Ontology-Based Information Extraction from Free-Form Text

Contract Number: DAAH01-00-C-R113

Final Report

Period Covered: 30-March-2000 to 30-September-2000

Report Date: October 06, 2000

Key Personnel
Principal Investigator: Ronald Braun

Stottler Henke Associates, Inc. (SHAI)
1660 So. Amphlett Blvd., Suite 350
San Mateo, CA 94402

Distribution Statement A: Approved for public release; distribution is unlimited.

Ontology-Based Information Extraction from Free-Form Text
Final Report

Report developed under SBIR contract. In this Phase I SBIR research we demonstrated the feasibility of an information extraction (IE) system that can leverage semantic representations to significantly increase end-to-end recall for the IE task while maintaining or improving precision. Our end-to-end Ontology-Based IE (OBIE) system combines machine learning techniques with a novel architecture built around a shared domain ontology. This architecture enables interaction between different levels of the IE processing stream simultaneously through the shared ontology. By incorporating hierarchical knowledge in their learning algorithms, IE modules can perform their extraction tasks with greater depth and accuracy. Bootstrapping algorithms were extended to automatically learn the ontology of a new domain, to assist in training the IE components, and to reduce the burden of annotation on the end-user. Broad-coverage and rare-case extraction rules were augmented by classifiers induced from the trained ontology to shore up the precision typically lost by such rules. Performance metrics allow a preliminary characterization of recall and precision gains enabled by the proposed architecture. Our Phase I research and development of a proof-of-concept prototype demonstrated the feasibility and utility of OBIE's ontology-based IE capability and provides a solid foundation for our Phase II implementation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 6, 2000

3. REPORT TYPE AND DATES COVERED
Final Report; March 30, 2000 through September 30, 2000

4. TITLE AND SUBTITLE
Ontology-Based Information Extraction from Free-Form Text

5. FUNDING NUMBERS
DAAH01-00-C-R113

6. AUTHORS
Ronald Braun

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stottler Henke Associates, Inc.
1660 South Amphlett Blvd., Suite 350
San Mateo, CA 94402

8. PERFORMING ORGANIZATION REPORT NUMBER

Report No. 208: OBIE Final Report

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Army Aviation and Missile Command
Bldg. 7804, Room 222
Redstone Arsenal, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
NONE

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Report developed under SBIR contract. In this Phase I SBIR research we demonstrated the feasibility of an information extraction (IE) system that can leverage semantic representations to significantly increase end-to-end recall for the IE task while maintaining or improving precision. Our end-to-end Ontology-Based IE (OBIE) system combines machine learning techniques with a novel architecture built around a shared domain ontology. This architecture enables interaction between different levels of the IE processing stream simultaneously through the shared ontology. By incorporating hierarchical knowledge in their learning algorithms, IE modules can perform their extraction tasks with greater depth and accuracy. Bootstrapping algorithms were extended to automatically learn the ontology of a new domain, to assist in training the IE components, and to reduce the burden of annotation on the end-user. Broad-coverage and rare-case extraction rules were augmented by classifiers induced from the trained ontology to shore up the precision typically lost by such rules. Performance metrics allow a preliminary characterization of recall and precision gains enabled by the proposed architecture. Our Phase I research and development of a proof-of-concept prototype demonstrated the feasibility and utility of OBIE's ontology-based IE capability and provides a solid foundation for our Phase II implementation.

14. SUBJECT TERMS
Ontologies, Information Extraction, Algorithms, Event Bootstrapping, Semantic Tagging

15. NUMBER OF PAGES
49

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT

UNLIMITED

Ontology-Based Information Extraction from Free-Form Text
Final Report

1 INTRODUCTION 4
1.1 PHASE I OBJECTIVES 5
2 PHASE I INVESTIGATION 8
2.1 PROTOTYPE DESIGN 11
2.1.1 Ontology Development *12*
2.1.2 Document Processing *14*
2.1.3 Entity Bootstrapping *14*
2.1.4 Classifier Development *16*
2.1.5 Event Bootstrapping *18*
2.1.6 Semi-Automatic Ontology Discovery *19*
2.1.7 Semantic Tagging *20*
2.1.8 Event Extraction *21*
2.1.9 Prototype Demo *23*
2.1.10 Prototype Summary *24*
2.2 DESCRIPTION OF METHODOLOGIES 25
2.2.1 Bootstrapping *25*
2.2.2 Classification *25*
2.2.3 Marker-Passing Algorithms *26*
2.2.4 Shrinkage *27*
2.2.5 Semantic Parsing *27*
2.3 LITERATURE SEARCH 27
2.4 LESSONS LEARNED DURING PHASE I 29
2.5 TECHNICAL FEASIBILITY 31
2.6 CONCLUSIONS 32
3 PHASE II DESIGN & FUTURE WORK 33
3.1 DESCRIPTION OF SYSTEM 33
3.1.1 Document Management *35*
3.1.2 The Ontology *35*
3.1.3 Training Components *36*
3.1.4 IE Stream Components *36*
3.1.5 Results Database *37*
3.2 PHASE II TECHNICAL OBJECTIVES 37
4 COMMERCIALIZATION PLANS 38
5 REFERENCES 39
APPENDIX A – DEMO SEQUENCE 41

1 Introduction

The unprecedented ability to access information on-line provided by the World Wide Web and other Open Source text databases has created an overwhelming amount of free text and a concurrent need for the automatic mining and extraction of information from that text. In response to these needs, the field of *information extraction* (IE) has arisen to address the goal of creating domain-independent, portable systems capable of extracting information from free text. Driven in large part by the DARPA-sponsored Message Understanding Conferences (MUCs) [SAIC, 1998], IE technology has evolved a capacity to extract shallow (i.e., verb-centered) events and named entities from text through a variety of both handcrafted and statistical, corpus-driven text analytic approaches. Deeper extraction is performed by identifying the relationships between the output lower-level entities and events via scenario pattern matching (refer to [Muslea, 1999] for a summary) and assembling them into scenario templates (e.g., terrorist activities, corporate succession events, etc.). While statistical techniques have made some inroads into this more challenging IE task of complete template extraction, hand-derived systems still enjoy a slight edge in performance due to the encoding and application of human ingenuity. Such handcrafted systems do not lend themselves to domain-portability, however, and the trend in IE over the last few years has been a shift to corpus-driven, automatic approaches for all levels of the IE task.

The traditional metrics for accuracy within the IE community are those of *recall* and *precision.* Recall is the ratio of correctly extracted items (entities, template slots, or complete templates, depending on the granularity of the IE task) to the number of items actually present in the text. Precision is the ratio of correctly extracted items to the number of items both correctly and erroneously extracted from the text. A low recall measure corresponds intuitively to many false negatives; that is, many items that should have been extracted were not. A low precision measure corresponds to many false positives; that is, many items were extracted when they should not have been. Another metric, the F-score, is sometimes used to combine both precision and recall into one score. The F-score used in this proposal, F = (2 * precision * recall) / (precision + recall), equally weights both metrics.

There is a generally recognized trade-off between precision and recall in that points in precision may typically be purchased at the cost of some points in recall, and vice versa. The reason for this trade-off stems from the high redundancy of natural language: an item may be referred to by many different words configured in many different ways, and a training set for an IE system is likely to have a sparse representation of this infinitely expressive space. This creates tension between the specificity of examples derived from a training corpus and the need for generality sufficient to process new text. This tension is of fundamental concern to all machine learning tasks.

IE systems generally have tended towards higher precision and lower recall, primarily because learning systems are driven by the features available in the training corpus, which as just mentioned are sparse with respect to the total expressiveness of natural language. Generalizing without over-generalizing (thus increasing recall without decreasing precision) has proven a difficult problem. State-of-the-art end-to-end IE systems have pretty much leveled-off at F = 60% [Appelt and Israel, 1999]. In the MUC-6 evaluations, for example, typical template-extracting systems performed in the range of a recall of 43 to 50% and a precision of 59 to 70%

for F-scores of 51 to 56% [Grishman, 1997]. Scores range higher in specific IE subtasks (e.g., named entity discovery, part-of-speech tagging, etc.) but preserve the general trend of higher precision and lower recall. The objective of improving recall while maintaining (if not also increasing) precision has therefore become an important goal in IE research. This final report describes our Phase I SBIR efforts in meeting this goal.

1.1 Phase I Objectives

The primary goal of the Phase I research was to develop and assess the feasibility and utility of a new architecture for portable end-to-end IE systems designed to significantly increase system recall while maintaining or increasing precision. All of the tasks set out in the Phase I proposal were successfully accomplished and indicate that an end-to-end IE system incorporating the developed techniques will perform high-precision, high-recall extraction of events, entities, and their relationships within a given domain of interest. Further, the resulting system will be user-centered, will require little domain or system expertise on the part of the user, and will facilitate the sharing of information between domains via shared ontologies.

The Phase I prototype addresses the goal of increased recall in two ways. First, it demonstrates that a centralized ontology facilitates the sharing of semantic information between modules in the IE stream and provides an efficient common representational structure (similar to a *blackboard* in many classical AI systems) for use by any component. The hierarchies of the ontology permit rules and lexicons to be specialized to nodes at precisely the right level of abstraction. This maintains precision by filtering out the application of rules to non-semantically related concepts while increasing recall by generalizing the rules to operate on all members of the conceptual hierarchy rooted at that level in the ontology. Rules that extract the entities related to explosions in a terrorist domain, for example, are specialized enough to preclude application to non-explosive weapons (e.g., to guns, a sibling concept in the weapons abstraction hierarchy) but are applied to all specializations of an explosive, including grenades, mines, and so forth.

The second manner in which OBIE increases recall is by allowing very general or rare-case rules to be added to its extraction rulebases. Many existing IE systems attempt to learn only the most predictive of rules, discarding rules that extract too many irrelevant items. This process favors precision at the expense of recall. OBIE takes the opposite approach, permitting any rule that extracts relevant information to be included in its rulebases. To ensure that recall is not simply being purchased at the expense of precision, OBIE then shores up the reduced precision via a different process. The training procedure by which lexicons and rulebases are associated with nodes in the ontology efficiently captures information that can be used to discriminate between different semantic interpretations of an item extracted by a high-recall rule. This information is exploited by OBIE to create a suite of classifiers that are capable of enforcing precision when any of OBIE's rules are applied. These classifiers thus restore the precision lost by favoring broad-coverage rules over only highly predictive rules.

In addition to addressing the primary goal of increased recall, OBIE also addresses several secondary goals deemed important to the successful deployment and commercialization of a fully-functional IE system. OBIE employs a *user-centered* approach to text analysis that assumes little domain or system expertise. Bootstrapping discovers the highest coverage rules first, allowing an iterative exploration of a new domain while maximizing the benefits of user training. The user can get a functioning extraction system up and going in a short amount of time, with each iterative cycle of training resulting in increased accuracy and coverage. Additionally, OBIE can assist in the actual modeling of the domain by integrating ontology development with the discovery and training phase of the system. Some preliminary investigations were conducted which characterized the degree to which semi-automated ontology discovery could be integrated within OBIE's training cycle.

Our Phase I research and development have laid the groundwork for the Phase II implementation of a complete system for high-accuracy, user-centered, domain-independent text extraction, and its eventual commercialization. This research addressed the primary objectives identified in the Phase I proposal as being necessary to support the goal of high-recall, ontology-driven information extraction:

- **Develop an ontology supporting generalization and compositional hierarchies accessible via an Application Programming Interface (API) to form the backbone of the IE processing pipeline.** During our Phase I research, we identified several types of ontological relationships necessary to represent entities and events within any arbitrary domain. We identified the need for a simple abstraction hierarchy to capture varying levels of generality among ontology nodes and a simple compositional hierarchy to specify role and part-of relationships between nodes. A frame-based representational scheme similar to Memory Organization Packets [Schank, 1982] encapsulates all of the necessary characteristics and was chosen as the basis of our implementation. This scheme supports abstraction and packaging links as well as the infrastructure required to attach rulebases and lexicons to specific nodes in the ontology. A rudimentary API mediates access to the ontology; any component of the system is free to elaborate or consult the ontology when desired. This common structure facilitates automatic communication and sharing between modules of the IE stream.
- **Create a simple, graphical toolkit to allow browsing of the ontology, to support viewing and annotation of corpus documents, and to support algorithmic interaction with the end-user.** While the GUI components of the prototype were not emphasized, sufficient user interfaces were developed to support the Phase I research effort. The interfaces permit visual inspection of the ontology and the current state of its hierarchies and infrastructure, including lexicons and node instantiations. Learning interfaces present entities and events for classification in their original textual context, giving maximum support to novice end-users unversed in the internals of the system. Document browsers permit the viewing of syntactic and semantic features within the corpus of interest. Clustering visualizations are used to uncover the structure of events of interest to the user.
- **Develop bootstrapping and active learning techniques to automatically discover the domain ontology, thereby reducing the burden on the end-user when porting OBIE into a new domain.** Our vision of a commercial version of OBIE involves both the automated elaboration of domain ontology structure from free text under guidance of a non-expert end-user and the automated discovery of the lexicons and rulebases that permit the mapping from free text into those ontological representations. The Phase I effort emphasized those processes that best support high-recall information extraction rather than those that address the more abstract questions of ontology induction from free text. For this reason, the OBIE prototype assumes a largely static initial ontology that is to be used during the IE process. New nodes (called *scaffolding nodes*) are created in the ontology during the training step to

support the learning of mappings between natural language and target nodes of the ontology. Functional knowledge is also represented explicitly within the ontology to permit efficient marker-passing algorithms to assist in text processing tasks.

Through bootstrapping techniques coupled with an active learning approach to solicit feedback from the user, the prototype demonstrates how lexicons and rulebases may be learned that allow ontological nodes and their relationships to be recognized from text. These rules and lexicons may be generated quite efficiently from only a few sample seed instances by an end-user engaged in simple classification tasks (i.e., "Is this entity in this context an example of the class you are interested in?"). By allowing the results of event and entity bootstrapping to cross-seed each other, the OBIE system can be rapidly deployed in a new domain. Classifiers were also developed from these rules and lexicons that permit high-recall rules (e.g., those that are highly general or rare-cases) to be used by extraction applications. These classifiers enforce the precision of high-recall rules, ensuring that recall is not purchased at the expense of precision.

In accordance with our second longer-term objective of automated ontology induction from free text, we also investigated algorithms and techniques that might permit the actual structure of the ontology to be similarly bootstrapped from text. We have identified (though not implemented) several promising techniques described in Section 2.1.6 based on rulebase and lexicon clustering that may be able to facilitate ontology discovery automatically or through the informed guidance of the end-user. The OBIE prototype includes several tools supporting the investigation of these techniques.

