

# Integrating Techniques for Event-based Business Intelligence Gathering

Kareem S. Aggour, John Interrante, Ibrahim Gokcen

GE Global Research  
One Research Circle  
Niskayuna, NY 12309, USA  
{aggour, interrante, gokcen}@research.ge.com

## Abstract

Gathering news events on companies to provide business intelligence to financial investors and creditors is a challenging problem. With a plethora of online news providers and tens of thousands of companies to monitor, automating the extraction and fusion of events is crucial. We developed an intelligent agent-based component framework to query and extract events from multiple providers. This framework integrates multiple machine learning and natural language processing techniques to down-select articles and extract targeted events. Results indicate that our system is capable of extracting focal events on a variety of topics with effective precision and recall.

## Introduction

In the wake of a number of high profile corporate scandals costing billions of dollars, many financial investors and creditors have been seeking methods to better identify financial distress in companies before those companies experience a significant decline in stock price or declare bankruptcy. Events such as CEO or CFO changes, mergers and acquisitions, and auditor changes can point to potentially significant shifts in a company's fiscal health. Schilit listed several qualitative events as warning signs for financial statement fraud, including frequent changes in auditors, CEOs, and CFOs (2002). Rezaee also identified a set of risk warning signs, including frequent organizational changes and turnover of senior management (2002).

News events can also be useful in sales applications. Announcements of pending changes such as a new facility construction or a major organizational change may be used to identify prospects for loan financing, equipment leasing, or investment opportunities, to name a few. Clearly, a system capable of identifying such events as they become public could be valuable for identifying new sales leads.

To collect event-based business intelligence, we developed an intelligent multi-agent system integrating a number of machine learning (ML) and natural language processing (NLP) techniques. There are many agent-based information-gathering systems available because of the advantages agents provide (e.g., flexibility, dynamism, and separation of control). From a computational intelligence

perspective, we identified the following common features in some of the existing systems:

- Tracking information utility (e.g. InfoSleuth (Nodine et al. 2000))
- Adaptation to news sources (e.g. Profusion (Fan and Gauch 1997) and InfoSleuth)
- Supervised learning (e.g. WARREN (Sycara et al. 1996))
- Reinforcement learning (e.g. Amalthea (Moukas 1997))
- Environment modeling (e.g. MOMIS (Bergamaschi et al. 2001))
- Customization to user (e.g. SIRUP (Ziegler and Dittrich 2004))
- Relevance feedback (e.g. Amalthea)

Our main contribution in this paper is the description of an end-to-end agent component framework called E-BIG (short for Event-based Business Intelligence Gathering), designed to solve a real-world business problem that is a productivity bottleneck in many organizations. In particular, we aimed at providing actionable and focused content to analysts in financial service businesses. E-BIG provides feature identification, query expansion, supervised learning, information extraction, relevance feedback and the distribution of extracted events to interested parties. A web front-end has been developed to facilitate the submission of queries and the review of extraction results. Glance et al. describe an end-to-end system with a similar philosophy, though it differs from E-BIG in terms of specific goals and the components and methods used to derive intelligence from free text (2005).

The paper is structured as follows: the next section provides further background on the problem and our solution. The multi-agent framework is then described, detailing the ML and NLP techniques integrated that enable the system to function. An analysis of the system's results is then provided. Finally, the paper concludes with a discussion of future work.

## Background

Even with the extensive research performed in the fields of information retrieval and information extraction, event gathering is still performed by hand in most commercial organizations. Businesses purchase licenses for news

sources, which analysts search manually. Two extreme cases arise from these manual efforts. Either an exorbitant amount of time is invested to perform a thorough analysis of the available sources, or they are effectively not searched at all. Compounding the problem, new sources regularly emerge and access rights to existing ones expire, resulting in a dynamic slate of sources at any given time.

