inward-looking scope of enterprise BI systems. By providing decision-makers within an organization with a means to systematically monitor a greatly expanded range of weak signals about the goings on outside the organization, corporate radar systems of this sort have the potential to provide management with a dramatically enhanced understanding of the competitive ecosystem and the business-relevant implications of events in that ecosystem. However, the potential of the Internet as a source for outward-looking BI, or corporate radar, is still largely untapped because executives do not yet have the kinds of tools they need to exploit it effectively.

In this paper we'll discuss the Business Event Advisor, a prototype we've developed to fill this need. Existing tools in common use for consuming Web-based information include Web browsers, RSS readers, and various keyword-based aggregation tools such as Google Alerts. The Business Event Advisor adds what we call a more *business-aware* twist to these technologies. By exploiting a simple set of semantic models that encode the kinds of knowledge of the structure and dynamics of a company's competitive ecosystem that an industry analyst can provide, our prototype produces a business-aware system for exploiting the Internet, producing a kind of corporate radar display that includes descriptions of events that may be relevant to the user's business, along with possible implications of those events. We believe it is only through such business-aware technologies that the true potential of the Internet as a tool for systematically detecting evidence of business-relevant events can be achieved.

Our prototype includes interactive model-building tools to allow business analysts to create models that drive the system. The models include representations of the entities and relationships that make up a company's competitive ecosystem, the event types that can impact that ecosystem, and the inference rules representing the implications that these events can have.

It also includes a run-time processing engine that uses those models to drive the detection and interpretation of business relevant events. Event detection involves lightweight statistical NLP to translate unstructured data (such as news reports) into structured event descriptions. Interpretation involves a rules engine that infers potential business implications from appropriate patterns of detected events. Users of the system are thus given a systematically processed view of events that are being reported on the

Web, as well as important events that although not being explicitly reported, represent reasonable inferences based on the weak signals that have been detected. In other words, various sources of (often unstructured) Web-based information are continuously scanned, and information relevant to a customer's business is translated into structured descriptions. The system then determines implications that the detected events might be evidence for. (Figure 1 depicts the basic conceptual overview of our system; elements shown in bold outline represent components that are implemented in the prototype, while those without outlines represent those that have not yet been implemented.)

Inspiration: Insight Captured from the Web

An episode recounted in a recent *Fortune* cover story (Vogelstein 2005) about Microsoft and Google helps illustrate two points: 1) how the Web can provide the modern corporation with important sources of insight, and 2) the reasons why the kinds of insight that are theoretically now accessible via the Web are not more consistently exploited. The episode that's relevant to our discussion involved Bill Gates' use of the Web to enhance his insight into the serious nature of the threat that Google could pose to Microsoft. While "poking around" on the Google corporate website, Gates decided to take a glance at the listing of open positions. When he did, he was gripped with a stark and worrisome realization. Google, which he had thought of as essentially a search company, was recruiting for all kinds of expertise that had nothing to do with search. In fact, Gates noted, Google's recruiting goals seemed to mirror Microsoft's. Google was not really a search company, as he had previously pigeonholed it; it was a software company, and it seemed to be planning a future in which it occupied much of the turf that Microsoft now dominates. It was time, Gates realized, to make defense against the Google insurgency a top priority for Microsoft.

This story about using the Web to recognize a competitive threat hints at how much useful information the Net can now provide, but also illustrates just how random and unsystematic the process of developing that Internet-derived insight still generally is: the dominant mode of information-gathering still involves individuals browsing or searching for keywords that they think are related to issues that happen to occur to them.