- **Implement a semantic analysis IE component and a scenario pattern-matching component capable of leveraging the ontology to achieve increased recall without reducing precision.** During Phase I, we implemented two application tools: A semantic tagger and a scenario-template event extractor, that integrate with the ontology to perform their tasks. Each tool leverages any work done in elaborating the ontology during training by the other tool to its own advantage. The training performed to allow OBIE to recognize entities for semantic tagging permits rapid bootstrapping of the event extractor and vice versa. Both components exploit the same representational structures and classification algorithms to perform their respective tasks. We present some preliminary results demonstrating the effectiveness of the OBIE approach.
- **Develop a limited prototype that highlights OBIE's most critical features and demonstrates the feasibility of the proposed approach.** The prototype developed as part of our investigation served three important functions. First, it was a useful research platform, allowing us to better understand the interplay of abstract ontological representations with the actual instances of their occurrence in text. This also allowed us to characterize the trade-off between precision and recall, permitting us to increase recall while maintaining precision through the use of trained classifiers. Second, the prototype was an effective demonstration tool, since it was a functional test model illustrating the actual process used to rapidly bootstrap IE applications in a free text environment. In a short twenty-five minute demo requiring just four seed terms ("bomb", "grenade", "mine", and "died"), a fully functioning extraction process is bootstrapped by the prototype. This process extracts 16 scenario templates describing bombing events (including the weapon used, any casualties, and the

bombing location) from a test set of 200 documents. Third, the prototype serves as the launching pad for the design of our Phase II system and the ultimate commercialization of an OBIE-like system.

In addition to meeting the primary objectives laid out in the Phase I proposal, we also discovered during the implementation of the prototype that the technique of employing classifiers to shore up high-coverage extraction rules could significantly increase recall. This technique was not predicted in the original proposal but has since become a foundation to our approach. We therefore introduced and satisfied an additional goal to those originally proposed:

- **Develop a suite of classifiers capable of restoring precision for broad-coverage extraction rules.** We developed three different types of classifiers to help enforce the precision lost by favoring rare-case or overly general (i.e., high-recall) extraction rules. These classifiers used lexical similarity, similarity in linguistic pattern frequencies, and sentence similarity to classify entities extracted from text via such extraction rules. The use of these classifiers permits OBIE to favor rules that enhance recall while avoiding the traditionally associated penalty of decreased precision.

2 Phase I Investigation

The Phase I research objective for OBIE was to develop and assess the feasibility and utility of a new architecture for portable end-to-end IE systems designed to significantly increase system recall while maintaining or increasing precision. The new approach departs from the traditional architecture consisting of a loosely coupled pipeline of IE components (Figure 1) to a tightly integrated, ontology-based IE system that permits a natural interaction between components and with a non-specialist end-user (Figure 2). Active learning and bootstrapping are used to intelligently adapt both the ontology of the system and its IE components to a new domain without imposing the heavy burden of text annotation on the end-user required by traditional supervised learning algorithms. Classification methods that exploit the results of bootstrapping permit the incorporation of general, broad-coverage extraction patterns by verifying the results of each extraction.

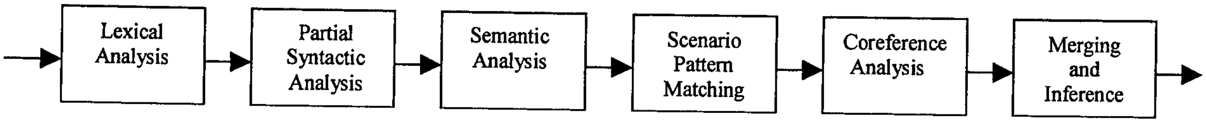


Figure 1. Traditional Pipeline IE Stream

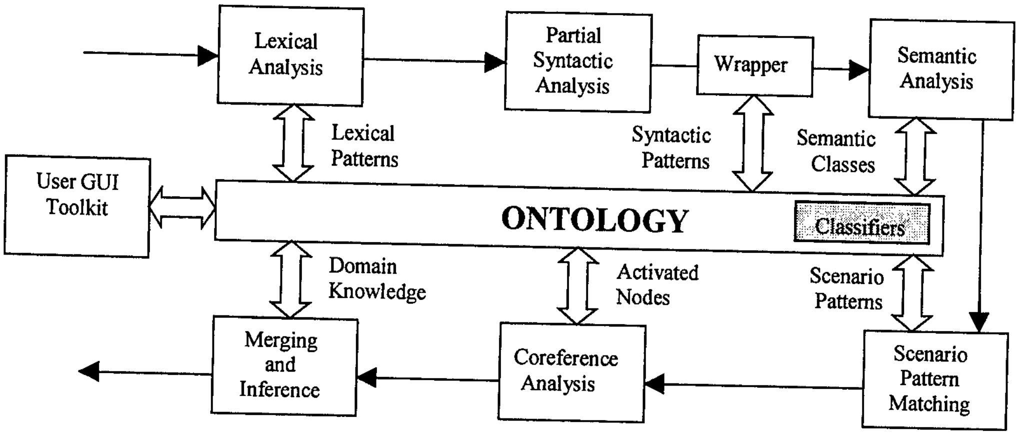


Figure 2. Ontology-Based IE Stream

This architecture provides a natural solution to several recall-limiting problems (identified early on from a review of the IE literature) that have kept the accuracy of IE systems artificially low:

1. *There is no feedback between modules in most IE systems.* Typically, information made available by components downstream is not accessible to modules upstream. This is largely a consequence of the black-box nature of each IE component. No effort is made to share representations or techniques across modules. The benefits of each representation are constrained to the module and are not available to other modules, even when such information might be useful. Additionally, errors that occur early on in the stream are propagated (and sometimes amplified) by subsequent processing stages. A methodology to allow all components of the system to work in an integrated manner and to learn from each other would be of benefit [Glickman and Jones, 1999; Roth, 1998]. OBIE offers such a methodology by providing a shared data structure (the ontology) to all modules. This ontology also enforces a common semantic interpretation by all modules of the IE stream during different phases of text processing. In the Phase I prototype, the event extractor application makes use of the ontological structures built up by the semantic tagger, and the semantic tagger automatically accesses and tags the entities bootstrapped during training by the event extractor.

2. *IE systems that rely purely on lexical or syntactic features of the input corpus are limited by the informational content of that text.* There is a general consensus that the F = 60% rough upper bound that limits current IE systems is a byproduct of the informational content readily accessible by systems capable of shallow syntactic processing with little or no recourse to domain models [Appelt and Israel, 1999]. External sources of knowledge may be necessary to surpass that 60% ceiling. For example, semantic networks have been used by [Humphreys, et. al., 1997] to make inferences from a source outside of the input text during event coreference resolution. (Coreference resolution involves recognizing different descriptions of the same entity or event, including pronoun use and anaphora.) External lexicons and semantic nets such as WordNet [Miller, 1995] have been employed by numerous IE researchers, but the gains in lexical or semantic breadth they provide are typically offset by having to deal with rare-use cases not relevant to the domain. The proposed ontology provides a portable, automatic knowledge source tuned precisely to the domain under consideration. Ontological representations of domain objects are sharable across domains, though new training is necessary to tune the lexicons and rulebases of the ontology to the idiosyncrasies of each domain.

3. *Many IE systems do not effectively leverage the involvement of an end-user.* While end-user involvement is crucial to the porting of an IE system to new domains, a consensus is emerging that traditional *supervised learning* (in which a human annotates a large set of test cases within a domain for use in training) is an inefficient use of human resources, creating a bottleneck in the IE process

[Kehler, et.al., 1998; Glickman and Jones, 1999; Jones, et. al., 1999; Soderland, 1999; Thompson, et. al., 1999]. Because of this bottleneck, insufficient amounts of annotated text exist from which to effectively train systems when porting them across domains, thereby decreasing the accuracy of those systems. Further, the annotations provided by a non-specialist human agent reduce the semantic knowledge of that agent into an impoverished representation (a simple mapping between text and templates or other entities) rather than capturing useful domain knowledge as a natural part of the interactive process. Bootstrapping and active learning are two techniques used by OBIE to diminish the burden of text annotation. Both techniques are part of the current trend away from purely supervised learning. The combination of both approaches allows iterative development of a text extraction system in which the best-coverage rules are discovered quickly, taking maximal advantage of whatever amount of effort the user is willing to invest in training components of the system.

4. *Many IE systems favor precision at the expense of recall.* There is a generally recognized trade-off between precision and recall in that points in precision may typically be purchased at the cost of some points in recall, and vice versa (see Section 1). IE systems generally have tended towards higher precision and lower recall, in part because many systems favor rules that extract many instances of a target item without introducing too much noise; that is, high-precision rules. Rare-case rules are not deemed reliable enough from which to generalize and high-coverage rules often introduce too much noise into the system to be worth the trouble of inclusion. OBIE exploits information gathered from the user during bootstrapping and the organizing structure of the ontology to generate suites of classifiers capable of shoring up the precision of rare-case and high-coverage rules. This permits OBIE to favor recall without paying the classical penalty of significantly reduced precision.

Over the course of the Phase I research, the following areas were addressed in approximate order:

- **Ontology Development** – An ontology accessible via a rudimentary API was developed capable of representing entities, events, and their relationships through abstraction and compositional hierarchies. A simple ontology viewer was developed and a sample ontology defined for use in processing texts from the domain of Latin American terrorist news articles featured in MUCs 3 and 4.

- **Document Processing** – A filter to convert texts into lower case (needed by the syntactic parser) and to strip off identifying header information was written in Perl. The Link Parser developed at CMU was adapted to perform phrasal segmentation and identification. Document browsers were written to view the results of syntactic and semantic processing.

- **Entity Bootstrapping** – Bootstrapping techniques developed by Riloff and Jones [1999] were adapted to learn dictionaries and extraction rulebases with respect to the domain ontology. User interactions in the form of simple, non-expert classification tasks were developed to capture user semantic information within the ontology and were integrated with the adapted bootstrapping cycle.

- **Classifier Development** – A key to incorporating high-recall extraction rules into OBIE was the development of several classifier suites capable of enforcing high precision on entities and events extracted by those rules. Classifiers were developed that relied on lexical similarity, pattern usage similarity, and sentence similarity between extracted literals and training examples captured within the ontology during bootstrapping.

- **Event Bootstrapping** – The bootstrapping techniques developed for entity extraction were extended to perform event bootstrapping. The additional task of role assignment was developed to learn the applicability of rules to compositional links within the ontology. The interplay between event and entity bootstrapping was exploited to permit work performed by one task to cross-seed the other.
- **Semi-Automatic Ontology Discovery** – Preliminary investigations were made to characterize the ability of OBIE to automatically discover some of the ontological structure of a new domain. Results suggested that event structure could be discerned through clustering techniques but that entity structure was more problematic.
- **Semantic Tagging** – A semantic tagging application was developed to demonstrate entity extraction and to evaluate the effectiveness of the classifier suites developed to shore up precision. Positive results indicate that high-coverage rules can be successfully added to the system without paying a significant precision penalty.
- **Event Extraction** – An event extractor was developed to demonstrate simple event extraction and more complicated inter-sentence scenario template extraction. Simple node generators were written to generate instantiated nodes into English for result summaries. Marker passing algorithms were developed to efficiently focus ontological processing.

These research and development tasks are implemented in the fully functioning research prototype developed for Phase I. That prototype is the topic of the next section.

2.1 Prototype Design

Figure 3 shows the system architecture of the Phase I OBIE research prototype. To support rapid prototyping and research, Allegro Common LISP was used for the majority of the development effort. Some off-the-shelf C code (the Link Parser) was adapted for use in performing syntactic phrasal segmentation and was linked with the LISP image via a Dynamic Link Library (DLL). Perl text-preprocessing scripts were written to transform corpus texts into a machine-readable format.

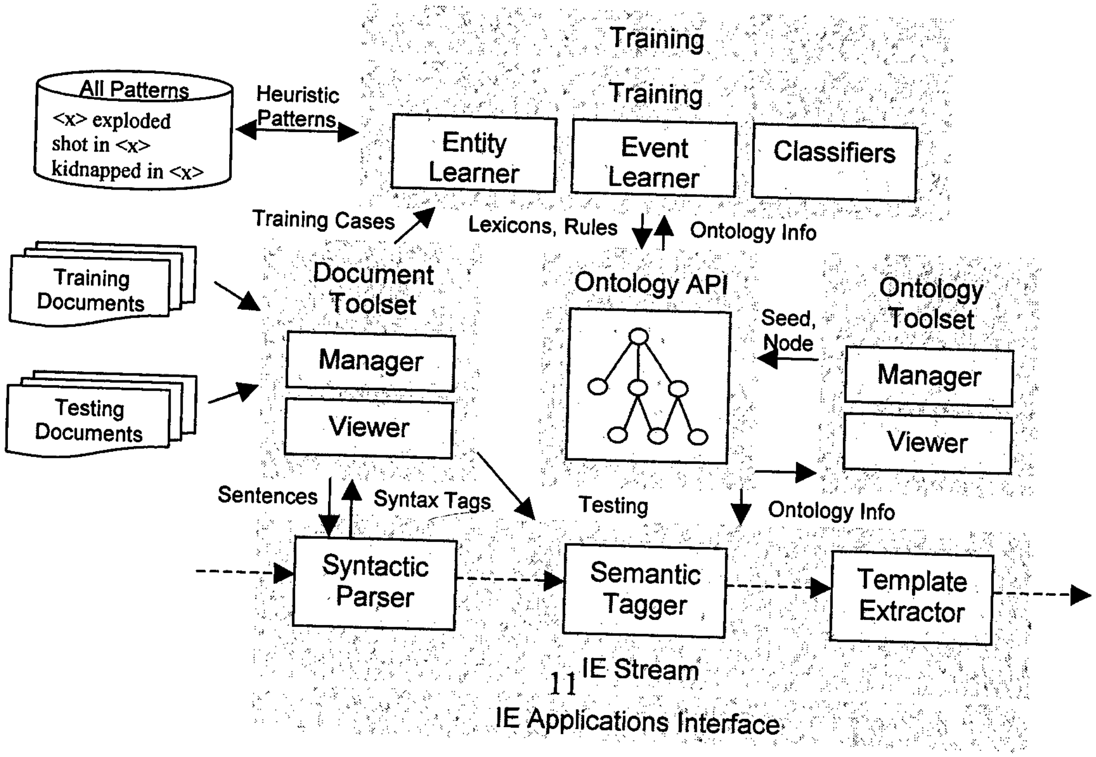


Figure 3. OBIE Phase I System Architecture

2.1.1 Ontology Development

The initial focus of OBIE prototype development was on implementing an ontology accessible via an API. This API permits different components of the IE stream to elaborate the ontology (by augmenting the lexicon or rulebase of a node, by adding new nodes, etc.) and to access the ontology during training or extraction tasks. A frame-based semantic representation was adopted which permits the specification of abstraction (IS-A) and composition (HAS-A) links between nodes. Lexical strings and extraction rules are also associated with nodes in the ontology. The semantics of a node are thus determined by the node lexicon that maps language into the node, the extraction patterns that extract or activate the node within some linguistic context, and the linkages between that node and other nodes of the ontology. (See Figure 4 for a diagram of a portion of the ontology used by OBIE.)