A benefit of our system is the automated extraction and integration of events from multiple providers, reducing the time spent searching sources and focusing analysts' attention on the critical few articles and events. E-BIG enables users to specify queries that they would like run against any number of news sources integrated in the back end. Queries can be run for specific companies (typically for risk monitoring applications), and also for entire industries (typically for marketing purposes).

A secondary benefit is the reduction of repetitious work. Traditionally, as analysts uncover news events by hand they collect and store them in local data stores accessible only by the person who found the event (e.g., a file on the person's desktop) or by a select audience (e.g., on a shared drive or internal web site known only to a limited audience). An automated system has the added benefit of synthesizing and centralizing events so that analysts across large organizations can utilize the same repository for searching and reviewing events. In addition, such a synthesis capability facilitates collaborative decision-making and paves the way for assigning confidence values.

E-BIG is structured to query and retrieve news articles from multiple online providers simultaneously. For each event type we generate a simple search string and then expand it by combining frequently occurring informative keywords. This string is sent to the news source to retrieve articles and metadata. Incoming articles are classified into any or none of eight different event types (*bankruptcy, management succession, change in auditors, rating change, outsourcing, facilities expansion, litigation, and mergers & acquisitions*) by linear Support Vector Machines trained on a manually labeled corpus.

Zero-to-many events are extracted from each article for each event type, utilizing domain-dependent natural language ontologies and a dependency grammar. These events are fused in a single data repository along with article metadata provided by the sources. Finally, events are emailed to users at a regular interval (e.g., daily, weekly, etc.) using a publish/subscribe model where users indicate the companies and/or industries and the types of events in which they are interested. Our web interface allows users to review the events and drill down and read the full articles from which those events were extracted.

Components can be easily swapped in and out to add, modify, or delete functionality from the system without recoding or recompiling the entire code base. For example, an industry research analyst may want to receive personalized articles using a simple set of keywords based on implicit or explicit feedback provided to the system. The analyst may only want to keep himself abreast of current developments in the industry without drilling down into the specific focal events. In this case, only the

relevance feedback and text classification components would be needed. Alternatively, a sales leader may want to receive a list of important events in an article with the ability to drill down on occasion. Her goal may be to deploy sales force resources by providing quick and relevant leads with respect to a certain geographical region and category. Keyword identification, multi-modal classification (using event type and geography), and targeted event extraction would all be necessary in this scenario. Designing a system nimble enough to accommodate different types of users is therefore crucial to satisfying the business requirements.

## Framework Description

E-BIG is a flexible multi-agent system comprised of a Supervisor Agent, one-to-many News Gathering Agents, and an Event Extraction Agent. The agents are functionally independent and those in use at any point in time are defined via an eXtensible Mark-up Language (XML) configuration file. We implemented E-BIG in the Java 1.5 programming language using Telecom Italia Lab's Java Agent DEvelopment Framework (JADE) version 3.3<sup>1</sup>. The principal method used to initiate search requests is to enter search parameters into a web-based form implemented using Java Server Pages (JSP). The web application takes the search parameters from the form and sends a message to the Supervisor Agent. The agents interact as shown in Figure 1, and are described below.

### Supervisor Agent

The Supervisor Agent is the primary agent with which external systems (such as the web app) interact. When a user enters a query for a new search, a query request is generated and sent to the Supervisor Agent. To initiate a search, the following information is required:

- Search type – whether the search is at the company or industry level
- Company – for a company-level search, the internal ID of the company
- SIC – for an industry-level search, the Standard Industry Classification (SIC) codes of the industries
- Event types – one-to-many event types to monitor (*bankruptcy, litigation, etc.*)
- Source(s) – sources from which to fetch news articles
- Begin date – the date from which to fetch articles
- End date – the date until which to fetch articles
- Email – the email addresses to be notified when the search is complete

The Supervisor Agent sends a copy of the message to the appropriate News Gathering Agent(s) to carry out the search.