Any system that helps users more consistently mine the Web for evidence that is relevant to their businesses must deal with three interrelated realities of Web-based business information. 1) The information-signaling events that are relevant to a user may be in broadly-varied formats, including various forms of unstructured text. 2) The information will likely be spread across a large number of high-volume sources, requiring a quick means of filtering the relevant from the rest, and of coordinating disparate signals to detect potential meaning. 3) Once translated to a standardized, structured form, relevant information will still often be in the form of weak signals that require the application of business knowledge to interpret; for

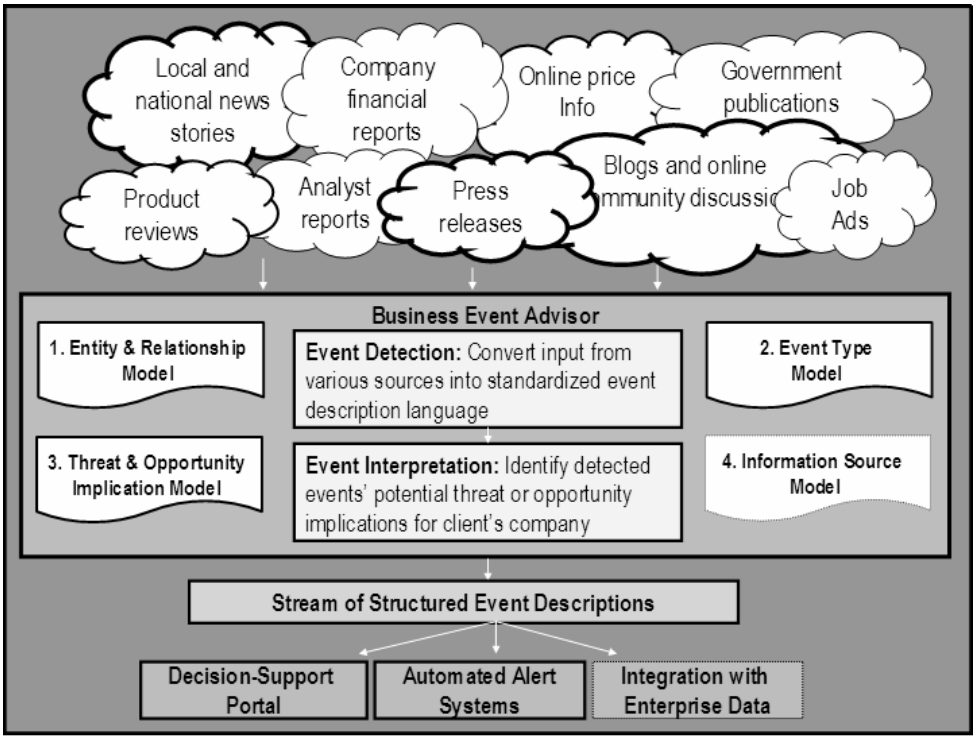

Figure 1. Conceptual framework for the Business Event Advisor

instance, only a system that models Microsoft's current niche and product mix would be able to detect the relevance of Google's recruiting priorities. Without such a model, it cannot analyze the indirect relationships between events it detects and the business objectives of the company it seeks to inform, leading it to either to ignore important events or cast its net too broadly.

We have focused our early efforts in this area on news stories and press releases rather than help-wanted ads, but we believe that the model-driven interpretation that drives our system is on the path to automating the monitoring of all kinds of online sources of competitive and market intelligence, including product advertising, financial disclosures, and regulatory announcements.

Interpreting Weak Signals

A key challenge that must be addressed to derive maximum insight from the Web is presented by the fact that the timely information found there is often in the form of indirect or weak signals. Helping the user see the relationship between what's being detected and the business outcomes he or she cares about requires a system with at least a skeletal model of the competitive dynamics in which a company operates. This allows the system to interpret the indirect relationships between events it can detect and the business objectives of the company it seeks to inform. The core idea behind the Business Event Advisor is that the system "knows" enough about the user's company to be able to interpret input data it collects from the Net and relate it to the user's business concerns.

