Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information[1]

Chung Hee Hwang
hwang@mcc.com
InfoSleuth™ Group
Microelectronics & Computer Technology Corp. (MCC)
3500 West Balcones Center Drive
Austin, Texas 78759-5398, U. S. A.

June 18, 1999

Abstract

We report on an approach to representation and retrieval of information from large textual databases. Our approach is based on dynamic ontologies that are automatically constructed from textual data by a new method combining techniques from knowledge representation, natural language processing, and machine learning. The method learns concepts automatically from documents, and uses them to build domain-specific ontologies and to organize the information contained in the documents. The ontologies generated are dynamic in that they are constantly updated and expanded as new documents are added, requiring minimal supervision from domain experts. Information contained in the documents are efficiently retrieved based on concepts in the ontology, allowing for precision and completeness to be traded off. A prototype implementation has been very encouraging.

[1]

InfoSleuth is a trademark of Microelectronics and Computer Technology Corporation. All other company, product, and service names mentioned are used for identification purposes only, and may be registered trademarks, trademarks or service marks of their respective owners. All analyses are done without participation, authorization, and endorsement of the manufacturer.

1 Background

Our access to data continues to grow exponentially, in terms of both intranets and internets. The enormous growth in the number of on-line textual information sources brings us to a vast amount of information waiting to be discovered, generating intense interest in research on efficient representation and retrieval of textual information. In addition, the database community is becoming increasingly interested in non-conventional types of data; timely access to textual information has become more and more important as well.

At MCC, the InfoSleuth$^{\mathrm{TM}}$ project [Bayardo *et al.* 1996, Fowler *et al.* 1999, Nodine *et al.* 1999] aims to retrieve and process information in an ever-changing network of information sources. Recent technologies such as internetworking and the World Wide Web have significantly expanded the type, availability and volume of data available to an information management system. However, most of the current Web technologies rely on keyword-based search engines and are incapable of accessing information based on concepts. InfoSleuth integrates new technological developments such as agent technology, domain ontologies, brokerage, and distributed computing, in support of mediated interoperation of data and services in a dynamic and open environment. Since there is minimal structure in the data on the World Wide Web and this structure usually bears little relationship to the semantics, there can be no static mapping of concepts to structured data sets. Consequently, querying is mostly delegated to search engines that dynamically locate relevant information based on keywords.

InfoSleuth views an information source at the level of relevant semantic concepts and aims to deal with possibly incomplete information. Of InfoSleuth technologies, the most relevant for the topic of this paper is its use of domain models, i.e., ontologies. The semantics of information from diverse sources is best captured by ontologies because they give a concise, uniform and declarative description of the information, independent of the underlying syntactic representation or the conceptual models embedded in various information sources. Domain models widen the accessibility of information by allowing multiple ontologies belonging to different user groups. An ontology is typically designed to perform a particular kind of task in a specific domain. Thus, it could mean different things for different people, e.g., from relational database schemas to lexical entries in a dictionary. As our goal is to extract and retrieve information from unstructured textual data, our notion of ontology is broader than that of database schemas. Our first effort trying to model and query textual data using domain ontologies within the InfoSleuth environment is described in [Kashyap and Rusinkiewicz 1997]. This paper discusses how domain ontologies can be constructed and how concepts in the ontology can be mapped to the underlying textual data and the role of our ontologies in information retrieval.

2 Lessons Learned from KR , NLP , IE and IR

The similar goal of mapping ontological concepts to underlying texts have been pursued by the information retrieval and extraction (IR/IE) communities (see, e.g., [MUC-6, TREC-6]). One approach commonly taken in IE is to completely analyze texts using KR and NLP techniques.

The result is collected in a relational database-like repository for future retrieval.