---

<sup>1</sup> <http://jade.tilab.com/>

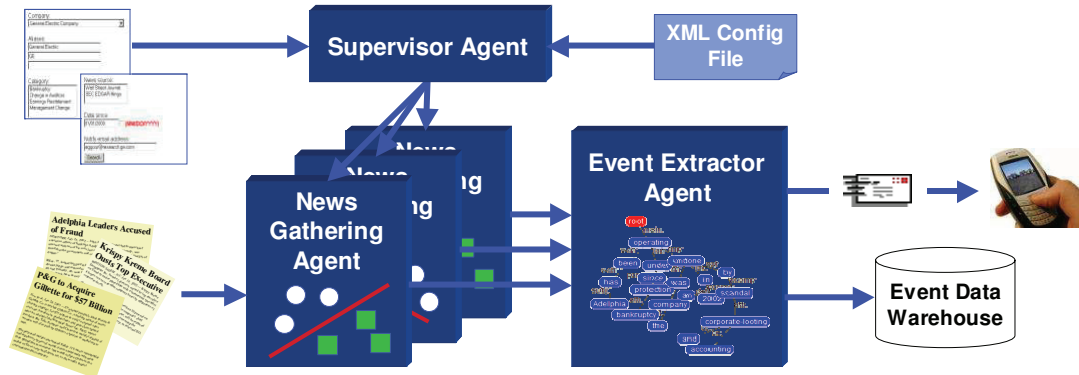


Figure 1: Agent Interaction in Framework

## News Gathering Agents

One-to-many News Gathering Agents can be associated with a single Supervisor Agent. Each agent is responsible for interacting with a single online provider (newspaper, news aggregator, company web site, etc.). These agents connect to the provider (including completing any required authentication), execute a search via the query mechanism supported by the provider (if any), and then retrieve any text (typically news articles) returned by the query. The agent parses the results so that any HTML or other mark-up language is removed and the desired text and relevant metadata (e.g., article date) are all that remain.

### Query Generation and Expansion

Queries are comprised of a Boolean combination of industry codes and/or company names, date ranges, and event types. A query  $Q$  can be denoted as the tuple  $\langle I, C, D, E \rangle$  where  $I = \{I_1, I_2, \dots, I_N\}$  is the set of industries,  $C = \{C_1, C_2, \dots, C_M\}$  is the set of companies,  $E = \{E_1, E_2, \dots, E_8\}$  is the set of event types and  $D$  is a date range. In a company or industry-monitoring scenario,  $C$  and  $I$  are already provided as inputs and are usually static. The existence of  $D$  and  $E$  however results in two interesting problems:

- 1) How do we eliminate duplicate retrievals if we already executed  $Q_1 = \langle I, C, D_1, E \rangle$  and would like to execute  $Q_2 = \langle I, C, D_2, E \rangle$ , where  $D_2$  partially overlaps  $D_1$ ?
- 2) How do we generate a set of keywords associated with an event type  $E_i$  to get relevant articles?

We solve the first by storing the executed queries in the database for each industry, company and event type. That is, we designate a row for each  $q = \langle I_i, C_j, D, E_k \rangle$  where  $i = 1, \dots, N$ ,  $j = 1, \dots, M$  and  $k = 1, \dots, 8$ . With this setup, if a date range has been previously searched for a certain company, industry and event type, we are able to retrieve any existing events from the database and then go to the news sources only for the range that has not been searched. The date range for the particular row is then updated to reflect the merged queries. As such, we minimize the access of news sources when possible, eliminating an important bottleneck and improving the system response time.

For the second problem, we use frequency-based keyword identification to find the most frequently occurring informative keywords in a hand-labeled corpus comprising articles labeled with each event type. These articles are represented as vectors of TFIDF (term frequency-inverse document frequency) values. There are eight classes corresponding to each event type  $E_i$  we cover. Positive and negative classes for each event type are determined in a one-vs.-all fashion. The set of keywords occurring most frequently in the positive-labeled articles were chosen to expand the query for that event type after stop word removal and manual inspection of the top ten keywords. For example, for the *litigation* event type this scheme identified “state,” “court,” “settlement,” “attorney” and “lawsuits” as the most informative keywords, coinciding well with intuition.