Imagine that you manage a manufacturer that attempts to use the Net as a radar by running a system that monitors news stories and price data. If your radar is merely able to notice that one of your four competitors has lowered prices on its widgets, this may be of some limited value. But it might be too late to react once that's happened, and at any rate, your company is likely already to have noticed something as directly relevant to your business. Now suppose instead that it's the price of a raw material that has changed, rather than the price of widgets. Perhaps it's a raw material that your company doesn't use in any of its products. Such price shifts happen all the time, and humans trying to track and interpret all of these shifts are likely to get overwhelmed pretty quickly. But if this raw material is used by one of your competitors, the price change might have a strong, though indirect, impact on your business. Suppose we have a system that has a simple model describing who your competitors are, which of their products compete with yours, and what raw materials each manufacturer uses in each product. A model like that, combined with simple rules about cost/price relationships, allows a corporate radar to see that although you don't use that particular raw material, a drop in the price of that material may mean that your competitor can lower its prices on widgets, thereby putting price pressure on you (see figure 2, and for a more detailed discussion of the models that drive our system and the methods that could be used to generate them, see Kass and Cowell-Shah 2006).

In order to put the Business Event Advisor into perspective, it can be helpful to understand it in terms of a logical progression of levels of support that could be provided to decision-makers who seek Web-based evidence of potential threats and opportunities. Let's start with three stages of evolution that have already appeared in systems now in common use:

1. Manual selection of raw information sources to view, typically via a Web browser, by users who know where to look for it.
2. Automated retrieval of relevant information, typically though Web-based search engines, using individually-constructed, text-based queries. This information is still in its original published form, but the user no longer needs to know where it is located.
3. Automated retrieval of relevant information on a regular, systematic basis, through caching of queries by automated clipping or alert systems. The user still forms queries using keyword search terms, but no longer needs to manually issue queries on a repeated

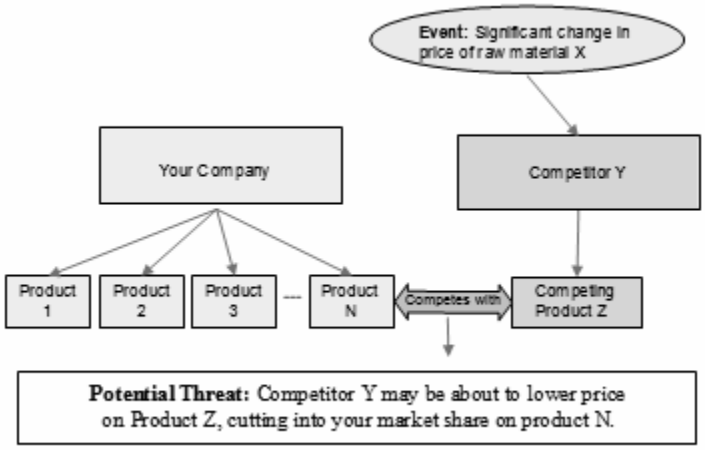

Figure 2. Inference derived from an event not directly related to you

basis to stay up to date.

The Business Event Advisor builds on this progression, using its semantic models to take the evolution a few steps beyond the approaches described above:

4. Automatic classification into semantic categories allows the documents collected to be categorized into business-relevant buckets, such as stories about a particular type of business event.
5. Attribute extraction allows the system to provide structured event descriptions that can be used to produce summaries, to further organize information displays organized around business-relevant entities, and to drive inference.
6. Inference rules allow the system to aggregate various detected events that could point to important implications, and to propose potential conclusions.

One way to look at this is that instead of telling our system what to look for and having it return documents that match your queries, you tell it what your company does (in the form of the semantic models of the competitive ecosystem). The system then forms queries or filters which make sense in that context, digests information sources to uncover business-relevant events, and organizes the events and their implications in ways that mirror the organization of the business relationships in its model. In the rest of this paper we will illustrate how the Business Event Advisor operates, and then close with a discussion of some further steps in the evolution of mining the Web for patterns of evidence that we believe will be important topics for future research.

The Business Event Advisor in Action

We do not have space here to describe all the details of an application of our Business Event Advisor system. As we've discussed, the system integrates a number of subsystems—some of which we have created ourselves, others we have pulled off the shelf—around a set of entity, relationship, and event type models and inference rules (all created with our GUI toolset), to produce a stream of reported and inferred event descriptions. (For a discussion of these subsystems and some of the implementation choices and challenges we addressed with each, see Kass and Cowell-Shah 2006. For a high-level architectural diagram, see figure 3.) These structured event descriptions can be integrated with enterprise data systems to generate automatic alerts, or used to populate a decision-support portal such as the one we have developed as a test-bed for the system.