The top level of the ontology includes abstract representations for the basic building blocks of a domain, namely the entities and simple (atomic) events that compose it. Functional knowledge may be associated with nodes in the ontology for efficient processing. For example, functions to generate nodes into English are attached to the abstract entity and event nodes of the ontology. Scenario templates are represented by template nodes, which consist of a collection (similar to a script) of atomic events that are to be extracted as a group from text. Finally, a class of *scaffolding nodes* captures irrelevant semantic senses for each entity or event node undergoing training in the ontology. These scaffolding nodes, in conjunction with the entity and event nodes they mirror, are used to store information captured from user classification tasks in support of the precision-enhancing classifiers mentioned earlier.

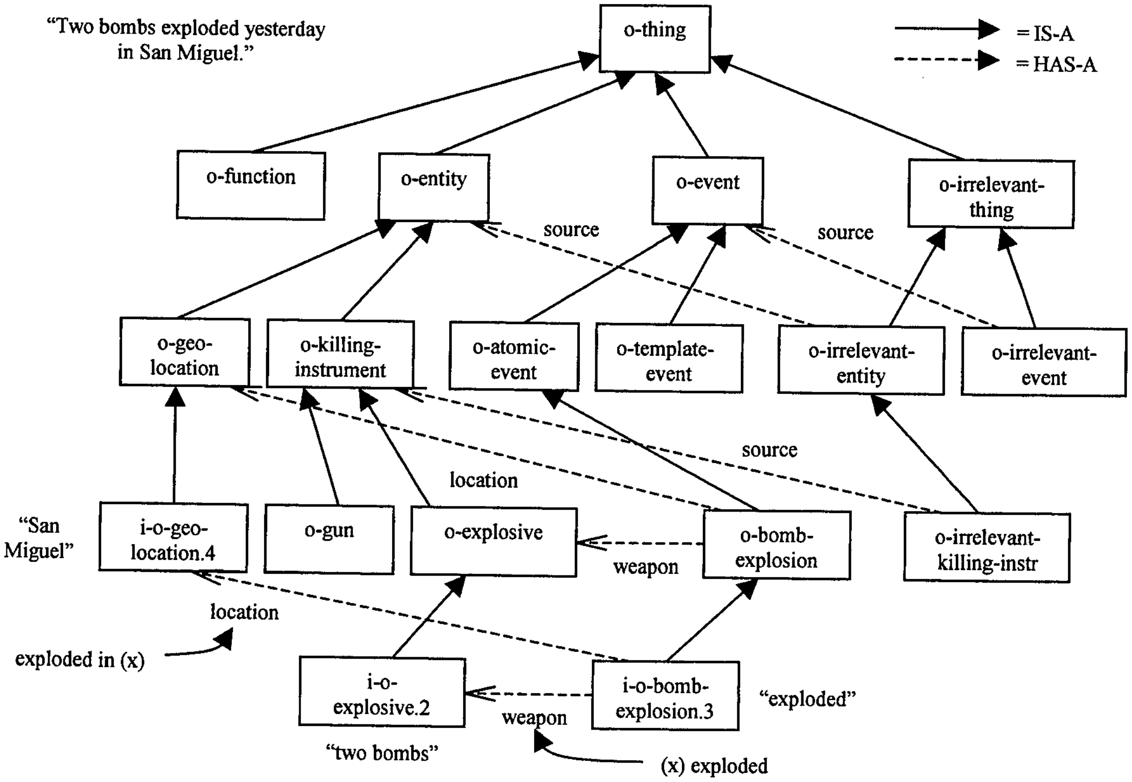


Figure 4. A Portion of the Prototype Ontology

The task of entity and event bootstrapping is to learn the linguistic mapping between natural language and the ontology. For example, "two bombs" refers to an instance of the node representing the semantics of an explosive device. "San Miguel" is a geographical location and should activate that concept in the ontology when encountered in input text. The verb phrase "exploded" indicates that a bomb explosion took place, an atomic event. The extraction pattern *exploded in <x>* extracts entities that potentially play the role of location in the bombing event frame. In ontology-based information extraction, the extraction task is thus recast as an ontology recognition task. Processing of the sentence "Two bombs exploded yesterday in San Miguel" involves instantiating a bombing event frame that captures the fact that two bombs were the weapon used and that the location was San Miguel. This explosion event itself is incorporated into a scenario template along with the other atomic events (e.g., people getting hurt) that fully describe terrorist bombings; this template is not shown in the diagram due to spatial constraints.

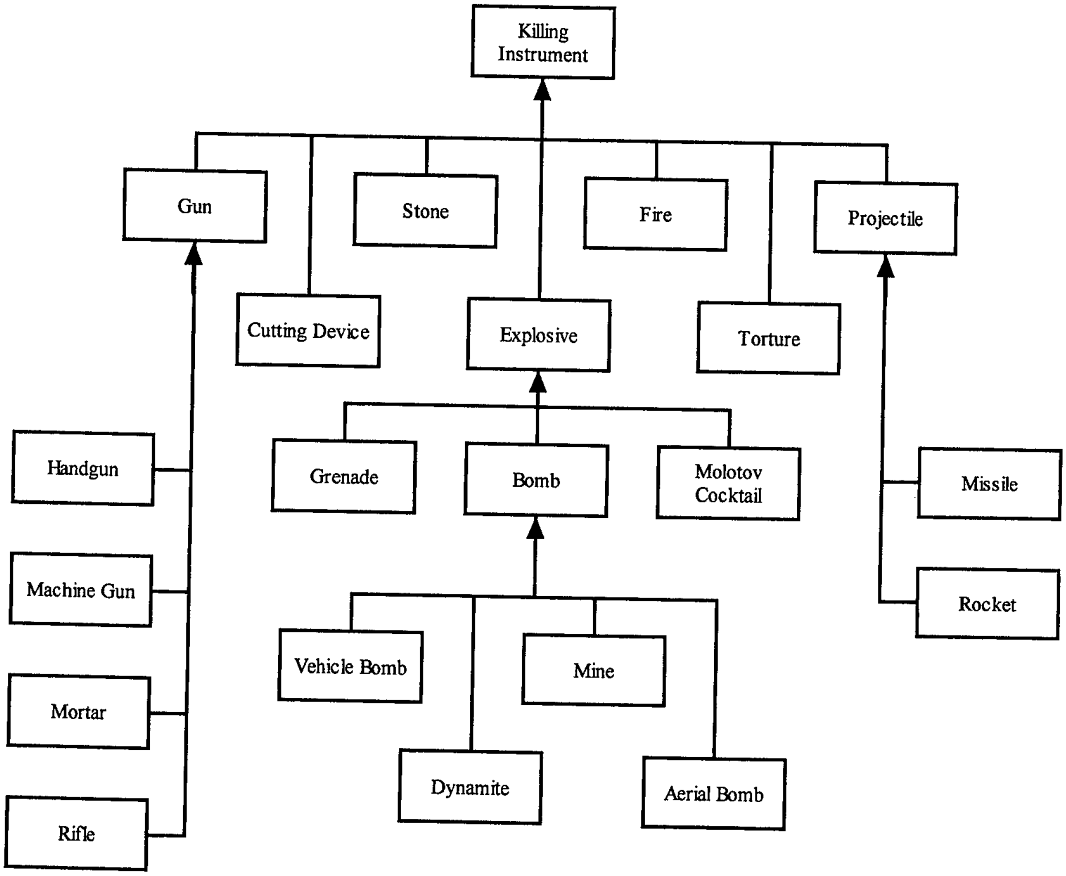


Figure 5. The MUC Killing Instruments Ontology

As a starting point for the ontology used in the prototype, we encoded the actual ontology for killing instruments defined for the MUC 3 and 4 participants. This hierarchy is shown in Figure 5. (Note that node i-o-explosive.2 in Figure 4 should really be i-o-bomb.2 according to the ontology of Figure 5 but has been changed for clarity within the diagram.)

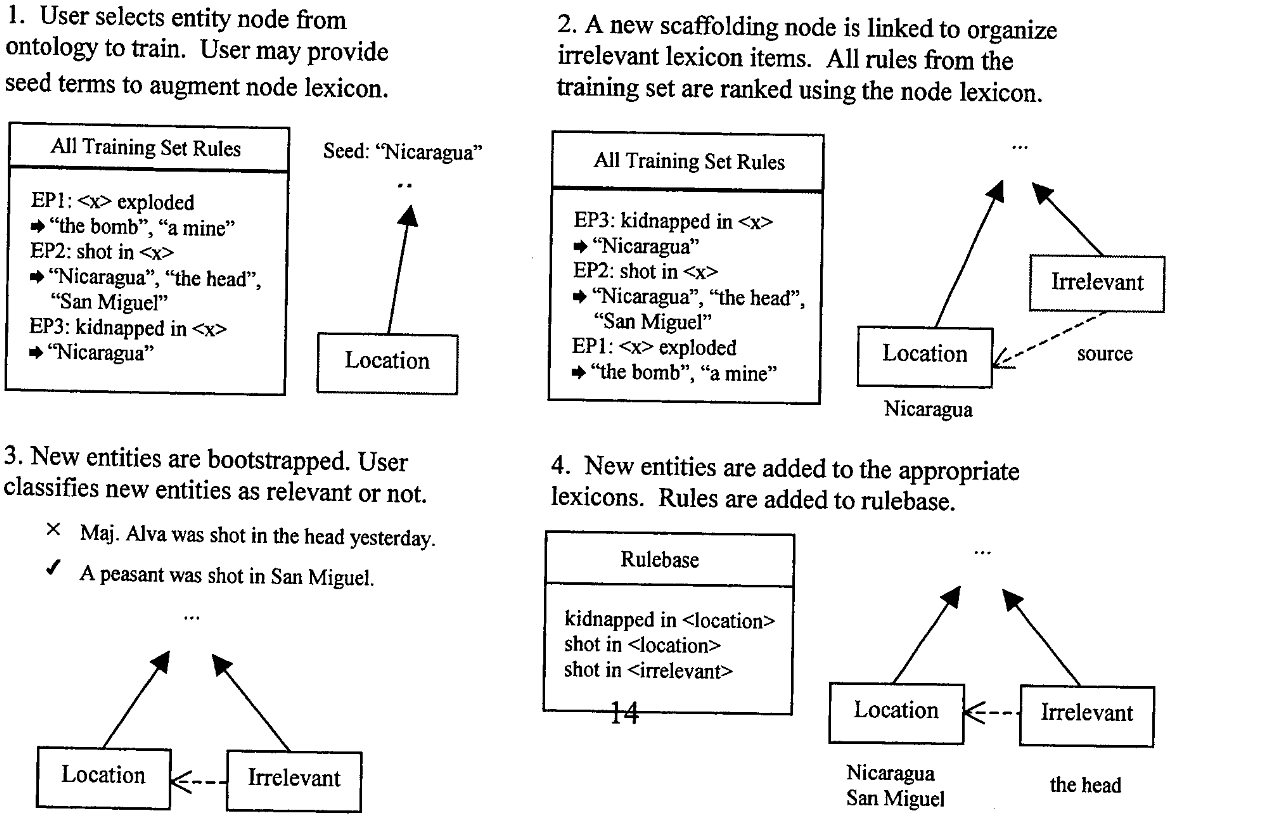
2.1.2 Document Processing

A document management system was developed to import and store training and test documents within OBIE. As a necessary precondition to bootstrapping, which relies on an exhaustive list of the phrasal patterns that occur in training text, a syntactic parser was needed to segment input sentences into noun, verb, object, and prepositional phrases. An off-the-shelf syntactic parser called the Link Parser developed at CMU [Sleator and Temperley, 1993] and available for research purposes was chosen for this task. (The Link Parser is virtually the only freely available, competent parser around.) This parser is used to segment the input text into syntactically tagged phrases. These phrases are then used to heuristically generate a library of patterns capable of extracting noun phrases from text.

The Link Parser is a full-sentence parser, which is non-optimal for our purposes (only partial parsing is necessary for phrasal segmentation) because full-sentence parsers inevitably cannot handle the richness of actual language use. The parser is thus a significant source of errors in our IE process. (Annoyingly, these are largely recall-diminishing errors, since significant sections of text are ignored or are incorrectly parsed.) Despite these errors, it certainly performed well enough to be of significant use to the research prototype. Fortunately, the techniques used by OBIE are quite robust and do not require perfect text parses.

2.1.3 Entity Bootstrapping

With document preprocessing in place, the next step was to adapt entity bootstrapping algorithms (developed by Riloff and Jones to learn dictionaries and extraction patterns; see Section 2.2.1) to the process of mapping natural language into OBIE's ontology. Riloff and Jones (henceforth called R&J) demonstrate how a dictionary and set of extraction rules can be automatically bootstrapped for a target entity (i.e., noun) class from a small set of seed words. We extended the R&J algorithms in two ways to perform ontology-based entity bootstrapping. First, we adapted the algorithms to learn *distributed* lexicons and rulebases. Each node of the ontology learns a dictionary of the literals that correspond to natural


1. User selects entity node from ontology to train. User may provide seed terms to augment node lexicon.

2. A new scaffolding node is linked to organize irrelevant lexicon items. All rules from the training set are ranked using the node lexicon.