### Text Classification with Support Vector Machines

The final task of the News Gathering Agents is to filter articles based on their relevance to an event type before sending them to the Event Extraction Agent. Event extraction will be described in more detail in the coming sections and is the most computationally intensive task of our architecture. The News Gathering Agents provide the first level of filtering so that irrelevant articles are not sent to the event extractor, eliminating a potentially significant amount of unnecessary processing.

We use linear Support Vector Machines (SVMs) for text classification (Joachims 1998). SVMs proved to be very effective in this task and the high dimensionality of the problem almost guarantees the linear separability of classes. Preliminary experiments indicate that with a small corpus of articles (180 training and 100 testing samples), we achieved 87% classification accuracy on average for *bankruptcy* and *litigation* event types, for example. Experts manually labeled each article vector into one of eight classes, corresponding to the event types. Since labeling is time intensive, we expect the accuracy to improve once we augment the corpus with analyst feedback (i.e., label corrections) and additional articles (i.e., new samples).

For corpus validation we used dimensionality reduction to visualize subsets of the corpus before training. Projecting all article vectors onto two dimensions using

centroid dimension reduction (Kim, Howland, and Park 2005) provided us with a good indication of article clusters and potentially mislabeled articles. An example is shown in Figure 2, using *rating change* articles as the positive class with 10 positive and 50 negative samples. The separation between positive and negative articles is clearly visible, with one positive article potentially mislabeled based on its two-dimensional representation.

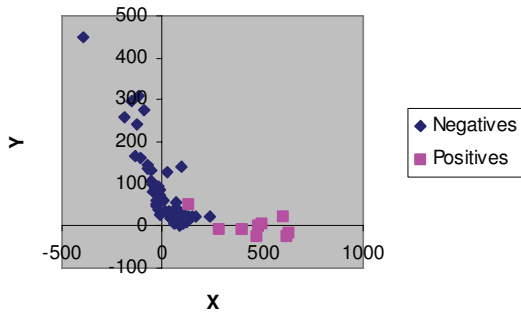


Figure 2: Articles Visualized in Two Dimensions

SVMs fit naturally in our architecture because they can be trained incrementally as feedback and new articles are obtained. The SVM coefficients and classes are saved and later updated with new information. As such, the retraining time is minimized, which becomes more pronounced as the number of articles in the corpus increases.

### Event Extraction Agent

Only one Event Extraction Agent is required in the system. It uses natural language processing techniques to analyze news articles and extract any pertinent content. This content includes both phrases of focal events, and entities (primarily companies) with which the events are associated. The techniques used are described below.

#### Topic Ontologies

Topic ontologies are used primarily to identify target sentences in news articles. They consist of topic patterns to first filter sentences and topic keywords to extract events. Topic keywords comprise key verbs (verbs that best describe an event occurring in the article) and key nouns (nouns that best describe the event in question). We used a combination of the query expansion method described previously and expert knowledge to build the topic ontologies. After a large ontology was created, we eliminated keywords and patterns that did not result in any significant change in the event extraction precision/recall. E.g., for *mergers & acquisitions* we used stemmed words such as “acqui”, “purchas” and “buy” as our keywords and “accept\*offer”, “agree\*acqui” as our patterns.

#### Targeted Phrase Extraction

Traditionally, there have been two ways to extract events: statistical and linguistic methods. Statistical methods generate models based on the inherent structures of sentences, usually identifying dependency structures using

an already annotated corpus of sentences. One drawback to this approach is that statistical methods cannot easily capture long-distance dependencies within a sentence. Linguistic methods attempt to capture linguists’ knowledge in determining constraints for syntax, morphology, and the disambiguation of both. We are particularly interested in constraint grammars (CG) (Karlsson 1990), dependency grammars (DG) (Tesnière 1959) and their combination, as exhibited by functional dependency grammar (FDG) (Tapanainen 1999). Dependency grammars are based on the premise that the syntax tree of a sentence has a unique root, which is the main verb of the sentence.