For example, consider an application of our system designed for an executive of a manufacturing company,

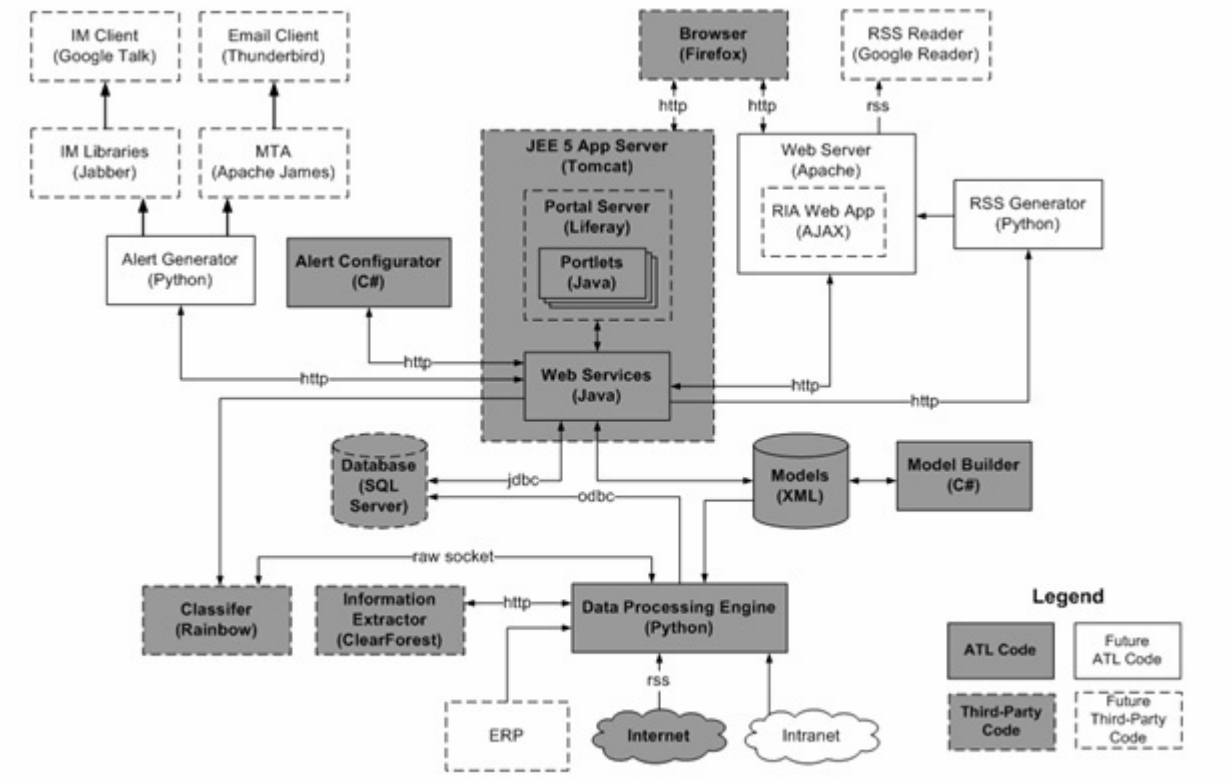

Figure 3. Component diagram for the Business Event Advisor

such as the Ford Motor Company. Using our portal as a front end, this executive can navigate around the competitive ecosystem to view displays of events affecting any portion of that ecosystem. For instance, the user might choose to view a summary of all events involving any of his company's suppliers (see figure 4). This view groups events according to the suppliers that they involve, categorizes each event into one of several predefined event type classes, assigns an importance level to each event, extracts information from each event's source text, and displays some subset of the extracted information (for example, the position and employer involved in a "Hire" event). The system produces this display by scanning all the data broadcast on the news sources that it monitors, filtering out any story that doesn't mention one of the entities in the entity/relationship model, classifying the events described in the relevant stories into the categories defined in the event model, and extracting as many of the event parameters as it can.