A quick review of KR uncovers a host of representations, from standard predicate calculus to description logic (e.g., [Schmolze and Lipkis 1983]) to semantic nets (e.g., [Shapiro 1979]) to discourse representation theory [Kamp 1981] to situational semantics [Barwise and Perry 1983]. They are all different with respect to their expressive power, ease or efficiency of reasoning, and the rigor of semantic underpinnings. For instance, Episodic Logic [Hwang and Schubert 1993b] is an extended first-order logic that is very expressive, formally interpretable, and easily derived from surface utterances, yet allows efficient inference. The EPILOG inference engine [Schaeffer *et al.* 1995], based on Episodic Logic, makes quite complex inferences, e.g., spontaneous input-driven inferences or inferences needed for question-answering, with narrative texts [Schubert and Hwang 1999], utterances from the TRAINS domain [Traum *et al.* 1996], or telex reports for aircraft mechanical problems in the ARMS project [Namioka *et al.* 1992]. However, the same kind of representation may not be appropriate for the vast amount of texts we face on the internet or other textual databases.

Insisting on in-depth analysis and an exact, complete match with the current state of the art is practically impossible. Also, there are hard problems with any approach to information extraction that adopts deep linguistic analysis: current NLP techniques are not completely satisfactory, neither sufficiently robust nor fast. Development in the area of syntactic analysis has made it possible to parse NL sentences reasonably quickly, particularly with an empirical approach using statistics and/or learning. However, most of the currently available parsers still cannot choose the best parses consistently. Semantics and discourse analysis are even more difficult areas. Full understanding based on inferences is currently not practical for any real-time application.

At the other extreme is keyword search against the raw data. This has the advantage of not having to analyze texts and not having to have any particular representation, but has obvious problems: the same concept may be represented in various ways using different terms.

Information retrieval, or the access of information from diverse sources relevant to a particular query, is an area with many applications, including on-line access (such as the use of search engines like Lycos) and data-mining and knowledge discovery (as in [Unruh *et al.* 1999]). Typical technology to address IR includes something akin to regular expression matching into large bodies of text. However, without knowing ahead of time the particular keywords authors will have used to describe a concept, the match will either overgenerate (provide information that does not meet the questioner's needs) or undergenerate (miss information that is relevant). In [Hwang 1998], we discuss our strategy to attack this problem. Basically, one will have to limit oneself to superficial understanding initially and then concentrate on portions of documents that are likely to contain answers.

Then, what kind of superficial understanding would be sufficient? What kind of concepts need to be understood and represented, and using what representation? Representation and retrieval are two inseparable sides of a coin, each strongly dependent on the other. After all, it is the choice of the "subjects" of representation which determines how readily we can capture the content of documents, how easily we can retrieve answers for the queries, and how accurate and precise the answers are going to be. Our answer for this is the automatic generation of ontologies and modularization of the knowledge base. We use "simple" natural language processing and machine learning techniques to induce the ontologies specific to the application domain, reducing the amount of subsequent "full" linguistic analysis or human intervention and expertise required for building large knowledge bases. The ontology is organized in simple hierarchies, separating the inheritance type hierarchies from the rest of the knowledge, thus accelerating the reasoning process (e.g., with a specialized hierarchy reasoner) and reducing the complexity involved in ontology maintenance.

3 Ontologies Revisited

Much attention has been paid to ontologies recently by both AI, NLP and database communities. The EL ontology [Hwang and Schubert 1993a] is an example of the kind of ontologies developed for AI and NLP. It is a very liberal ontology and has all the machinery one needs to express various concepts including kinds, ideas, facts and events, and also has an inference engine capable of making complex inference efficiently. However, someone still needs to construct the ontology and knowledge base for a particular task domain.

An example of an organized semantic base is Cyc$^{(R)}$ [Lenat 1995], the construction of which originally started for common-sense reasoning. Despite its broad coverage however, its ontology and knowledge may not be adequate for specialized domains. It still requires new terms and concepts be added when faced with a new domain. Also, the costs involved in the creation of Cyc are legendary and could be an impediment to replication at a deep level for any particular task area. Moreover, in an undertaking like Cyc, there can be a significant delay between a new concept's use and its addition to the ontology because an expert must be involved in concept creation. This can be problematic for a number of fields with a continuous flow of novel ideas and techniques, such as computer science.