There are two kinds of dependency relationships between a particular token (word) and all others. Every token depends on another token but can have zero to many dependents. As a precursor to event extraction, we create a normative dependency structure where each token is linked to both the token on which it depends and its dependents. The main verb of the main clause is the only exception to this rule, as it depends on the sentence itself. The main verbs of other clauses in the sentence are linked to the main verb. An FDG parse tree displaying the dependency structure for the sentence “Delta announced last September that it was purchasing Western” is shown in Figure 3. In this simple sentence, “last September” is the contextual event, and the rest of the sentence is the focal event.

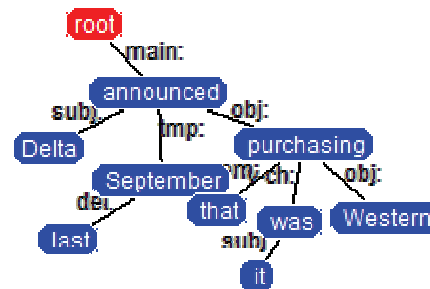


Figure 3: Example Parse Tree for Dependency Structure

In Figure 3, “announced” is identified as the main verb of the sentence. Once a sentence’s dependency structure is constructed, it is easy to extract clauses by finding the root of the tree (by definition all roots are verbs), and then going through its dependents recursively, stopping at the leaf nodes. We also call a clause a “closure list” in terms of syntactic dependency. As a special case, finding the closure list of the main verb essentially results in finding the main clause of the sentence.

Targeted phrase extraction (TPE) makes use of a sentence’s dependency structure. TPE is “targeted” because, given any target string, its linearly ordered closure list is constructed. Target strings are generally either company names or ontology keywords that assume certain roles in the sentence. Company names are extracted from each sentence using the named entity recognition product



Machine Extractor (MEX) from Connexor<sup>1</sup>. In our scenario, most of the focal events appear in sentences that have one or more company entities. TPE works as follows:

1. If the target string is the main verb of the sentence, we compute its closure list (parse tree) and output the resulting clause as the event form.
2. If the target string is either the subject or the object of the sentence, we find the verb it is connected to using the dependency structure and compute the closure list of that verb.
3. If the target string is a modifier of a subject or an object, we compute the closure list of the head verb.
4. If the target string does not fall into any of the categories mentioned above or the closure list has fewer than 3 elements, no focal events are extracted.

The rationale behind the rules can be understood when we consider some sample sentences. In a *management succession* example, say we have the word “appoint” as our target string. The extracted clause most likely will contain “appoint” as its verb, the company who appoints a given person as its subject, and the appointed person as its object. Similarly, if GE is the target string, we will have a chance to extract all clauses wherein GE is taking an action (i.e., the subject of the clause) or is influenced by an action (i.e., the object of the clause). As such we are able to filter out contextual events from the sentences.

There have been previous studies identifying rules for event extraction using dependency grammars. Yangarber et al. defined rules in the predicate-argument (PA) form, i.e. verb-object or subject-object pairs (2000). We adopt a similar strategy and denote a rule as an unordered tuple R as follows:  $R = ((w_1, r_1) (w_2, r_2) \dots (w_n, r_n))$ . Each  $w_i$  represent a word or concept class in a sentence such as the verb “appoint” or a Person concept class denoted as “C-Person”. Each  $r_i$  denotes the role of  $w_i$  in the sentence, such as VERB, SUBJ, OBJ or  $\lambda$  for any role. For example, for a *management succession* scenario, ((C-Company, SUBJ) (“appoint”, VERB) (C-Person, OBJ)) is a PA extraction rule designed to extract phrases such as “GE appoints Mr. Smith” and “Mr. Smith was appointed by GE”. Similarly, for a *mergers & acquisitions* scenario, some possible TPE rules are:  $R_1 = ((C-Company, SUBJ), R_2 = ((C-Company, OBJ), R_3 = ((C-Company, \lambda), (“acquire”, VERB))$  and  $R_4 = ((C-Company, \lambda), (“takeover”, MOD-OBJ))$ . In this example, all company actions can be captured with  $R_1$  and  $R_2$ .  $R_3$  and  $R_4$  extract clauses only if there is a company present in the sentence and “acquire” and “takeover” assume restricted roles.