The system then derives implied events from the directly detected events, and displays these implied events in the same fashion (italicized text indicates that they are implied). In figure 4 we see events that involve a number of Ford's suppliers. Each event's classification is displayed within a colored banner at the top of the event; each classification type is represented with its own color. A five-star scale in the top right corner of each event indicates the system's estimate of the importance level of that event. A subset of the information extracted from each event is displayed beneath the colored banner.

By clicking on the plus sign at the left of any event, the user may open the event up to reveal more details. Figure 5 shows a close-up of a detailed description of a new product introduction by Denso, one of Ford's suppliers. This view displays all of the event parameters (as specified by the system's event type model) that the system was able to extract from the RSS-provided text that was used to generate the event. The text itself (in this case, the title and summary fields of an RSS entry) is displayed under the "Event Signals" header. When several different stories contribute to a single event, the text of each is listed under this same header. Each text also serves as a clickable link to a website that contains the full, original text that was summarized in the RSS feed. The buttons in the bottom left corner allow the user to correct the event type or any of the extracted pieces of information, should the system misclassify an event. This provides additional training data for future rounds of category learning.

Inferring Unseen Events

The "Implications" header in the opened event description in figure 5 displays descriptions of implied events that the system has inferred from this event. In this case, the system realizes that new products coming from one of the user's suppliers mean that both the user's company and one of the user's rivals may be able to expand their product lines or change features on existing products.

Our system uses a simple conceptual inference engine of the sort originally associated with Charles Rieger (see, for instance, Schank and Rieger 1974), in which the inferences are not logical implications, but commonsense inferences which are sometimes true, sometimes not. There are currently no probabilities associated with the rules, meaning that the system is purely qualitative. It identifies things that might be implied by the inputs, but does not provide any quantitative estimates of likelihoods.

Rules created with our graphical model-building tools contain rule-firing conditions specifying event types and constraints that must be met by key event attributes. When a rule's criteria are met by a directly detected event (i.e., an event that is generated directly from information available on the Net), an implied event is generated and populated with attribute values as specified in the rule. Any implied events that are created could then be fed back into the processing loop, where they would be processed just like directly detected events. We currently have no inference-control heuristics implemented, so we cut the system off at one level removed from directly-detected facts. While this basic conceptual inference technique has been in the AI toolkit for quite some time, we think it is exciting that the Internet (along with the text- and data-mining techniques required to turn the unstructured information on the Net into structured event descriptions) is now finally able to provide large amounts of data to exercise these engines.

It's worth noting that the Business Event Advisor considers the relationship between the entity involved in a directly detected event and the application's focus entity (in this case, Ford) when inferring events from that directly detected event. For example, a product introduction involving one of Ford's competitors would produce an entirely different set of implied events than would a product introduction involving one of Ford's suppliers. The former might imply that Ford could face price pressure on its products, while the latter might suggest that Ford could expand its product range or change features on existing products.

Figure 6 shows a close-up an implied event: a prediction that General Motors may experience reduced sales in the near future. Whereas the Denso product introduction event in figure 5 was directly detected from online sources, this reduced sales event was implied from eighteen separate directly detected events (this is indicated in the "Event Signals" banner). Notice that the directly detected events that imply this event are not all of a single event type: the reduced sales event was implied by both product introduction events and recall events.

Of course, because the signals leading from directly detected to implied events are weak signals, *not* logical implications, many of the implications suggested by the system will be false. An appropriate interpretation of the system's rules is a suggestion that the implied event *could* be happening, not that it actually is. In the end, it's up to the user to decide how likely the implication actually is; the system helps the user to make that judgment by reporting how many other signals of the same implication

have been detected. A weak signal combined with a number of others that point in the same direction can form a pattern that a user may give much more credence than he would have given to the lone signal without corroboration. In our next version of the system, users will be able to browse directly to view all the signals that point to a particular implication to decide how strong the overall pattern is. In more distant future systems we would like to have the system itself weigh the evidence, but at this point we consider that to be the user's job. The job of the current system is to help the user see the relevant patterns, not to make the final assessment of likelihood.