Constructing an ontology that is sufficient for all purposes and domains is indeed an impossible task. Furthermore, in this rapidly changing world, it may not even be desirable to build a comprehensive, stable ontology. Then what are the criteria we should consider in constructing an ontology? We list below some of the desiderata we perceive for large-scale ontologies. Although we list them point by point for the sake of discussion, it should be clear that they are largely interdependent.

- *Open and dynamic.* To adapt to the changes and new developments in a domain, an ontology should be open and dynamic both algorithmically and structurally for easy construction and modification. Ideally, the system should be automated, capable of "creating" concepts on its own with minimal help from a human and with minimum time delay.

- *Scalable and interoperable.* An ontology should be easily scaled to a wider domain and adapt itself to new requirements. It should also be simple to merge multiple ontologies into one; this is especially important in a distributed environment such as InfoSleuth. Integration of ontologies requires aligning the concepts from each existing ontology, which could become a daunting task if the ontologies have different conceptual taxonomies. This process could be considerably streamlined if the structure of ontologies is kept simple and clean.

- *Easily maintained.* It should be easy to keep ontologies up-to-date. Ontologies that are not easily maintained or do not scale will be short-lived. Again, it is important that ontologies have a simple, clean structure as well as being modular. They should also be easy for humans to inspect.

- *Semantically consistent.* The representation used by an ontology as well as the relations connecting the concepts in the ontology should have sound semantic underpinnings to assure semantic coherence.

- *Context independent.* It is desirable not to include context-charged or indexical terms or "vague" predicates in the ontology.[2] In database applications dealing with structured information, context dependent terms may be less of a problem (e.g., see [Jannink *et al.* 1998] for discussion on encapsulation). For ontologies dealing with large-scale, unstructured data sources, it is risky to include context sensitive terms. It is not practical to specify context-discharging axioms and compile de-indexing rules since such knowledge is usually "hidden" in the text, requiring deep linguistic analysis to make them explicit. Also, with context dependent terms in the ontology, integration of multiple ontologies would be a lot more complicated; as the scope of ontology is widened, the context will change.

What we wish to address in the rest of this paper is, how we create a useful ontology for the purposes of representing and retrieving information, minimizing the cost of initial creation, adapting to the concepts that are specific to a particular domain, and allowing for novel concepts to be added with minimum intervention and delay. The costs involved in generating ontologies by hand may be considerable. To reduce this cost will allow for better search of concepts online, relevant to an inquisitor, hence improving overall productivity within an organization. Because information content changes constantly as new things are discovered in the actual world, we need to make our system capable of detecting new ideas and developments when they are mentioned in texts, letting them be immediately searchable without the cost of additional ontological authorship.

4 Automatic Construction of Ontologies: How It Works

As part of on-going research at MCC, we have been developing a method to automatically construct an ontology from textual databases. There are two issues involved: where and how to get the ontology, and how to map concepts to documents and vice versa. Our approach is as follows. We let human experts provide the system with a small number of *seedwords* that

[2] "Wide" is vague, whereas "42-inch" isn't; "last year" is indexical, while "year 1998" is not (see [Hwang and Schubert 1994]).

represent high-level concepts. The system then processes the incoming documents, extracting phrases that involve seedwords, generates corresponding concept terms, and places them in the "right" place in the ontology. At the same time, it collects candidates for seedwords for the next round of processing. The iteration continues for a predefined number of times. As a simple example, suppose we are interested in the technology related to images and their display techniques. Given seedwords *display* and *image* and a set of documents in the general science and technology area, the system automatically learns the following concepts and constructs a hierarchical ontology (the root concept is omitted).

```
display
   field emission display
      diamond field emission display
   flat panel display
   *display panel
      *display panel substrate
image
   video image
      *video image retrieval
         *multiaccess video image retrieval server
```

The method considers only those sentences that involve seedwords, in particular, noun phrases such as "flat panel display" or "display on a flat panel." Several kinds of relations are extracted: `is-a` subclass relation (e.g., display – flat panel display), `part-of` (e.g., TV – TV screen), `manufactured-by` or `owned-by` (e.g., GE – GE image processing software), etc. The `is-a` subclass relation between concepts is usually clear, but other relations aren't and need further disambiguation. Instead of trying to disambiguate them however, we use an "under-specified" predicate `assoc-with` (associated-with) as a grab bag of all the relations other than `is-a`. The concepts marked with '*' in the above hierarchy are in `assoc-with` relation to their parents (`assoc-with` is a bidirectional relation).