During preliminary experiments we achieved an average of 77% precision and 64% recall for three of the eight event types (*mergers & acquisitions* (M&A), *bankruptcy* (B), and *facilities expansion* (FE)) using a hand-labeled corpus of Wall Street Journal (WSJ) articles and a Reuters corpus. Reuters-21578 is the only publicly available corpus we found to include *mergers & acquisitions* as an

event type (denoted by ACQ). It features 21,578 articles spanning 135 topics and we used the Apte-90 split, resulting in 2,143 articles with ACQ events. As a second corpus, we downloaded 300 articles from the WSJ by searching the news provider ProQuest<sup>2</sup> with keywords. We used “acqui”, “facility” and “bankruptcy” to bias those articles towards the three targeted event types. We then manually annotated all of the articles to assess E-BIG’s performance, as shown in Table 1. Performance results on the other five event types is left as future work, primarily due to a lack of publicly available benchmark corpuses.

	Precision	Recall	F1
Reuters-M&A	0.70	0.67	0.68
WSJ-M&A	0.78	0.66	0.72
WSJ-B	0.87	0.65	0.74
WSJ-FE	0.74	0.58	0.65

Table 1: Experimental Results

## Solution Analysis

E-BIG has proven extremely flexible in integrating a variety of ML and NLP techniques to provide targeted business intelligence to users. We initially implemented the framework with a single News Gathering Agent to integrate with the Wall Street Journal (WSJ). We then implemented News Gathering Agents to integrate with the Securities and Exchange Commission (SEC) website and Mergent’s Corporate Actions Service<sup>3</sup>. Although the agents differed in how they formulated their searches and gathered their news (HTTP for the WSJ, FTP for the SEC and Mergent), they shared similar code to process requests, check previous search results, and evaluate article relevance.

Due to the many similarities between the News Gathering Agents, we were able to take advantage of the WSJ agent to quickly author the new agents. Similarly, future users of this framework will be able to utilize the existing source code to rapidly implement and deploy additional functionality.

Currently, we run the framework on a SUN Solaris E4500 with 12 400-MHz CPUs and 12 GB RAM. On this machine, it takes an average of 3 seconds to complete the end-to-end processing of a single article. Considering the system selectivity (only about 10% of the gathered articles are sent to the event extractor), this allows the system to potentially process tens of thousands of articles a day, well beyond the capacity of even a large team of analysts. Some example key events (underlined) extracted by E-BIG are:

First Financial Management Corp said it has offered to acquire Comdata Network Inc for \$18 per share in cash and stock, or a total of about \$342.7 million.

<sup>1</sup> <http://www.connexor.com/>

<sup>2</sup> <http://www.proquest.com/>

<sup>3</sup> <http://www.mergent.com/>

Delta announced last September that it was purchasing Western.

Nerco Inc said its oil and gas unit closed the acquisition of a 47% working interest in the Broussard oil and gas field from Davis Oil Co for about \$22.5 million in cash.

Gander Mountain Inc said it acquired the privately held Western Ranchman Outfitters, a catalog and point-of-purchase retailer of western apparel based in Cheyenne, WY.

## Conclusions and Future Work

This work demonstrates the applicability of a multi-agent system integrated with a suite of machine learning and natural language processing techniques to the problem of collecting actionable events from multiple textual news providers. E-BIG was designed and developed to facilitate the collection and distribution of business intelligence for both risk and marketing applications.

Future work includes integrating additional data sources to provide a larger pool from which to extract events. This will include some major online news aggregators that can provide access to hundreds and thousands of different news sources through a single interface.