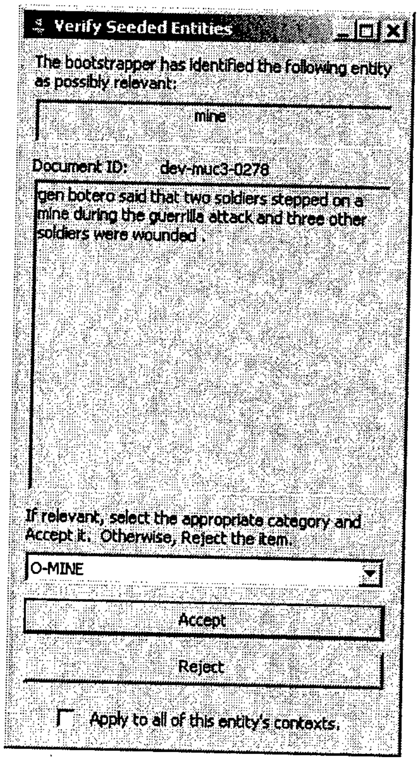
3. New entities are bootstrapped. User classifies new entities as relevant or not.

4. New entities are added to the appropriate lexicons. Rules are added to rulebase.

Figure 6. Entity Bootstrapping Algorithm

node and a set of extraction patterns that extract the node's lexicon items (and thus the node) from text. Figure 6 summarizes the adapted entity-bootstrapping algorithm.

We also used OBIE's abstraction hierarchy to specialize the extraction patterns to appropriate ontological levels. By restricting patterns to a specific level of the hierarchy, we enforce the optimal balance between precision and recall for an extraction pattern. As a concrete example, consider the pattern <*x*> *exploded*, where *x* is the entity being extracted by the pattern. In the Killing Instrument ontology shown in Figure 5, the rule should be specialized to the Explosive node. Through inheritance, all specializations of the Explosive node can access the pattern ("two bombs exploded", "the grenade exploded", "a mine exploded"). Abstractions of the Explosive node can also use the pattern, since more general terms sometimes follow a specific reference in text, as in "The bombs were planted under a car. The weapons exploded two hours later." (Note that the reference to weapons can be unified with the previous reference to bombs via the *Bomb IS-A Explosive* relationship if the literal "weapons" is in the Killing Instrument node's lexicon and the extraction pattern <*x*> *were planted* is also resident in the system. In this fashion, OBIE can perform referent disambiguation on the fly early in the IE stream instead of relying on decontextualized frame merging rules later in the stream to perform this step.) Siblings of the Explosive node do not have access to the rule, however. The phrase "the rifle exploded" has a passive semantic sense similar to that of "the melon exploded" and is different than the active sense of "the bomb exploded". We are likely only interested in the active sense for terrorist domain extraction. Of course, the system could be trained to specialize the rule to the level of the Entity node if we wanted these passive senses to be extracted. This is up to the user who trains the system to decide.



Verify Seeded Entities

The bootstrapper has identified the following entity as possibly relevant:

mine

Document ID: dev-muc3-0278

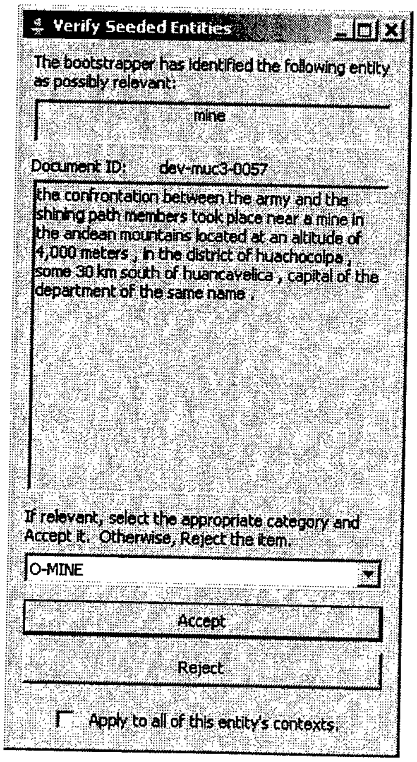
gen botero said that two soldiers stepped on a mine during the guerrilla attack and three other soldiers were wounded .

If relevant, select the appropriate category and Accept it. Otherwise, Reject the item.

O-MINE

☐ Apply to all of this entity's contexts.

(a)



Verify Seeded Entities

The bootstrapper has identified the following entity as possibly relevant:

mine

Document ID: dev-muc3-0057

the confrontation between the army and the shining path members took place near a mine in the andean mountains located at an altitude of 4,000 meters , in the district of huachocolpa , some 30 km south of huancavelica , capital of the department of the same name .

If relevant, select the appropriate category and Accept it. Otherwise, Reject the item.

O-MINE

☐ Apply to all of this entity's contexts.

(b)

Figure 7. User Classification of "Mine" in Context

The second extension to entity bootstrapping involved the insertion of the user into the bootstrapping process much earlier than R&J suggest. They have the user evaluate the final list of extraction rules once bootstrapping is concluded – a task which requires some system expertise by the user who must be taught what an extraction rule is and what constitutes "good" or "bad" rules. By having the user perform simple classification tasks in context (see Figure 7) we require *no system expertise* from the user.

We can also capture additional semantic information that is overlooked by R&J. Because the contents of a node lexicon drive the bootstrapping process, unambiguous lexical items are desirable when performing bootstrapping or the lexicons and rulebases become corrupted with irrelevant semantic classes. (R&J utilize a clever but somewhat ad-hoc meta-bootstrapping step to help offset this, but it remains a problem.) For example, R&J bootstrapping might add the literal "mine" to the weapons dictionary and use it to suggest additional weapon entries. However, the bootstrapping process then becomes quickly corrupted by multiple semantic senses of the word. Consider the two semantic classes shown for in Figure 7. Case (a) is an instance of a Killing Instrument mine; this semantic sense of "mine" should be added to the lexicon for the mine ontology node. Case (b), however, is clearly an irrelevant semantic sense of "mine" and should be rejected for use in the bootstrapping process, lest patterns that extract ore mines work their way into the Killing Instrument extraction rulebase.

By storing the second instance as a negative example (once the user so classifies it) attached to a scaffolding node, OBIE learns that multiple senses of "mine" exist and knows to be careful when dealing with the literal. For example, a Semantic Tagger should tag "mines" in the first sentence as a Killing Instrument but not in the second sentence. We realized that the semantic information captured by OBIE from the user via classification tasks could be used to automatically train a set of classifiers after bootstrapping to discriminate between different semantic senses of a word. The ontology facilitates this process by efficiently storing positive and negative training instances as part of the distributed lexicons learned during bootstrapping. It was also apparent that the classifiers we trained could be generalized to discriminate any literal *with respect to actual nodes in the ontology*. That is, given a specific literal like "the copper mine", "the land mine", or "the foreign debt crisis", a classifier could determine whether the literal was likely to be affiliated with the Mine node, the Gun node, or any other (possibly irrelevant) node. With this realization, it occurred to us that we could afford to allow low-precision, high-recall rules into the rulebases by using these classifiers to shore up their precision. When a rule from a rulebase is used to extract an entity, we pass the extracted literal to a classifier to verify that the item is in fact of the desired type.

2.1.4 Classifier Development

While not predicted by our original Phase I proposal, the importance of allowing high-recall, low-precision rules into OBIE's rulebases was immediately apparent in pursuit of the goal of high-recall extraction. The more rules there are in a rulebase capable of extracting an entity (at whatever precision), the greater the coverage and recall. We therefore extended the focus of our research to perform preliminary investigations into classification methods that could be used by OBIE to boost the precision of high-recall rules. Three categories of classifier were developed and tested (see also Section 2.2.2).

- **Lexical Similarity** – The first type of classifier was a simple dictionary lookup procedure that checked the lexicons of a target node and its scaffolding node (with negative instances) for the best match using right-to-left (head-based) relative similarity.
- **Pattern Similarity** – A second set of classifiers uses k-nearest-neighbor classification techniques based on vectors of extraction pattern feature similarity. These classifiers essentially find the node with lexicon items that are on average extracted by extraction patterns with the same frequency distribution as the input literal. This set of classifiers in effect looks at the pattern context for a literal over all training documents.
- **Sentence Similarity** – The third class of classifiers uses Naïve Bayes and bag-of-word frequencies to capture sentence contexts. These classifiers determine how similar the sentence containing the input literal is to the sentences that contain positive instances of lexicon items.

Generally, dictionary lookup was the most accurate, followed by pattern context, followed by sentence context. We also investigated serial committees of classifiers. In many cases, multiple views of an item are useful in performing disambiguation. In the ambiguous case of "mine" from the last section, the word is listed in both the mine node lexicon and its irrelevant scaffolding node lexicon. Simple dictionary lookup is thus not sufficient to perform classification of any given extraction of the phrase, so contextual classifiers must contribute to the disambiguation. An optimal committee composition mirrored the intuition of what people do when confronted with an unknown phrase. First, see if the head of the phrase is recognized from a dictionary. If not, see if the immediate pattern context is indicative of meaning. Otherwise, check the other words in the sentence for clues to meaning.

Figure 8 lists some results from a study of classifier performance. We trained the system to recognize the Killing Instrument ontology, spending about 2 hours of effort (using decidedly non-optimized algorithms) to build a lexicon of 283 relevant entities and a rulebase of 79 extraction rules. We then applied these rules to 200 new test documents and handed each extraction to the classifiers to determine if the item was a Killing Instrument or was irrelevant. The rules extracted 170 entities, 95 (or 55.9%) of which were classified by the author as irrelevant and 75 (or 44.1%) of which were Killing Instruments. Each classifier could attempt a classification or abstain when insufficient information was available.

	Tried	Abstain	Correct	Wrong	Precision	F-Score
RHM	0.706	0.294	0.706	0.000	1.000	0.828
NB	1.000	0.000	0.724	0.276	0.724	0.840
LPS	0.912	0.088	0.812	0.100	0.890	0.901
RHM + NB	1.000	0.000	0.906	0.094	0.906	0.951
RHM + LPS	0.958	0.042	0.929	0.029	0.970	0.964
RHM + LPS + NB	1.000	0.000	0.947	0.053	0.947	0.973
RHM + AI	1.000	0.000	0.982	0.018	0.982	0.991

RHM = Lexical Similarity LPS = Pattern Similarity NB = Sentence Similarity AI = Guess Irrelevant

Figure 8. Some Classifier Results for Killing Instruments

Dictionary lookup (RHM) within the Killing Instrument ontology and the scaffolding nodes that mirror it correctly classifies about 71% of the entities with perfect precision. ("Tried" is the analog of the *recall* statistic indicating the total proportion of cases covered by the classifier.)

Pattern and sentence context classifiers (LPS and NB, respectively) are both willing to classify more items (at 91% and 100% coverage), but they do so with decreased precision (89% and 72%). The committee of all three classifiers in sequence does better than any individual classifier, classifying everything and achieving almost 95% accuracy in doing so. The final row represents the heuristic "classify the item using the dictionaries and if it isn't listed, assume the item isn't relevant." The fact that this strategy works so well indicates that entity bootstrapping built up a pretty comprehensive Killing Instrument lexicon, one sufficient to identify all extractions of actual weapons. While this heuristic is effective for a relatively closed class of entities like weapons, open classes (company names, people, etc.) must rely more heavily on context classifiers.

2.1.5 Event Bootstrapping

Two innovations were necessary to adapt ontology-based entity bootstrapping to ontology-based event bootstrapping. First, patterns associated with verb phrases must be identified, as verb phrases instead of noun-phrases comprise an event's lexicon. These patterns extract a set of entities that can still be used to perform bootstrapping in much the same way that an entity's lexicon is used to bootstrap an entity. The entities that comprise the roles of an event are also pertinent and can be added to this set for use in ranking extraction patterns. This idea is captured in Figure 9.

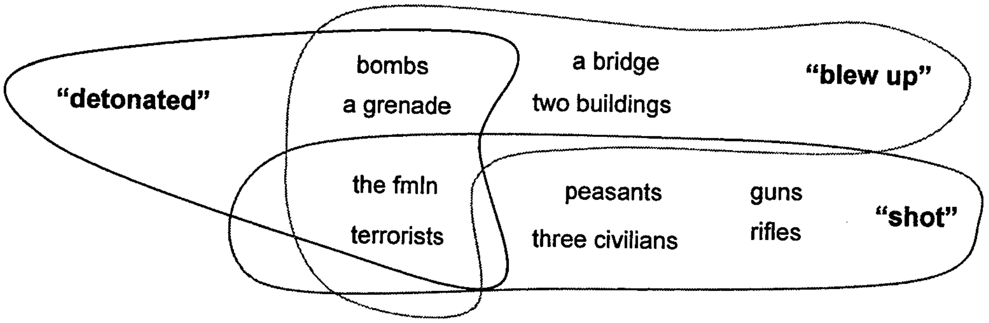
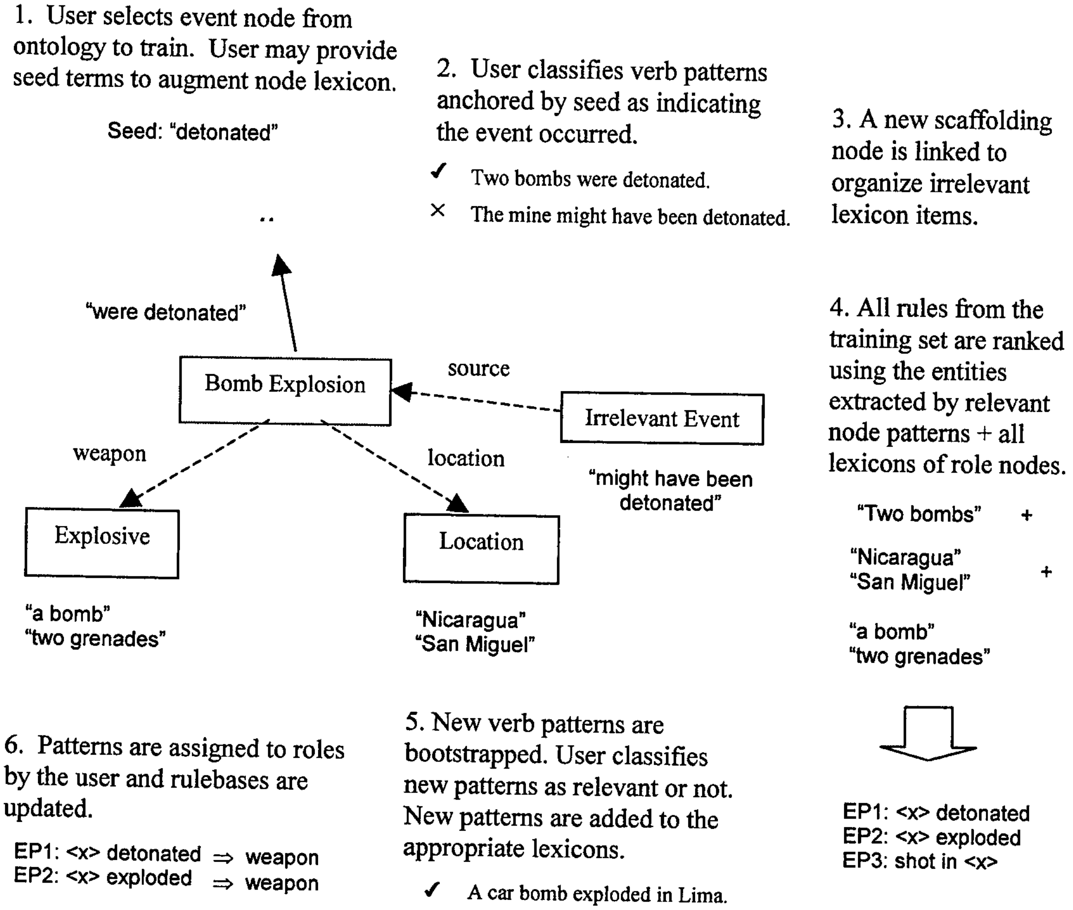


Figure 9. Role Entity Overlapping in Event Bootstrapping

Assume that the lexicon for the explosion event has already been seeded with the phrase "detonated". Two roles are represented by the entities associated with the phrase, the perpetrator of the detonation and the weapon detonated. We would expect that other phrases that are associated with the bombing event would share the same roles and thus would have high overlap with the entities the seed phrase extracts. We see that "blew up" has significant overlap in extracted entities (due to shared roles), more so than another pattern like "shot". By ranking patterns based on the entities extracted by event patterns plus the entities associated with entity lexicons from the roles of the event, OBIE can quickly identify new patterns that have a similar semantic function to verb phrases in the event lexicon.