The distinction between `is-a` and `assoc-with` relations is based on a linguistic property of noun compounds. When two nominals form a compound, the second one is the head of the compound and the first one is a modifier. For instance, in 'video image', 'image' is the head noun and 'video' is a modifier. Although the exact nature of specification due to modifier 'video' may not be immediately clear at the time of ontology construction, it can be reliably asserted that *video image* is a kind (i.e., subclass) of *image.* The relation between *video* and *video image* is more subtle, but it is obvious that there is some kind of *association* between the two concepts.

Between each round of iteration, we involve a human "coach" to ascertain that the concept augmentation is, on the whole, correct and useful. I.e., a human expert will prune the ontology constructed and check whether the seedwords suggested by the machine make sense. For instance, *field emission display* in the above is actually a subclass of *flat panel display*, and the correction is provided by a human expert. As for seedwords for the next round of concept learning, the method will "correctly" suggest *flat panel* and *video.* The revised part of the ontology would then look like the following.

```
display
   flat panel display
      field emission display
         diamond field emission display
  *display panel
     *display panel substrate
```

As more documents arrive, the system expands the ontology with new concepts it learns from the new documents, alerting the human experts of the new concepts[3] and *how* they occur in the source documents. This "discover-and-alert" is very important, and is a novel feature of our method. Since the construction of ontology is lexically driven, extending or merging ontologies is also straightforward as one can see. Note that ontology construction is easily distributed over various time periods and over various sites.

Another interesting feature of our approach is that the method may also discover some of the attributes associated with certain concepts. For instance, from the phrases shown below, the method may discover that objects of class *display* have attributes of physical dimensions or number of pixels and might even try to learn the range of their possible values.

```
17.5 Kg display
1.5-cm thick 12-inch display
42-inch wide-screen display
352x288-pixel liquid crystal display
```

Since the ontology is organized as hierarchies, attributes are automatically inherited following `is-a` links.

While constructing the ontology, our method also indexes documents for future retrieval, optionally saving the results in a relational database (e.g., in a table with attributes *document* and *set-of-concepts*).[4] In addition, it collects "context lines" for each concept generated, showing how the concept was used in the text, as well as frequency and co-occurrence statistics for word association discovery and data mining (this will be the focus of our future work).

Space limitations prevent a detailed description of various aspects of the algorithm, but we reemphasize the crucial features of our ontology: its simplicity in mapping information in the documents to concepts and potential retrieval of concepts that are *yet-unknown* to the user at the time of query. The method combines computational linguistics knowledge and techniques with machine learning techniques to avoid time-consuming, detailed syntactic/semantic analysis. Instead, it uses simple part-of-speech (POS) tagging to support superficial syntactic analysis. As any corpus-based approach, the more data we have the more reliable results we will get.

[3] This amounts to computing a set difference.

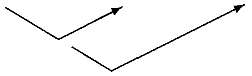
[4] The current implementation has an SQL query interface.

5 Difficult Problems

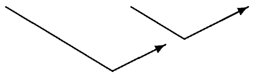
Our method of automatically constructing an ontology is conceptually simple and allows for efficient implementation. The ontologies generated by a prototype implementation have proved the method's practicality as well. However, there is much room for further improvement of the quality of the constructed ontologies and our current effort is centered around this issue. The main problem is how to distinguish concepts and non-concepts. We will first illustrate with examples some of the difficulties one may encounter in implementing our method, without getting into technical discussions.