Directions for Future Research

The Business Event Advisor is very much a work in progress, and there are many more ways in which additional research is needed to improve its Web-mining performance then we have space to discuss here. For instance, the precision/recall performance of its text-classification and extraction engines is far from perfect, there is research needed to create an inference engine that is more carefully matched to this application, and the modeling tools we have created could be much smarter in the ways that they assist analysts in creating the models that drive applications of the system. But rather than discuss potential improvements in the system's algorithms, for the purposes of this forum we will concentrate on two research avenues that could add new types of functionality to more dramatically extend the evolution of Web-based insight tools outlined earlier in the paper.

Recall that we outlined a six-step progression: (1) manual browsing, (2) search, (3) aggregation, (4) event categorization, (5) event-parameter extraction and (6) weak inference. We believe that there are many more steps in this progression to come. Two next steps that we see as most important are as follows:

7. Learning weights for the inference rules used over time, so that assessing the strength of rules, and of the conclusions that those rules generate, can be further automated.
8. Learning about new entities that are relevant to the business ecosystem and their relationships within that ecosystem from the event data that the system processes, thus turning the static, hand-engineered entity-relationship model into one that can grow dynamically over time.

Much of the value of the Business Event Advisor derives from its ability to infer events that are not directly detected via information available on the Net, but that are implied by that information. The system is an evidence interpreter in that sense. It collects all the signals that might point toward a particular conclusion and packages them for easy perusal by the user.

A challenge in applying our system is that the inference rules in it are hand generated by content developers (who, under production conditions, would typically be industry analysts). This process is entirely dependent on the domain knowledge of the person or team creating the rules. It would be wonderful if we could replace this painstaking manual process partially or entirely with a machine-learning approach: an automated algorithm for generating rules based on emerging patterns of directly detected events. In theory, a rule-learning system of this type could not only dramatically reduce the burden of initial configuration, but could also improve performance of a particular application of the system over time, with new implied events emerging as recurring event patterns are detected and analyzed.

However, we don't think this is likely to be feasible at this time given the lower-than-needed density of news data available online. The data accessible online seems to be rich enough to provide useful alerts based on a model, but does not yet seem to be rich, consistent, or plentiful enough to allow for the automatic learning of the model itself. But while learning the rules themselves may be a bit too ambitious, we think it may be promising to at least have the system learn to associate strengths, or likelihoods with the rules over time. Because the system currently has no notion of rule strength, it cannot assign likelihoods to implied events. It tells the user how many signals contributed to an implied event, and it lets the user see full details about each of those signals, but it cannot decide how likely it is that an implied event will occur. That task is currently left to the user. It would be trivial to enhance our tools to allow analysts to add static strengths to rules, but it is not clear that analysts have reliable knowledge about what those strengths should be.

An obvious direction for future research, then, is to determine how the system could learn about rule strengths from the event patterns it detects, and then use these strengths to attach probabilities to inferences. The system would need to monitor all the predictions that it makes and compare them against the events that it later detects. When an actual event matches a predicted event, this would be taken as confirmation of the rules involved in that prediction and this would contribute to strengthening those rules. There are many challenging issues to address in order to flesh out this idea. What is the best way to match new and predicted events, and how should the system translate varying degrees of matches into varying rule strengths? Should temporal proximity matter? Should partial matches (say, an implied event about a competitor hiring a new treasurer, and a directly detected event concerning a supplier's new assistant treasurer) produce weaker rule strengths than full matches? Under what conditions should a rule's strength be reduced? While open questions certainly remain as to how exactly this feature should work, adding an automated means of collecting evidence about the likelihood of each inference rule's predictions coming to pass would enhance the system's role as a decision support tool.

Another topic on our research agenda is to investigate ways in which the system could automatically update its

entity and relationship model based on the details extracted from directly detected events. Any business ecosystem will change over time, and the task of keeping the model of a particular user's ecosystem up-to-date could become prohibitively burdensome if done manually. It would of course be far better for the system itself to modify the model as it notices changes in the ecosystem it is monitoring. This should, in theory, be feasible since the system is already identifying the events that signal model changes. For example, when the system detects a new product introduction, this could be used to add a new product to the model. Of course, there will be false positives, and a mechanism for pruning (either manually or automatically) will need to be a key component of such a scheme.