The second extension necessary to perform event bootstrapping is to learn the mapping between extraction patterns and the roles of the event to which they correspond. Because OBIE utilizes single-slot extraction patterns, each pattern will be typically associated with a single role of the event (or it will extract semantic classes the user considers irrelevant to the event). The user is asked to perform a categorization task as part of the event training process for role learning. This task presents to the user a list of the entities extracted by a target pattern and a list of the available roles in the event (e.g., location and weapon). If any of the items in the list are part of a role category then the user is asked to specify that category and to classify each unknown entity in the list with respect to that category. This process is sufficient to allow OBIE to learn which roles are mapped into by each relevant extraction pattern. The final algorithm for event bootstrapping developed for the prototype is summarized in Figure 10.



1. User selects event node from ontology to train. User may provide seed terms to augment node lexicon.

2. User classifies verb patterns anchored by seed as indicating the event occurred.

✓ Two bombs were detonated.

× The mine might have been detonated.

3. A new scaffolding node is linked to organize irrelevant lexicon items.

4. All rules from the training set are ranked using the entities extracted by relevant node patterns + all lexicons of role nodes.

5. New verb patterns are bootstrapped. User classifies new patterns as relevant or not. New patterns are added to the appropriate lexicons.

✓ A car bomb exploded in Lima.

6. Patterns are assigned to roles by the user and rulebases are updated.

EP1: <x> detonated ⇒ weapon
EP2: <x> exploded ⇒ weapon

Figure 10. Event Bootstrapping Algorithm

2.1.6 Semi-Automatic Ontology Discovery

The success of the classifiers at identifying the relevance of entities with respect to a given ontology node and the manner in which role overlapping helps bootstrap events suggests that perhaps clustering techniques might be able to discern some of the relationships that structure the domain of interest. That is, clustering might help to automatically elucidate the structure of the ontology. While not specifically a recall-enhancing activity, this ability could assist a user in defining a domain ontology for extraction. Such a tool would be a useful feature for a commercial version of the system. We therefore invested a short amount of time on an implementation of pattern-based agglomerative clustering.

This clustering technique was applied to event nodes that were trained via bootstrapping. Visual inspection suggested that the role structure of an event could indeed be discerned within the resulting clusters. Specifically, clustering on the bomb explosion event yielded clusters suggestive of structural targets (bridges, buildings), locations (Bogota, Medellin), casualties (peasants, nuns), and weaponry (explosives, bombs). The clusters were not entirely well-defined, indicating additional techniques and/or user-guidance would have to be integrated with the technique to achieve better results. The possibility of semi-automatic event structure induction raises the hope that the definition of scenario templates can be derived from text automatically under user supervision, a central goal for user-driven information extraction systems [Wilks and Catizone, 1999].

We also ran the clustering algorithm on the Killing Instrument lexicon to see if we could recover some of the ontological structure defined in the MUC ontology of Figure 5. While we were able to discern the basic division between guns and explosives, little additional structure could be recovered. This is likely because linguistic pattern usage is fairly similar across levels of the Killing Instrument ontology, making fine-grained structure hard to recover via this technique.

2.1.7 *Semantic Tagging*

To test the techniques developed thus far, two applications similar in function to traditional IE system components were implemented. The first of these was a semantic tagger, capable of taking an arbitrary entity node from the ontology and tagging it (and its children) within new input text. This component takes a parsed input test document from the document manager and applies the rulebases for all nodes in the subtree rooted at the ontological node of interest. If any of these rules extract a literal from the sentence, that literal is then classified by the classifier suites. Recall that each rule in the rulebase is specialized to the nodes that it extracts. For example, <*x*> *detonated* might be specialized to the Bomb, Mine, Explosive, and one or more (irrelevant) scaffolding nodes based on the training data. Each of these nodes becomes the root of a subtree in the ontology that the classifiers will consider as candidates for the literal. Ontology search is thus constrained quite efficiently. If the classifier identifies the best fitting node of the ontology as being the node of interest or a specialization of the node of interest, the node is tagged with the semantic class (i.e., the ontology node name). Otherwise, the extraction is ignored.

The semantic tagger also served as a testbed for evaluating the classifier suites (see Section 2.1.4). The interface to the tagger permits the user to choose which classifiers to employ in performing the semantic tagging. We could thus evaluate the relative accuracy of each classification method and of different combinations of classifiers. The data for Figure 8 was generated via this interface.

To verify that the addition of increasingly more imprecise rules to OBIE's extraction databases would increase recall while not significantly harming system precision (thanks to the classifiers), we ran another set of tests on the Killing Instrument ontology. We started with the bootstrapped lexicon used in the Figure 8 experiment and deleted all of the extraction rules that had been trained up. We then relearned the rules incrementally and plotted the gains in recall and the loss in precision entailed by adding each new set of rules. We ran the experiment over a test set of 50 documents. The author exhaustively catalogued all Killing Instrument references within that text, discovering 71 instances. Visual inspection of the parsed documents indicated that only 53 of those instances could physically be extracted from the text via extraction rules. The remaining 18 were either completely missed by the parser or the phrases that would normally anchor the extraction rules around those entities were absent due to parse errors. (This will be mitigated in Phase II by better parsing techniques and by secondary dictionary scans of text missed by the parser.) Recall statistics are therefore assumed from a basis of 53 total instances.

Figure 11 summarizes the results of this experiment. A total of 91 patterns were learned in six increments. (OBIE's highly non-optimized rulebase learning algorithms are polynomial with respect to lexicon size due to a constant re-ranking of rules during every bootstrapping step; each set of patterns added to the rulebase thus significantly increased processing time. While many additional patterns remained for inclusion, we stopped at 91 rules for this experiment due to time constraints. If implemented correctly, the rule-learning algorithm is theoretically linear in time.) The diagram on the left shows the total number of extractions after each addition of rules. The relative number of relevant (e.g., Killing Instrument instances) and irrelevant extractions is also indicated. Note that as additional less-precise rules are added to the system, more irrelevant items become extracted in relation to the number of relevant items.

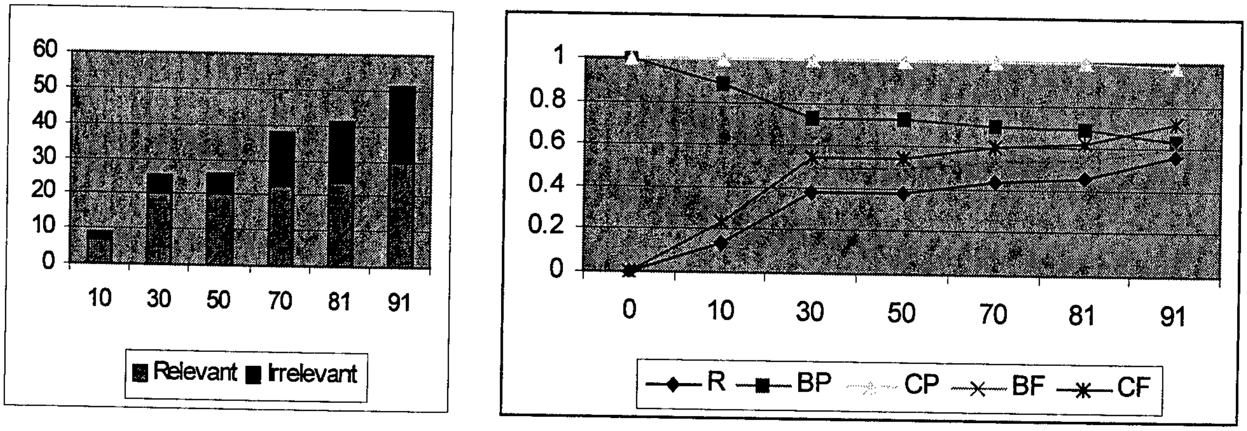


Figure 11. Behavior as Extraction Rules are Added to Rulebases

The right side of Figure 11 shows the interplay of recall and precision as rules are added. The lowest line (labeled "R") represents the recall, which as expected increases as more rules are added to the rulebase. The line labeled "BP" represents the baseline precision of all extracted items *without the classification step*; that is, all items are assumed to be relevant since no means to disambiguate them exist. As less precise rules are added to the rulebase, the precision drops with the increase in irrelevant items extracted. The "BF" line is the baseline F-score. Note the classical tradeoff between recall and precision, with the F-score balanced in the 50% range. The line labeled "CP" represents the precision of all extracted items *with classification* by the lexicon/pattern/sentence committee of classifiers (the RHM + LPS + NB classifier of Section 2.1.4). Perfect precision is maintained until over 80 patterns are added to the rulebase. The "CF" line represents the F-score computed with the augmented precision. It is clear from this diagram that the strategy of adding broad-coverage rules to enhance recall while shoring up precision through classification is effective in practice.

2.1.8 Event Extraction

The second application to be implemented was an event extractor, similar to the scenario template extractors of typical IE systems. The OBIE extractor had two tasks: recognize and extract from text atomic events (usually corresponding to verb phrases) trained via event bootstrapping, and then pull together the atomic events that compose a scenario template event (possibly across sentence boundaries).

The recognition of atomic event nodes was performed in a manner similar to that of entity recognition during semantic tagging. The rulebases associated with atomic events are used to recognize events from text, and rudimentary event classifiers (similar to the entity classifiers but not nearly as developed due to time constraints) are used to confirm that the actual event phrase in context corresponds to a relevant (rather than irrelevant) node. Once an indicator is recognized for an event, the roles of the event can be predicted to occur in the text. When this prediction is made, a check is performed to see if the predicted nodes have already been activated within the ontology during processing of the current document. If so, the atomic event node is informed of the fact. Otherwise, the prediction is stored with the predicted node. If that node is activated during the processing of subsequent sentences, the predicting atomic event is again informed. Once the document has been processed completely, all events pull together the results of their fulfilled predictions and instantiate themselves with all applicable frame slots filled in.

A spreading activation model in combination with marker passing algorithms (see Section 2.2.3) is used to efficiently implement these activation and prediction mechanisms. When an entity or atomic event is recognized within the ontology via the classified results of extraction rules, an activation marker is placed on that node in the ontology. This activation spreads up the abstraction hierarchy to all generalizations of the node. Event predictions are performed by passing prediction markers through the compositional hierarchy of the event. Collisions of activation and prediction markers indicate that a prediction may have been fulfilled (subject to certain lexical adjacency and semantic constraints). For example, the prediction that an explosive will be found in the text will be satisfied if an instance of a bomb is encountered, since activation spreads from the bomb instance up to the Explosive node, where a collision between the two markers occurs.

These techniques constrain processing to relevant areas of the ontology. Natural language is used as the insertion point into the ontology and the structure of the ontology itself constrains subsequent processing and permits the correlation of related information. The elaboration and sharing of ontologies across domains is an important goal for the overall OBIE system; this will potentially create quite large ontologies. Efficiently limiting the scope of search and activation in the ontology is therefore vital for a scalable system. These techniques thus contribute to the scalability of our proposed architecture.

The recognition of scenario templates (also called template events) from text occurs in a similar manner. Each template script has an anchoring event that indicates that the template should be activated. For example, the Bombing scenario template used by the prototype consists of two atomic events, the Bomb Explosion event and the People Harmed event. The Bomb Explosion is considered to be the anchor for the template, since it is a necessary component of any instantiated template (people get harmed for all sorts of reasons, not just through bomb explosions, so People Harmed is not an anchoring event). A permanent prediction marker is therefore passed through a role link to the Bomb Explosion node. When an atomic Bomb Explosion event is activated through text processing, the Bombing template becomes activated through the collision of the prediction and activation markers and non-anchoring roles in the template are predicted. Once document processing finishes, the template events collect together instantiated components and an instance template is created.

To control the combinatorial problems of multiple events of the same type activating from an input document, a simple heuristic is used in the prototype to constrain the events of a template to cluster together within adjacent sentences of text. (Semantic constraints resident within node definitions also limit the fusion of activated nodes; all components of the larger template must be semantically compatible with each other.) A sentence that contributes nothing to the template essentially terminates collection of role fillers for that template. An important issue to be addressed in Phase II is the actual recognition of topic shifts during text processing via a more sophisticated discourse model. Such topic shifts should be the actual stimulus that terminates open event templates. While the Phase I heuristic worked fairly well in practice, it is far too simplistic to capture the actual complexities of input text.

As a final aid in viewing the extracted templates, events, and entities, we attached functionality to those nodes in the ontology. This functionality generates a canned summary of the node in question in English, which can then be displayed to the user. In practice, the precise linguistic patterns that map into a node should be used (assuming sufficient syntactic knowledge is resident within the system) to map back out of a node into English. However, generation is not relevant to the goals of the Phase I project and was thus not supported in any depth. We note merely that the structure of the ontology and the existing linguistic information captured during bootstrapping already lay the foundation for more sophisticated generation and summarization capabilities.