One source of difficulty is structural ambiguity. A correct structural analysis of a phrase is important because the decision whether to regard a certain sub-sequence of a phrase as concept often depends on the syntactic structure. From the phrase *image processing software*, for instance, we may get the concepts *image, image processing* and *image processing software,* but not *processing software.* This is explained from the phrase structure. The correct analysis is (1a), not (1b), i.e., "processing software" is not a constituent.

```
(1a)  [[image processing] software]
```



```
(1b)  [image [processing software]]
```



In case of phrases like *low-temperature polysilicon TFT panel* however, we may admit as concept-terms all the subphrases shown below even though the phrase shares the same kind of syntactic ("front-anchored") structure as *image processing software*:

```
(2) TFT panel
    polysilicon TFT panel
    low-temperature polysilicon TFT panel
```

The next example involves a slightly different kind of problem, i.e., how to recognize different phrases that refer to the same concept.

```
(3) quartz crystal oscillator
    crystal quartz oscillator
```

The above two phrases refer to the same concept, and so do *quartz crystal* and *crystal quartz* (although the latter is not easy for the system to decide without actually seeing both occurrences in the literature). On the other hand, neither *electroluminescence color* nor *color electroluminescence* in example (4) is a concept despite the surface resemblance between examples (3) and (4).

```
(4) electroluminescence color display
    color electroluminescence display
```

Proper attachment of adjectival modifiers is also important to avoid creating non-concepts. For instance, we should not generate terms like *organic display, relational management, nmematic color* or *global display* from the following phrases.

(5) `organic EL` display
`relational database` management
`nmematic` color `LCD`
`global` display `market`

The next set of examples illustrates yet another potential source of spurious concepts.

(6) `flat panel display`
`touch panel display`
`plasma panel display`

Although these phrases seem to suggest creating a super-class concept *panel display*, such a concept would be controversial at best.

Syntactic structure disambiguation is a difficult problem itself, but errors can be made even after disambiguation is successfully done. We need to rely on statistical analysis of training corpora and tutoring by humans with sample "bracketing." Thanks to new advancements in empirical NLP and natural language learning techniques [Brill and Mooney 1997, Cardie and Mooney 1999] however, we are optimistic that we will get substantial improvements in the quality of the ontologies constructed.

Polysemic words present another kind of problem. For instance, 'panel' has several different senses as indicated by "flat panel display" and "international advisory panel." We expect domain experts will easily detect cases where the same word is used with different senses. We also expect that modification of ontologies for different subtrees depending on their sense wouldn't be very hard although it will require semantic analysis to a certain extent.

A validation of an ontology requires classification of a large set of objects. One may regard this as another source of difficulty, especially considering that terminological inconsistencies are common when document sources are diverse. However, we have not run into particularly difficult cases in practice.

Finally, one should bear in mind that automatically constructed ontologies may be too prolific and deficient at the same time. Excessively prolific ontologies could hinder domain experts' browsing and correction. We hope that tools and reasonable choice of seedwords and initial training data should limit this risk, but we do not yet know what the characteristics of an appropriate choice are. We need to run an extensive number of experiments with various data sets in order to appropriately tune our algorithm and cut down overgeneration. Next, automatically generated ontologies could be deficient since they rely on seedwords only. One promising technique could be synonym learning as discussed in [Riloff and Shepherd 1997].

6 Incomplete and Imprecise Retrieval

When the breadth of coverage is large, it is unavoidable that precision suffers. In some sense, having imprecision in retrieval is acceptable when it provides better recall. However, excessive recall overwhelms the information seeker and makes answer hard to locate. In this section, we discuss briefly how our ontologies provide a tradeoff between precision and recall.

Since the ontology is organized as hierarchies, inference based on inheritance is obtained automatically following `is-a` paths. Let us consider again the "display ontology" we discussed in section 4 (repeated below).

```
display
   flat panel display
      field emission display
         diamond field emission display
  *display panel
     *display panel substrate
```

Suppose we receive a retrieval query for *field emission display.* All the database entries (e.g., document ID's) linked to concepts that appear in the subtree rooted by *field emission display* and are in an `is-a` relation to its parent node constitute the answer space. The answer space would be {*field emission display, diamond field emission display*}.