Other next steps include developing a web interface for industry experts to review articles extracted by the system and to provide their feedback. Once this feedback is available, it will be provided to the agents so they can learn which articles were relevant and which were irrelevant, to learn to better identify key articles and become more effective at extracting focal events over time. Cycles between modules can also be incorporated so that analyst feedback on the relevancy and specificity of news articles can be used to train and improve the performance of text classification. Feedback from text classification can then be used to improve query expansion in an incremental fashion. This allows us to identify a meaningful and useful relevance measure: determining relevancy based on the analyst's interest, which is guided primarily by the event's potential to generate revenue for the business.

Finally, we previously investigated the use of agents to query fee-based news sources for events (Aggour, Interrante, and LaComb 2006). These agents learned to manage the costs associated with the different providers, spending money on searches for business critical information and relying on free sources otherwise. Incorporating these agents to administer the News Gathering Agents would increase the number of news providers that could be cost-effectively integrated.

## References

- Aggour, K.S., Interrante, J., and LaComb, C. 2006. Agent Learning to Manage Costs for Event Detection. *3rd IEEE Conference on Intelligent Systems*. Forthcoming.
- Bergamaschi, S., Cabri, G., Guerra, F., Leonardi, L., Vincini, M., and Zambonelli, F. 2001. Supporting Information Integration with Autonomous Agents. In *Proceedings of the 5<sup>th</sup> International Workshop on Cooperative Information Agents*, 88-99. M. Klusch and F. Zambonelli, Eds. Lecture Notes In Computer Science, vol. 2182. London: Springer-Verlag.
- Fan, Y., and Gauch, S. 1999. Adaptive Agents for Information Gathering from Multiple, Distributed Information Sources. In *1999 AAAI Symposium on Intelligent Agents in Cyberspace*, 40-46. Stanford University.
- Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton R., and Tomokiyo T. 2005. Deriving Marketing Intelligence from Online Discussion. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 419-428.
- Joachims, T. 1998. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning, Springer.
- Karlsson, F. 1990. Constraint grammar as a framework for parsing running text. In *Proceedings of 13th International Conference on Computational Linguistics*, 3:168-173.
- Kim, H., Howland, P., and Park, H. 2005. Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research* 6, pp. 37-53.
- Moukas, A. 1997. Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. *Applied Artificial Intelligence* 11(5):437-457.
- Nodine, M., Fowler, J., Ksiezzyk, T., Perry, B., Taylor, M., and Unruh, A. 2000. Active Information Gathering in InfoSleuth. *International Journal of Cooperative Information Systems* 9(1/2):3-28.
- Rezaee, Z. 2002. *Financial Statement Fraud: Prevention and Detection*. New York: John Wiley & Sons.
- Schilit, H. 2002. *Financial Shenanigans: How to Detect Accounting Gimmicks and Fraud in Financial Reports*, 2nd edition, New York: McGraw-Hill.
- Sycara, K., Pannu, A., Williamson, M., Zeng, D., and Decker, K. 1996. Distributed Intelligent Agents. *IEEE Expert: Intelligent Systems and Their Applications* 11(6), 36-46.
- Tapanainen, P. 1999. Parsing in two frameworks: finite-state and functional dependency grammar. *PhD Thesis*, University of Helsinki, FI.
- Tesniere, L. 1959. *Elements de Syntaxe Structurale*. Paris/FR: Klincksiek.
- Yangarber, R., Grishman, R., Tapanainen P, and Huttunen, S. 2000. Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. *Proceedings of 18<sup>th</sup> International Conference on Computational Linguistics*.
- Ziegler, P., and Dittrich, K. 2004. User-Specific Semantic Integration of Heterogeneous Data: The SIRUP Approach. In *First International IFIP Conference on Semantics of a Networked World (ICSNW 2004)*, 44-64.
- M. Bouzeghoub, C. Goble, V. Kashyap, and S. Spaccapietra, Eds., *Lecture Notes in Computer Science*, vol. 3226. Paris, France: Springer.