Without a complete information extraction stream in place to support processing, meaningful recall statistics are difficult to characterize for event extraction. After approximately forty-five minutes of training, 9 out of 16 (56%) event bombings were recognized in a testing corpus of 50 documents. Three limits to better event recall keep this metric artificially low within the Phase I prototype. First, parse errors caused several events to be missed because the explosion description was erroneously parsed. Second, several event descriptions required some inference. The fact that a bomb placed by terrorists causes some damage implies that an explosion took place. While inference generation is an important aspect of semantic processing, it was not a focus of investigation within the Phase I prototype. Finally, events referred to by noun descriptions (e.g., "an explosion took place") are not handled by the prototype due to its artificial distinction between event and entity node lexicons. In Phase II, event and entity node activation and training will be generalized and this distinction removed.

2.1.9 Prototype Demo

For the final briefing of this contract, we prepared a prototype demonstration. This demo consists of a half-hour exercise of all of the components described thus far. A training set of 400 documents is loaded into the system. The system is initialized to create a database of all possible extraction patterns from the input text. The Killing Instrument concept is seeded with three phrases: "bomb", "grenade", and "mine". The user performs the necessary classification tasks to disambiguate and specialize these terms in context. Entity bootstrapping is then performed on the Killing Instrument node and a rulebase of six extraction patterns is generated from the result.

To demonstrate the interplay of different components of the ontology, the Bomb Explosion event is next bootstrapped *without requiring any seed terms*. The fact that the weapons role has already been trained via the Killing Instrument hierarchy is sufficient to quickly discover several lexicon items and three extraction rules for bomb explosion events via simple user classification tasks. Just as ontological training during entity bootstrapping can be leveraged during event bootstrapping, we also demonstrate the converse by bootstrapping the Geo Location entity, a role of the bomb explosion event, again *without seeding*. From this process, a node lexicon and four extraction rules are learned for geographic locations.

We next demonstrate that seeding can be used for events as well as entities by providing a single seed term for the People Harmed atomic event, "died". This allows us to bootstrap a lexicon (including different tenses of "wounded") and three extraction rules. We finally generate extraction patterns for the People entity from the results of that event training without actually bootstrapping (the event training captures enough information to immediately generate several extraction patterns for People). The ontology can be browsed to view the results of this training process. We also demonstrate how clustering can be used to discern the structure of the ontology automatically at this point.

With training done (one iteration of this whole process takes about 15 minutes), we induce the various classifiers from the ontology. A testing set of 200 documents is then loaded, and we run the semantic tagger application. The use of different classifiers and their varying results is demonstrated during semantic classification. A visual document browser allows tags to be viewed in the training documents. We also demonstrate how different levels of the entity hierarchy can be tagged by changing between all Explosive instances and just Bomb instances.

We conclude the demo by running the event extractor. This application processes the test documents and creates approximately 16 instances of the Bombing template. We visually browse the results via the English generators attached to the event nodes of the ontology.

A demo walk-through script describing these steps is contained in Appendix A.

2.1.10 Prototype Summary

As mentioned, the prototype demonstration consists of a twenty-five minute exercise of all of the components described thus far. We bootstrap and demonstrate a complete extraction process over a predefined ontology for a bombing template consisting of a bomb explosion and of people getting hurt while requiring only four seed terms and some non-expert classification tasks. Just this amount of work is sufficient to extract 16 templates from a test set of 50 documents. The user is then free to return to the training documents and to resume bootstrapping of the entities and events to better increase recall (through the learning of more extraction patterns and lexicon entries) and precision (by providing the classifiers with more training data). Alternately, the user might flesh out the ontology to represent additional items to be extracted, or to move on to a different analytical focus.

As a rough baseline, even given the decidedly non-optimized implementation of the algorithms used by OBIE, approximately two hours is sufficient to learn a fairly complete entity lexicon consisting of almost 300 literal phrases for Killing Instruments. By actively ranking the patterns to favor broad coverage during bootstrapping and by the cross-leveraging of training that the ontology facilitates, we maximize the gains made by user investment in training and support an iterative approach to text analysis. The ability to add increasingly tenuous extraction patterns shored up by classifiers to the system increases the recall of the extraction system without paying a significant precision penalty. We feel the Phase I research prototype thus successfully validates OBIE's approach to user-centered, ontology-based information extraction.

2.2 Description of Methodologies

2.2.1 Bootstrapping

Bootstrapping is a general mechanism for improving a learner using unlabeled data [Jones, et. al., 1999]. It is an iterative method, in which a small set of user-labeled data is allowed to seed a clustering algorithm that takes as input the corpus of unlabeled text. The resultant clusters are used to estimate new labels from the unlabeled data. These labels are evaluated for applicability; accepted labels are added to the seed set and the process repeats until some threshold of diminished returns is reached.

[Riloff and Jones, 1999] describe a bootstrapping algorithm for generating a domain-specific lexicon and ruleset capable of extracting a class of domain entities. A user provides an initial list of examples for a class to be extracted. Using an existing program called AutoSlog [Riloff, 1993], an exhaustive list of lexical patterns capable of extracting all noun phrases is collected from the domain corpus. Patterns that extract any element in the given seed set are identified. These patterns are typically general enough to extract other elements from the corpus. An assumption implicit to this process is that the lexical and semantic constraints encoded by these patterns will extract semantic classes of which the seed set elements are members. Non-seed elements extracted by the patterns are scored for applicability and the top few elements are added to the seed list. This process is iterated until a threshold is passed.

As we demonstrated during Phase I, this algorithm can be generalized to extract events and role relationships from text. In fact, any concept that is correlated with specific configurations of natural language should be amenable to the bootstrapping process. We also discovered that the bootstrapping of one conceptual type (e.g., entities) can automatically seed the bootstrapping of other conceptual types (e.g., events), minimizing the amount of seed information the user must provide.

2.2.2 Classification

Classification may be seen as the task of assigning group membership to an instance based on similarity metrics between group members and the instance. The induction of classifiers from data sets of annotated instances is a central problem in machine learning. In OBIE, classifiers are used to assign instances extracted from text via broad-coverage extraction rules into appropriate ontological structures. As such, classification is one of the foundations supporting high-recall extraction, since it permits the system to use highly general or rare-case extraction rules without the decrease in precision that use of such rules typically invites.

Numerous approaches to the problem of classification exist. Phase I investigated three classification techniques. The first was a simple *dictionary lookup* scheme in which node lexicons bootstrapped during system training are used to interpret new extraction instances. Right-to-left phrasal head matching is used to determine the relative similarity of new instances to previous instances in both relevant and irrelevant lexicons. Depending on the quality of the lexicons (as determined by the amount of time spent building them up through training), this technique quickly and efficiently classifies a majority of new instances. Novel literals not yet seen in training data must be classified by more statistical methods.

Naïve Bayes classifiers [Mitchell, 1997] have proven successful for classifying text-based documents during information retrieval tasks. This second classifier type takes a set of training data and learns the conditional probability of each attribute A, given the class label C. Classification is then done by applying a Bayes rule to compute the probability of C given the particular instance of $A_1, \ldots, A_n$ and then predicting the class with the highest posterior probability. This computation is rendered feasible by making a strong independence assumption: all the attributes A_i are conditionally independent given the value of the class C. We adapted this technique to classify extractions based on word frequency features of the sentence containing the extracted item in relation to other sentences with extractions known to correspond to a class of interest.

The third type of classification technique used in Phase I is that of the *k-nearest neighbors* approach [Mitchell, 1997]. The linguistic patterns in which an extracted literal occurs within training data are used to create pattern feature vectors describing the linguistic context of the literal. Each lexicon associated with a node in the ontology contains multiple such literal items. A group average feature vector in pattern space can thus be computed for each node of the ontology (forming a sort of pattern neighborhood in pattern space). Classification consists of determining which node has the most similar pattern vector to that of the instance or, put another way, which lexicon items in pattern space represent the closest neighbors to the instance being classified.

2.2.3 *Marker-Passing Algorithms*

Marker-passing algorithms are a class of techniques used to constrain search over large semantic networks. A marker is simply a data structure containing useful information (such as a pointer to the originating semantic node). Markers are passed from node to node via the links that structure an ontology. In the Phase I prototype, markers were used to activate all abstractions of a node recognized from text. Markers are also used to store predictions about other semantic concepts that are expected to appear in text. The collision of activation and prediction markers represents fulfilled expectations and permits the efficient correlation of activated concepts without requiring additional search over the ontology. Marker-passing algorithms thus facilitate scalable semantic representations since processing is focussed within the representation itself.

Marker-passing techniques have also been used to perform semantic inference [Norvig, 1989]. Using this technique, markers are passed along different link types of a semantic network. Collisions between markers from different nodes indicate a semantic relationship between those nodes; the exact relationship is determined by the topology of the links traversed by the markers. Techniques of this nature may be incorporated within the Inference Generation module of the Phase II system to perform automated semantic inferencing using the structure of the ontology.

By passing activation markers up the abstraction hierarchies of activated nodes, co-reference resolution is facilitated. Assume the ontology of Figure 5 and consider the sentences "Two ak-47s were confiscated from a Lima apartment. The weapons were believed to have been used in last evening's attack." The instantiation of a semantic node representing the ak-47 rifles causes activation of the Rifle hierarchy, including the Gun and Weapon nodes. When the reference to weapons in the second sentence activates the Weapon node, the Reference Resolution module can unify the two references by examining the activation markers on the node and recognizing that the weapons reference likely refers to the two rifles. In a similar manner pronoun or anaphoric disambiguation can be efficiently performed by examining patterns of activation over the ontology during text processing.

2.2.4 Shrinkage

Shrinkage [Freitag and McCallum, 1999] is a statistical technique whereby probability estimates for children in a hierarchy borrow from their ancestors through interpolation. Weighted frequency probabilities from a node's abstraction hierarchy can be incorporated into an estimation of a node's frequency distribution. This technique has shown considerable promise in dealing with sparse data, since probability estimates garnered from data at various levels of generality can be leveraged to help characterize rare-case instances.

Shrinkage has been used quite successfully with Hidden Markov Models for part-of-speech tagging and speech recognition. The technique will be generalized within OBIE to similarly leverage probability distributions during any applicable statistical machine learning task, including implementations of the IE stream components and the classifier suites.

2.2.5 Semantic Parsing

OBIE draws considerable inspiration from the *case-based parsing* paradigm developed in Martin's Direct Memory Access Parser [Martin, 1989]. Case-based parsing recasts the traditional text parsing process as the recognition and activation of nodes in a semantic network during natural language understanding. In this approach, stereotypical patterns of natural language are typically hand-crafted into rules that form indices into a semantic case-library. This hand-crafting makes deployment into new domains difficult and leads to rather brittle processing systems.

This semantics-laden approach was largely abandoned in the early 1990's with the rise of more automatic, statistical approaches to language processing. The goal of natural language understanding gave way to the sub-discipline of information extraction, a more constrained and viable task. OBIE represents a fusion of these more recent statistical techniques with some of the earlier goals of deeper semantic understanding. Traditional statistical IE approaches combined with user guidance through bootstrapping are employed to learn the actual patterns of language use that index into the ontology, with the ontology providing a consistent basis from which to perform inference and limited semantic processing.

2.3 Literature Search

Automatic learning algorithms that make use of a concept hierarchy are rare (a few are discussed briefly below). Handcrafted, rule-based components more frequently use a custom-designed ontology (particularly in the form of abstraction hierarchies), but this makes them highly non-portable. While individual IE modules exist that rely on a semantic hierarchy of some sort, OBIE is the first IE system to incorporate an ontology as the central organizing backbone at all levels of the IE pipeline. It is also the first to suggest how such an ontology can be automatically constructed by the IE modules themselves. In doing so, OBIE provides a completely integrated mechanism to allow communication and feedback across *all* levels of the IE process, a capability lacking in other IE end-to-end systems.

One system that has attempted to allow at least some components to share information is FASTUS [Kehler, et. al., 1998]. FASTUS provides a structure called a *lattice* that allows multiple interpretations of an ambiguous linguistic phenomenon to be passed between phases in the IE pipeline. Because modules at different levels of the IE stream are handcrafted (rather than corpus-generated), heuristics can be added to each module to allow the processing of the lattice structure. The capacity of a lattice is more naturally provided by OBIE's ontology. Multiple interpretations of a phenomenon by an IE component could be simply represented by multiple active nodes in the ontology as processing flows from module to module down the end-to-end processing stream.

[Embley, et. al., 1998] propose an ontology-centered approach for performing wrapper induction, a close cousin of the IE task, for semi-structured documents on the WWW. They develop a parser, named entity recognizer, and a structured text generator guided by a system ontology to induce wrappers from a corpus of web pages. Their approach is in spirit similar to OBIE's and they report a significant benefit to the use of an ontology-centered approach. Their work provides evidence that the application of an ontological backbone to the IE processing stream can lead to significant gains in accuracy.

[Roth, 1998] proposes the idea of using a *single* learning algorithm for all levels of the IE task. Roth notes that multiple existing learning algorithms for disambiguation tasks can be recast as learning linear separators in the feature space of the domain. A sparse network of linear separators that employ the Winnow learning algorithm is used to solve three tasks: context-sensitive spelling correction, part-of-speech tagging, and prepositional phrase attachment. Roth speculates on the benefits of using a single algorithm across all levels of the IE task. It is a matter for empirical investigation as to whether or not a single algorithm is well-suited to all task domains. Somewhat contrary to that approach, OBIE allows each component to use any algorithm it feels is best suited to the task, providing a common representational structure to aid the algorithm and to permit feedback processes.

Researchers have used Hidden Markov Models for several IE subtasks, including part-of-speech tagging and named entity extraction [Glickman and Jones, 1999]. A technique suitable for combination with HMM models is that of *shrinkage*, which takes advantage of class hierarchies to combine relevance estimates in a weighted average across levels of an abstraction hierarchy [Freitag and McCallum, 1999]. This technique makes such estimates more robust in the face of sparse training data. Shrinkage has also been used in bootstrapping [Jones, et. al., 1999]. Because OBIE's ontology makes such class abstraction hierarchies available to all components of the IE stream, it is likely that other learning algorithms could benefit from the application of shrinkage techniques.

At the level of scenario pattern matching, PALKA [Kim and Moldovan, 1993] used an abstraction hierarchy to determine the appropriate generality of rules. Using a candidate elimination type algorithm, PALKA constrained rules by specializing them down the hierarchy or by generalizing them up the hierarchy in a manner similar to that proposed for OBIE. The Proteus system also uses a semantic concept hierarchy to specialize rules [Yangarber and Grishman, 1997]. OBIE draws its methodology for allowing the end-user to graphically specialize rules from its ontology from the Proteus system.