Often, however, entries linked to an ancestor node of the queried concept are also relevant to the query. That is, some of the database entries mapped to concept *flat panel display* may contain answers to the *field emission display-* query. Similarly, entries mapped to a concept that is in an `assoc-with` relation to the queried concept node may contain answers. (For instance, if the queried concept is *flat panel display*, there is a chance some of the database entries linked to concept *display panel* contains an answer.) Thus, if we take as answer space only those database entries that are linked to the `is-a` nodes in the subtree of *field emission display,* the answer may be incomplete. On the other hand, if we also return entries linked to ancestor nodes of *field emission display* (as well as to the nodes connected by `assoc-with` relations, if any), we may lose precision. So, there is a tradeoff with respect to precision and completeness in what to include in the answer space, and our method is capable of allowing the user to decide between the precision and completeness. The downward aggregation is sound but not necessarily complete, while the upward and sideway aggregation may be closer to completeness but less sound. A typical application might provide a GUI with samples from each category and buttons to include the category in the result. A user then can decide whether to include them or not.

7 Concluding Remarks

We reported here on an approach to the management of domain- and application-specific knowledge specialization, namely, an approach to practical information representation and retrieval that aspires to be efficient, adaptive, and domain independent, with respect to the concepts encountered in textual information sources. The centerpiece of our method is a dynamic ontology that is automatically constructed using natural language processing and machine learning techniques. The method is based on detecting the relationship between linguistic features and simple ontology algebra based on inheritance hierarchy and set operations. It is capable of composing knowledge from diverse sources; it is scalable in that ontologies constructed by different groups can be merged, and expanded, without requiring an *a priori* agreed-upon ontology.

The method requires minimal input: a small number of seedwords of high-level concepts and POS-tagged, but otherwise unmarked, texts. We are also developing a web-sweeping agent [Perry 1999] that will crawl on the web, collecting documents potentially relevant to the task domain, which will then be piped to the concept-learner and ontology-builder.

The conception is not yet complete or fully debugged, but it is sufficiently far along to have provided a basis for a start-up implementation. Our initial prototype implementation has been successful. An internal MCC group, Global Technology Services (GTS), used our technology to support their efforts to inform their subscribers of what is new in the world in a series of technology focus points.

Acknowledgements

I gratefully acknowledge the encouragement and helpful comments received from Brad Miller and many members of InfoSleuth, especially Brad Perry and Gale Martin. I am also grateful to Helmut Schmid for making English Tree Tagger [Schmid 1994] publicly available on the internet at `http://www.ims.uni-stuttgart.de`.

References

[Barwise and Perry 1983] J. Barwise and J. Perry. *Situations and Attitudes.* MIT Press, Cambridge, MA., 1983.

[Bayardo *et al.* 1996] R. Bayardo, W. Bohrer, R. Brice, A. Cichocki, G. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk. *Semantic Integration of Information in Open and Dynamic Environments.* MCC Technical Report MCC-INSL-088-96, 1996.

[Brill and Mooney 1997] E. Brill and R. Mooney, editors. *Special Issue on Empirical natural language processing, AI Magazine, Vol. 18, No. 4.* 1997.

[Cardie and Mooney 1999] C. Cardie and R. J. Mooney, editors. *Special Issue on Natural Language Learning, Machine Learning, Vol. 34.* 1999.

[Fowler *et al.* 1999] J. Fowler, M. Nodine, B. Perry, and B. Bargmeyer. Agent-based semantic interoperability in InfoSleuth. *SIGMOD Record*, 28, 1999.

[Hwang and Schubert 1993a] C. H. Hwang and L. K. Schubert. EL: A Representation that lets you say it all. In *Proc. Internat. Workshop on Formal Ontology: Conceptual Analysis and Knowledge Representation*, pages 277–290, Padova, Italy, 1993.