Conclusion

The tools available for analyzing business-relevant data on the web have evolved significantly from the days when users had to locate and digest any available information manually, to a stage where technology can now play a significant role in helping users see patterns of evidence that are relevant to their concerns. The Business Event Advisor contributes to this evolution by providing users with data that is more highly structured and analyzed in a way that directly relates to their business. In this paper we have discussed how semantic models can drive this type of analysis, illustrated what the output of the analysis can look like, and identified some of the next steps that tools of this sort might take as the evolution continues.

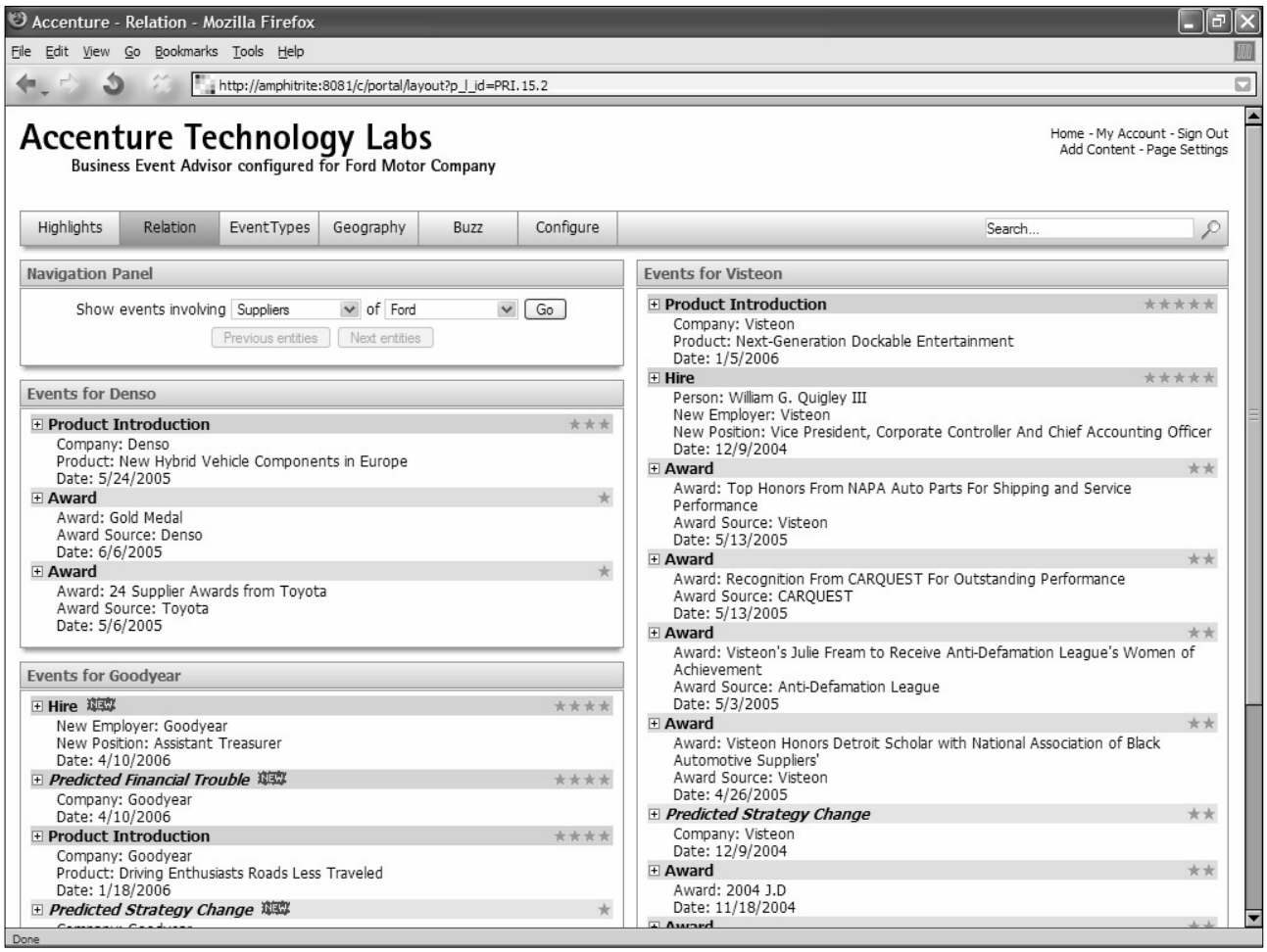

Figure 4. A summary of events relevant to Ford's suppliers

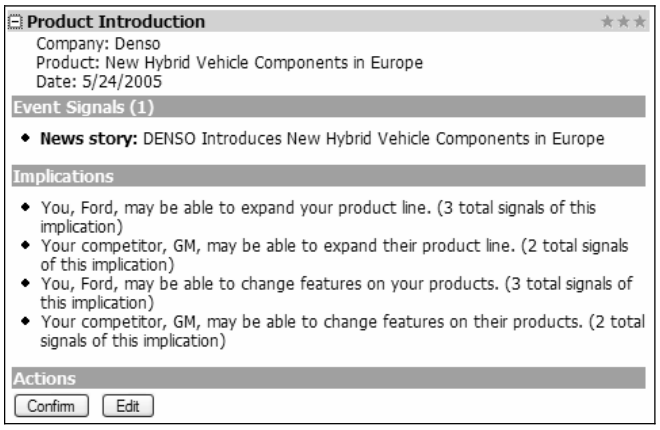

Figure 5. A pop-up window displaying more information about a Denso product introduction event

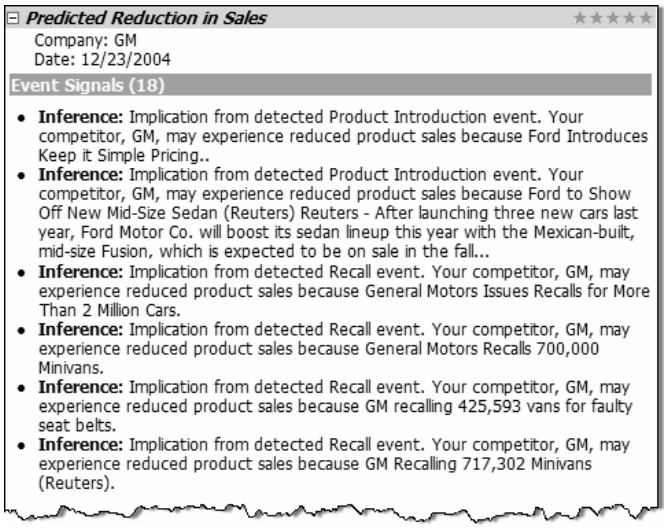

Figure 6. Detailed view of an implied GM product feature change event

References

Allen, J. 1995. *Natural Language Understanding*. Redwood City, California: Benjamin/Cummings.

Domingos, P. and Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29: 103–137.

Fuhr, N. 1989. Models for retrieval with probabilistic indexing. *Information Processing and Management* 25(1): 55–72.

Kass, A. and Cowell-Shah, C.W. 2006. Business Event Advisor: Mining the Net for Business Insight with Semantic Models, Light NLP, and Conceptual Inference. Forthcoming.

Manning, C. and Schutze, H. 1999. *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts: MIT Press.

Proceedings of 7th Message Understanding Conference, Fairfax, VA, 19 April–1 May, 1998. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html

Schank, R.C. and Rieger, C.J. 1974. Inference and the Computer Understanding of Natural Language. *Artificial Intelligence* 5(4): 373–412.

Vogelstein, F. 2005. Gates vs. Google: Search and Destroy. *Fortune* 151 (9). http://money.cnn.com/magazines/fortune/fortune_archive/2005/05/02/8258478/index.htm