As guiding principles, OBIE will incorporate techniques from the ontology research community in creation of its ontology. [Guarino, 1997] provides useful insights into the formulation of a top-level ontology for information extraction and retrieval tasks. [Noy and

Hafner, 1997] provides a useful survey of different ontologies developed by AI researchers and the issues of expressiveness and portability that must be addressed.

[Woods, 2000] presents positive results for the creation of large-scale subsumption (i.e., abstraction) hierarchies from lexical and phrasal analysis of free text. By analyzing relationships among constituents of phrases and compound morphemes, lexical strings from text can be automatically placed at appropriate levels of generality within a hierarchy encoding subsumption relationships. The algorithms developed by Woods for automatically determining the level of generality for new lexical phrases may be adaptable for use by OBIE in mapping novel lexical items to appropriately specialized ontological lexicons.

2.4 Lessons Learned During Phase I

Our Phase I investigations and prototype development have proved quite valuable. Our explorations of potential techniques for applying semantic knowledge to the information extraction task and of attempting to increase recall through broad-coverage rule use have allowed us to characterize multiple interactions within our proposed architecture that together lead to increased extraction recall. In addition, the development of our limited prototype has provided us with a vehicle for testing our preconceptions and for highlighting the key challenges in providing this functionality. We list the primary results of our investigation below. Note that each of these findings has been folded into our Phase II approach discussed in Section 3.

- **Patterns of language use contain information exploitable for ontology-based IE.**

  Our bootstrapping algorithms allow the rapid deployment of OBIE into a new domain by discovering correlations of language use with ontological structures. This technique leads to the generation of lexicons and rulebases for use in performing the extraction task. While the literature has previously described techniques for entity bootstrapping, it wasn't clear at the outset that patterns of language use are sufficient to also bootstrap events and relationships between entities and events. Nor was it clear that rulebases and lexicons could be attached to ontological structures with ease. The development of ontologically-based bootstrapping algorithms for entities, events, and relationships has confirmed that language patterns contain sufficient information to support the extraction task at all levels of processing.

  In addition to bootstrapping, patterns of language use play a key role in developing classifiers to shore up the precision of high-recall rules. Several suites of classifiers rely on both phrasal similarity and sentence word-frequency similarity to classify extracted entities as either relevant or not to the task at hand.

- **Classifiers permit points of recall to be purchased by incorporating broad-coverage or rare-case rules without sacrificing precision.**

  Classifiers can be trained from the ontology using information captured during bootstrapping processes. These classifiers permit OBIE to use extremely general, broad-coverage, or rare-case rules during extraction tasks. Traditionally, using such rules to increase recall has meant a corresponding decrease in precision as more irrelevant items are extracted from text. By using classifiers to evaluate the results of rule extractions, we shore up this lost precision and reap the benefits that such rules afford.

- **Work performed by one component in training the ontology is of use by other components.**

  As we show during the prototype demonstration, work done to train entities can be leveraged directly by events that package those entities, allowing the user to bootstrap events without the need to seed the event nodes in the ontology. Similarly, event bootstrapping captures information about entities as a natural by-product that can be exploited by entity bootstrapping to learn entity rules without the need for additional user seeding. Both processes in effect cross-seed each other, allowing very rapid deployment into a new domain. For example, during the demo we bootstrap a complete extraction process capable of extracting 16 complex bombing events from a training corpus, including the weapons involved, the location, and the people injured, from just three entity seed words and one event seed word.

- **Automatic ontology and scenario template discovery from free text may be accomplishable via clustering techniques.**

  As part of our investigation, we applied agglomerative clustering techniques to entity and event node lexicons and rulebases in the ontology in an attempt to discern the ontological structure and relationships involving those items. Interestingly, event roles are largely differentiable via clustering. For bombing events, we could discern the roles of target, perpetrator, weapon, and location. These clusters are not completely well-defined, indicating that additional refinements or user-interaction may be necessary. These results are suggestive that the ontology itself might be induced from free text.

  Our results on entity clustering were less conclusive. While we could discern gross divisions in weaponry by clustering on weapon lexicons and rulebases (e.g., guns and explosives formed two broad categories), we couldn't discern any of the finer grained divisions present in our hand-coded ontology. We conclude therefore that patterns of language use *within* a semantic class may not have enough granularity to be exploitable through the attempted techniques.

- **Bootstrapping techniques incorporate high-payoff patterns quickly, maximizing use of limited user time.**

  The bootstrapping algorithms developed in the prototype have the useful benefit of bringing the most accurate patterns (i.e., those with the best ratio of signal to noise) to the forefront for user evaluation before less accurate patterns. The scarce resource of user training time is therefore maximized during system porting. This permits a user to quickly get a system up and going, and to then iteratively broaden the coverage of the system through additional training.

- **Simple contextual classification tasks are sufficient to capture useful parts of a user's semantic knowledge.**

  By requiring only simple classification or categorization tasks of the user in which lexical strings are presented in context for evaluation, OBIE assumes little domain and no system expertise by the end-user. This contrasts with IE systems in which users must become versed in rule representation languages or other arcane system features.

- **Spreading activation via marker-passing over the ontology is an efficient means of managing contextual predictions.**

  Because potentially large ontologies will be used with OBIE, it is crucial that all algorithms scale with respect to ontology size. The use of marker-passing techniques fulfills this condition by allowing the structure of the ontology itself to constrain processing to precisely those areas where it should occur. Natural language (in the form of extraction rules and dictionaries) serves as an index into the ontology during processing, further obviating the need for expensive ontology search strategies. Additionally, spreading activation techniques should permit inferential and reference disambiguation methods that similarly scale in the Phase II implementation.

- **Entity recognition is mediated by the roles the entity plays in events from text.**

  During the course of our Phase I investigation, we realized the deficiencies inherent in the simple ontological hierarchies that we employed for OBIE. The use of the packaging hierarchy to capture role information is a slight abuse of the semantics of composition. As such, we will likely add a role link type to the ontology in Phase II. Other link types useful in capturing other semantic relationships will be added to the ontology after a thorough requirements analysis in Phase II. As long as these link types are correlated with specific configurations of natural language, the bootstrapping techniques developed during Phase I should be adaptable for use in learning to recognize and extract those links.

2.5 Technical Feasibility

Several factors contribute to the assurance of the technical feasibility of the proposed OBIE system:

- **Scalability**. Scalability is a crucial requirement if OBIE is to be employed by military and commercial users. Two tacks are taken to ensure that OBIE scales with respect to ontology size. First, lexicons and rulebases are distributed throughout the ontology by attaching them to actual nodes and links within the system. In Phase II, classifier data will also be distributed in this manner. This obviates the need for expensive searches, since such information can be applied directly during node or link extraction. Second, marker-passing algorithms allow the structure of the ontology itself to constrain processing. When combined with the use of natural language patterns as indices into the ontology, processing is directed and restricted only to relevant areas within the ontology, mitigating the need for ontology search and thus ensuring scalability.
- **Consistency**. By wrapping the IE stream around the central backbone of the ontology, OBIE ensures that a consistent semantics is imposed on all IE modules. Additionally, components are free to share information and views regarding an emerging interpretation of input text. Additionally, each component can contribute different information to the ontology, using it as a sort of blackboard to guide processing. As a worse case, IE modules may ignore the ontology; this is simply the degenerate case of existing non-ontology-based IE systems.
- **Robustness**. The design of OBIE focuses on the development of robust strategies for training, classification, and extraction. Iterative bootstrapping algorithms permit natural language to be mapped onto ontological structures with increasing coverage as more training is performed. By forming committees of classifiers, the system can pool multiple viewpoints

to come to a consensus regarding the classification of entities, events, and relationships. Both ontology-based processing strategies such as inference and reference disambiguation are combined with machine learning approaches in performing information extraction. As the Phase I prototype demonstrates, the system is robust with regards to the numerous parsing errors introduced via an adapted, off-the-shelf syntactic parser.

- **Modularity**. Object-oriented design practices are used at all stages of the development process. The ontology is accessible only through an Application Programming Interface to ensure that all modules interact with it in a formalized and well-understood manner. Because IE stream modules interact only through the shared ontology, between-module interactions and unintended side-effects are minimized.

2.6 Conclusions

The OBIE project offers several innovations that together facilitate the primary goal of higher-recall information extraction and the secondary goal of developing a non-expert, user-centered, domain-portable, commercial text processing system:

- **A Tightly-Coupled Architecture** – Integration of the IE stream with a shared ontology reduces cumulative errors in the processing stream by providing a blackboard-like data structure that facilitates knowledge sharing, integration, and feedback between components during both training and extraction. Disambiguation is a joint and cumulative pooling of different viewpoints across modules of the IE stream.

- **Semantic Processing** – The ontology imposes a single, shared semantic view of the world. Ontological relationships (e.g., abstraction, composition, etc.) permit inference and co-referencing to be performed early and automatically in the extraction cycle. Non-shallow levels of information extraction, including role recognition and across-sentence relationships, are achieved by recasting the extraction process as a semantic recognition and extraction task. Automatic machine learning algorithms may be augmented to take advantage of relationships (like abstraction) specified by the ontology.

- **Bootstrapping** – Extensions to existing entity-bootstrapping algorithms permit not only entity but event and role learning, as well, all with respect to a given domain ontology. These techniques rapidly discover high-quality rules and lexicons using simple seed terms and patterns of natural language use inherent in the training text. The approach also captures the highest-payoff rules first, maximizing the benefits of potentially scarce user interactions, and allows an iterative developmental cycle. Any relationship between entities or events that is correlated with specific patterns of language use should be learnable via bootstrapping processes.

- **Classifier-Enhanced, Broad-Coverage Rulebases** – By building classifiers from information already captured as part of the bootstrapping process, OBIE permits high-recall but low-precision (e.g., broad-coverage or rare-use) rules to be incorporated into its rulebases. The classifiers use multiple contextual or lexical viewpoints to shore up and elevate the precision of such rules, ensuring that recall is not simply purchased at the price of precision.

- **A User-Centered Approach** – By allowing an arbitrary, user-provided ontology to structure the information extraction task and by relying on simple classification or categorization tasks performed from the user's perspective and according to the user's objectives, the user is placed firmly in the driver's seat of the text analysis and extraction process.

3 Phase II Design & Future Work

The goal of our OBIE system is to increase the accuracy of the information extraction task by expanding the coverage of the extraction process while still maintaining high precision. This will be accomplished through a synthesis of several techniques incorporated into an end-to-end information extraction system. First, we apply semantic knowledge to the extraction process through the use of a centralized domain ontology. This semantic processing permits inference and disambiguation not possible given access to only surface (syntactic) features of text. Second, we tightly couple all elements of the IE stream to the common representational structure of the ontology, permitting feedback and integrated processing absent in a completely modularized IE pipeline. This also minimizes the accumulation and amplification of errors when text processing is performed by the sequential stages of a traditional IE system. Finally, we allow extremely general and rare-case extraction rules to be included at any level of the extraction process by augmenting all extraction rules with suites of classifiers capable of shoring up the low precision typically afforded by such rules. We also note that the common representational structure of the ontology permits the integration of the IE task with a wide range of text mining and text processing applications, including link extraction, text summarization, and multilingual extraction. While these tasks are outside the scope of the Phase II effort, it is clear that the development of a robust extraction capability that facilitates such applications will significantly enhance the commercial potential of the OBIE system.

3.1 Description of System

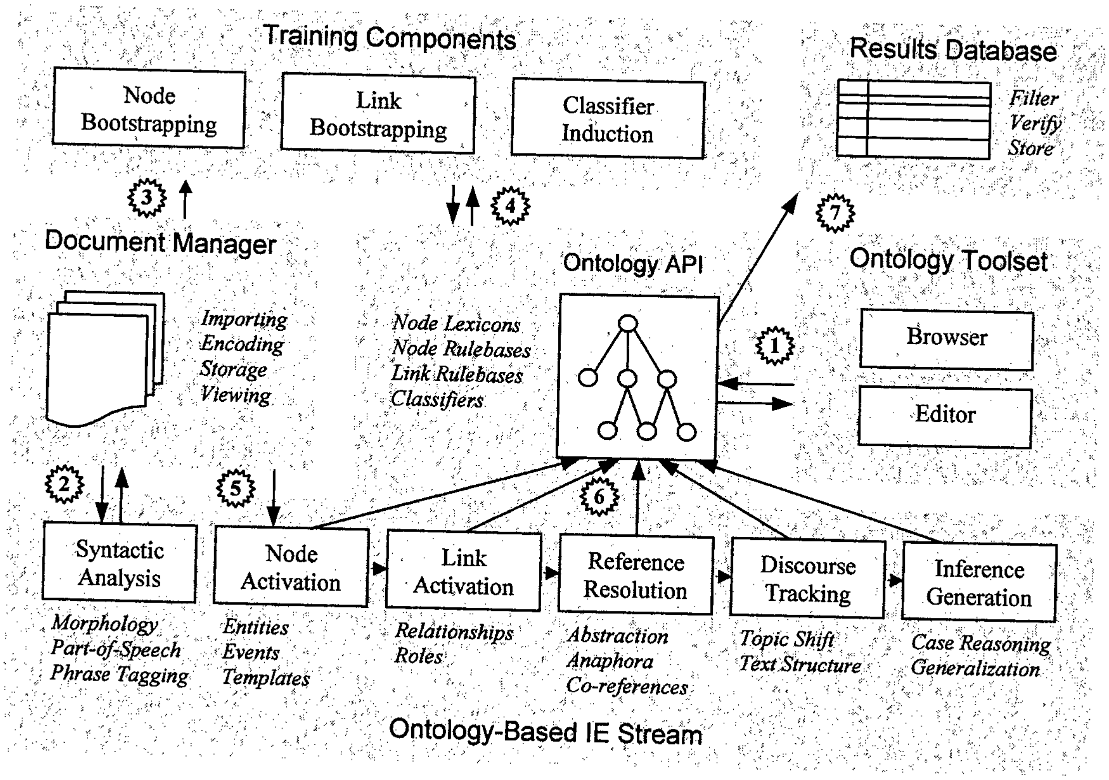


Figure 12. Phase II System Overview

Figure 12 gives an overview of the components and processing flow of our proposed Phase II OBIE system, building on the lessons we learned during the execution of Phase I. General processing flow involves roughly seven steps (refer to the numbered stars in the diagram):

1. **Definition of the domain ontology.** A basic ontology toolset will be constructed to support user definition of a domain ontology. This toolset may include viewers, node and link editors, and other elements necessary to visualize and manipulate the ontology. Functionality may be implemented to import ontological definitions from other sources (e.g., Ontolingua) as needed.