[Hwang and Schubert 1993b] C. H. Hwang and L. K. Schubert. Episodic Logic: A situational logic for natural language processing. In *Situation Theory and Its Applications*, volume 3, pages 303–338. CSLI, 1993.

[Hwang and Schubert 1994] C. H. Hwang and L. K. Schubert. Interpreting tense, aspect and time adverbials: A compositional, unified approach. In *Temporal Logic: Proc. 1st Internat. Conf, ICTL '94*, pages 238–264, Bonn, Germany, 1994. Lecture Notes in AI, Springer-Verlag.

[Hwang 1998] C. H. Hwang. *Information Extraction in the Long Run: Utilizing NLP and Machine Learning.* MCC Technical Report MCC-INSL-006-98, 1998.

[Jannink *et al.* 1998] J. Jannink, S. Pichai, D. Verheijen, and G. Wiederhold. Encapsulation and composition of ontologies. In *Proc. AAAI Workshop on AI & Information Integration*, 1998.

[Kamp 1981] H. Kamp. A theory of truth and semantic representation. In Groenendijk, Janssen, and Stokhof, editors, *Formal Methods in the Study of Language.* 1981.

[Kashyap and Rusinkiewicz 1997] V. Kashyap and M.
Rusinkiewicz. Modeling and querying textual data using E-R models and SQL. In *Proc. Workshop on Management of Semi-structured Data*, 1997.

[Lenat 1995] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38, 1995.

[MUC-6] *Proc. 6th Message Understanding Conference.* Columbia, Maryland, 1995.

[Namioka *et al.* 1992] A. Namioka, C. H. Hwang, and S. Schaeffer. Using the inference tool EPILOG for a message processing application. *Internat. J. of Expert Systems*, 5:55–82, 1992.

[Nodine *et al.* 1999] M. Nodine, J. Fowler, and B. Perry. Active information gathering in InfoSleuth. In *Proc. Internat. Symposium on Cooperative Database Systems for Advanced Applications*, 1999.

[Perry 1999] B. Perry. InfoSleuth Web Agent (tentative title), in preparation, 1999.

[Riloff and Shepherd 1997] E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. In *Proc. Second Conf. on Empirical Methods in Natural Language Processing*, 1997.

[Schaeffer *et al.* 1995] S. Schaeffer, C. H. Hwang, J. de Haan, and L. K. Schubert. *The User's Guide to EPILOG* (Prepared for the Boeing Co. under Purchase Contract W-278258). Edmonton, Canada, 1995.

[Schmid 1994] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. Internat. Conf. on New Methods in Language Processing*, Manchester, U. K., 1994.

[Schmolze and Lipkis 1983] J. G. Schmolze and T. A. Lipkis. Classification in the KL-ONE representation system. In *Proc. 8th Internat. Joint Conf. on AI (IJCAI-83)*, Karlsruhe, Germany, 1983.

[Schubert and Hwang 1999] L. K. Schubert and C. H. Hwang. Episodic Logic meets Little Red Riding Hood: A comprehensive, natural representation for language understanding. In L. Iwanska and S. C. Shapiro, editors, *Natural Language Processing and Knowledge Representation: Language for Knowledge or Knowledge for Language* (in press). 1999.

[Shapiro 1979] S. C. Shapiro. The SNePS semantic network processing system. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers.* 1979.

[Traum *et al.* 1996] D. R. Traum, L. K. Schubert, M. Poesio, N. G. Martin, M. Light, C. H. Hwang, P. Heeman, G. Ferguson, and J. Allen. Knowledge representation in the TRAINS-93 conversation system. *Internat. J. of Expert Systems*, 9:173–223, 1996.

[TREC-6] *Proc. 6th Text REtrieval Conference.* 1997. NIST Special Publication 500-240.

[Unruh *et al.* 1999] A. Unruh, G. Martin, and B. Perry. Getting only what you want: Data mining and event detection using InfoSleuth agents. In *Proc. Agents '99*, Seattle, WA., 1999.