2. **Loading and pre-processing of training documents.** Documents are typically imported in some machine-readable format from a corpus source. Text may be encoded for efficiency using word and phrase hash tables. Lexical processing is performed, including morphological analysis and part-of-speech tagging. Text will be further segmented into phrasal constituents to support the bootstrapping step. Processed documents are stored internally and are managed by OBIE to provide efficient access by all components of the IE stream.

3. **Generation of pattern sets to support bootstrapping.** An exhaustive set of heuristic patterns is generated from a training corpus as a precursor to the entity, event, and relationship bootstrapping processes. These patterns form the engine from which extraction rulebases and lexicons are populated for the new domain.

4. **Bootstrapping of the ontology and classifiers.** Bootstrapping algorithms are run in conjunction with user training to learn the mappings between natural language and the entities, events, and relationship linkages that compose the ontology. During this training, node lexicons and extraction rulebases are learned, as are the rulebases used to recognize relationships between nodes from text. After the user concludes an iteration of the training process, precision-enhancing classifiers are induced from training data captured during the bootstrapping process.

5. **Loading and pre-processing of source documents for extraction.** At this point, the trained ontology is ready to be used by the IE stream to perform text extraction. The same pre-processing mechanisms used on the training document set are applied to new source documents.

6. **Semantic interpretation by components of the IE stream.** Input documents are fed to the IE processing stream. Entity and event nodes in the ontology are instantiated via extraction rules by the Node Activation module. These rules may be quite broad in coverage; classifiers are used by the activation modules to restore the precision lost by favoring high-recall rules. Relationships between activated nodes are recognized and instantiated by the Link Activation module. The ontology itself serves as a blackboard to organize the semantic content of the input text. Because abstraction, composition, and other relationships are encoded in the ontology, anaphora and co-reference disambiguation can occur dynamically by the Reference Resolution module by unifying different references to the same conceptual nodes or links. A Discourse Tracking module determines when topic shifts occur so that template instantiation can be terminated at appropriate times. Also, an Inference Generation module can help make explicit those

connections that are left implicit in the text, facilitating more robust instantiations of scenario template nodes.

It is important to note that all of the modules can work in unison, each contributing to the emerging semantic interpretation of input text via the shared ontology. Data is not passed between modules as in a traditional IE stream, but rather each module is free to work with the unfolding parse of the input text within the ontology. Certainty factors may be attached to node and link instantiations depending on the consensus of views by all modules, if desired.

7. **Harvest instantiated templates from the ontology and update knowledge bases.** Once a document has been processed, fields of interest from instantiated templates in the ontology can be exported from the ontology and used to populate knowledge bases. Additionally, user verification or filtering of highly uncertain extraction data may occur.

The remainder of this section will briefly examine each of the components represented in Figure 12.

3.1.1 Document Management

As part of the infrastructure necessary to support an end-to-end extraction system, a document management subsystem will coordinate the input and preprocessing of corpus text. This task includes recognition of source data formats. Word strings will likely be hashed into integer identifiers to facilitate efficient comparison and pattern matching operations. The manager will guide the application of morphological analysis, part-of-speech annotation, and syntactic phrase segmentation by the Syntactic Analysis module. The preprocessed and annotated documents will be stored internally for access by the training and extraction systems. Document viewers and other tools will be developed as needed to support the end-to-end extraction process.

3.1.2 The Ontology

One of the first tasks for the Phase II effort will be to decide on the representative capacity of the ontology. The Phase I ontology represented entity and event nodes as well as abstraction and compositional (part-of) links between nodes. Role relationships were represented implicitly as part of the compositional hierarchy (a slight abuse of semantics). To formalize the expressiveness of the ontology, additional link types will likely be incorporated, including explicit role links and part-whole relationships. We will also consider current trends in ontological research to ensure that the ontology is sufficiently well specified to facilitate sharing across domains. For example, work by Guarino and Welty [2000] seeks to identify meta-features of the ontology that, if properly adhered to, facilitate ontological transfer between applications and domains.

In addition to their structural context, nodes of the ontology are further semantically defined by the lexicons and rulebases that map into them. Phase II will augment these with link rulebases to capture the patterns of language use that correlate with different link types. Data structures to support classification will also be attached to nodes to ensure that scalable and incremental classification can occur. (In Phase I, all classifier data was maintained in a global lookup table, a less-scalable solution.) By distributing all semantically-laden information within the nodes and links of the ontology and by allowing natural language to activate that information as it is needed, OBIE will avoid expensive ontology search algorithms, ensuring a scalable architecture.

Access and elaboration of the ontology will be restricted to a well-defined API to ensure that the ontology is structured consistently and predictably by all modules of the training component and the IE stream. A toolset sufficient to enable a user to view and modify the ontology will be developed as part of the system infrastructure.

3.1.3 Training Components

Our Phase I research extended entity-bootstrapping algorithms to integrate them with the ontology and to perform both entity and event bootstrapping. In Phase II, these algorithms will be generalized to support generic bootstrapping of nodes in the ontology via the Node Bootstrapping component. We also developed a technique to recognize the role components of events. This technique will be generalized and other techniques developed to perform link bootstrapping, in which language patterns correlated with different link types are discovered and added to link rulebases.

Another Phase I discovery was that information recorded as part of the bootstrapping process was sufficient to induce multiple suites of classifiers capable of enhancing the precision of node rulebases. The Classifier Induction component will organize these classifiers. We will also develop analogous classifiers to assist in the link recognition process. Every rule-based mechanism resident within OBIE will be augmented by precision-enhancing classifiers to permit rules of maximum breadth (and thus recall) to reside in the system.

3.1.4 IE Stream Components

The largest part of the Phase II development cycle will involve the implementation of the actual modules that compose the ontology-based IE stream. These modules will not form a processing pipeline as in traditional IE systems, wherein the results of one module are handed off to the next for further processing. Rather, all modules will interact with the domain ontology, instantiating increasingly more complex and detailed interpretations of input text through node and link activation, template correlation, ambiguous reference resolution, and semantic inference.

The Syntactic Analysis module will be driven by the document manager and will perform morphological analysis, part-of-speech tagging, and phrasal segmentation. This pre-processing will support the bootstrapping process and will form the pool from which OBIE's extraction rules are drawn.

The Node Activation module will be responsible for recognizing and activating references to ontological nodes in natural language text, including entity, event, and more complex template nodes. Each instantiation of a node represents an extraction of that component from text. We will build on the semantic tagger and event extractor applications developed in Phase I in designing and implementing this module.

The Link Activation module is responsible for recognizing and activating references to relationships between nodes within the ontology. Relationships (such as the roles played by entities within events) mediate the interpretation of entities and events as they occur in text. This process is similar to traditional link discovery and extraction, except that the links are typically part of a more organized ontology of relationships. Extraction thus occurs at a deeper semantic level than just surface (syntactic) features.

The abstraction and compositional hierarchies (plus other link types chosen as part of the ontological formalization) permit a rich set of referent disambiguations to occur, including anaphor and co-reference resolution. One automated technique will leverage patterns of activation over the ontology to correlate references to more abstract or general entities during discourse. This process is efficiently implemented via marker passing (see Section 2.2.3). Other techniques may include directed ontological search (e.g., for pronoun dereferencing) that can be performed on the fly.

As templates are being constructed within the ontology, some mechanism is required to recognize topic shifts such that template construction can be concluded. In Phase I, we employed a simple heuristic that assumed that references to a template event were always constrained to contiguous sentences within a text. The advent of a sentence with no new template information was cause to terminate template construction. While the heuristic works fairly well in many cases, clearly more sophisticated discourse analysis should be performed. This is the function of the Discourse Tracking module.

Finally, a mechanism to perform semantic inference will be implemented by the Inference Generation module. Such inferences explicitly fill in information or activate relationships that are left implicit in text. A strength of OBIE's ontology-based approach is its ability to use the structure of the ontology to make such inferences in an automated manner. This can be accomplished by several methods. As with reference resolution, marker-passing techniques can efficiently connect related nodes in a semantic network. Additionally, the ontology may be viewed as a large case-library of extracted events, entities, and relationships. As such, generalizations may be drawn from previous extractions for use in understanding new text.

3.1.5 Results Database

Since extraction results are stored as instantiated nodes and links in the ontology, some mechanism must exist to pull this information from the ontology and package it as database templates for export to a knowledge base. This basic function is performed by the Results Database component. This module may also perform some filtering and user verification of extraction results. This process can be facilitated by attaching certainty factors to extractions based on the cohesiveness of various system viewpoints. (Heavy inferencing may, for example, diminish the certainty of the extraction.) Elaboration of this capability will depend on the degree to which it is useful to the system development effort.

3.2 Phase II Technical Objectives

Phase II research and development will build on the significant progress made in Phase I and result in a complete implementation of OBIE, an ontology-based, end-to-end, high-recall information extraction system. The primary goals of the Phase II research are to:

1. **Elaborate the key algorithms of the OBIE system:**

   - We will **formalize the specification of the ontology** used by OBIE. In addition to abstraction and compositional links, role and attribute links may be useful to more precisely define the semantics of a node. We will also investigate the capacity to represent domain-specific abstract relationships of interest to a user.

   - We will **generalize the bootstrapping procedures** to operate on all formalized relationship (link) types, in addition to the existing entity, event, and role capabilities.

Essentially any relationship that has a correlated pattern of language use should be amenable to discovery by bootstrapping.

- We will **extend the capabilities of OBIE's precision-enhancing classifiers**. This will include refinements to the existing algorithms, as well as the development of additional classifier types. Committee-based approaches, in which weighted classifiers become voting members of a committee, typically outperform individual members and will be investigated.
- We will **augment the existing training interface** to incorporate training tasks other than simple classification and categorization, while still assuming a user without system or domain expertise. Such tasks might include reverse extraction, in which rare-case extraction rules are generalized over other ontological nodes by presenting to the user original training contexts for the rule with alternative lexical items substituted from other candidate nodes. Such a capability would allow rules generated from sparse data to be reliably generalized, thereby increasing their coverage.

2. **Implement each component of the end-to-end information extraction stream:**
   - A comprehensive review will be made of existing approaches to each element of the IE stream. Favoring the best current techniques, we will **characterize how each approach might benefit from the ontology**. For example, we know that statistical techniques can probably utilize the abstraction hierarchy to incorporate shrinkage-like techniques (see Section 2.2.4). We will implement the most promising technique for each module, ensuring that it is fully integrated with the ontology.
   - The **semantic tagging and event extraction applications will be converted into true IE stream modules** to perform event, entity, and relationship extraction.
   - **Additional necessary ontology-based IE modules will be developed**, including a Syntactic Analysis module (to perform morphology, part-of-speech, and phrasal analysis), a Reference Resolution module, a Discourse Tracking module, and an Inference Generation module.
3. **Evaluate OBIE's accuracy metrics and scalability** during a larger-scale demonstration in a DARPA-chosen task domain. At a minimum, OBIE might be tested in at least one of the Message Understanding Conference task domains, permitting a baseline comparison to previous systems.

4 Commercialization Plans

Over the past several years, global competition and other complexities have increased the reliance of U.S. military and civilian institutions on computers and the large amount of information to which they provide access. This trend offers a unique opportunity for the marketing of tools that can increase productivity by augmenting the ability of institutions to process electronic documents. We feel that the combination of features that we plan for this project have direct commercial applicability to commercial health care and insurance corporations, as well as to electronic commerce. In each of these domains, the ability to generate accurate databases automatically from the large quantities of free-text documents they process would be a great productivity boon. While other information extraction tools exist, their ability to extract content at high recall and precision is very limited, as is their ability to adapt to new domains.

There are two primary types of products for commercialization. First, we can market OBIE as a stand-alone system and allow customers to develop their own information extraction applications. Since off-the-shelf modules may integrate with OBIE's ontology-based architecture via its API to form a customized and portable end-to-end IE system for use directly by end users, this tool should prove extremely attractive to a wide variety of potential clients. Our expectation is that we will be able to attract substantial outside commercial investment from one of the major knowledge management software development companies or venture capitalists within 12 months after the beginning of Phase II. Further, we expect that OBIE will start generating commercial revenues (in the realm of $200,000 to $300,000) within six months after the end of Phase II, most likely through licensing agreements.

We are also particularly interested in developing "competitive intelligence" solutions that would provide our clients with, for example, a dramatically improved ability to plan product development and marketing strategies and monitor their competitors. This application area is very attractive because of the increasingly aggressive strategies required to succeed in an era where information is disseminated widely and rapidly via the World Wide Web. Marketing such specialized situation assessment system development services is similar to SHAI's core business of marketing AI research and development services, at which we are very successful. SHAI's current annual revenue from such services is approximately $4 million. Within six months after the end of Phase II, we expect revenues from this commercialization direction could amount to over $200,000 annually.

5 References

[Appelt and Israel, 1999] D. E. Appelt and D. J. Israel. "Introduction to Information Extraction Technology." IJCAI-99 Tutorial. SRI International.

[Embley, et. al., 1998] D. W. Embley, D. M. Campbell, S. W. Liddle, and R. D. Smith. "Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents." CIKM'98 Proceedings.

[Freitag and McCallum, 1999] D. Freitag and A. K. McCallum. "Information Extraction with HMMs and Shrinkage." To appear in Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction.

[Glickman and Jones, 1999] O. Glickman and R. Jones. "Examining Machine Learning for Adaptable End-to-End Information Extraction Systems." AAAI 1999 Workshop on Machine Learning for Information Extraction, to appear.

[Grishman, 1997] R. Grishman. "Information Extraction: Techniques and Challenges." M. T. Pazienza (ed.), Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology (International Summer School SCIE-97), pp. 10-27, Springer-Verlag.

[Guarino, 1997] N. Guarino. "Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration." M. T. Pazienza (ed.), Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, pp. 139-170, Springer-Verlag.

[Guarino and Welty, 2000] N. Guarino and C. Welty. "Conceptual Modeling and Ontological Analysis." AAAI-2000 Tutorial.

[Humphreys, et.al, 1997] K. Humphreys, R. Gaizauskas, and S. Azzam. "Event Coreference for Information Extraction." Proceedings of the ACL/EACL '97 Workshop: Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts.

[Jones, et. al., 1999] R. Jones, A. McCallum, K. Nigam, and E. Riloff. "Bootstrapping for Text Learning Tasks." IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, to appear.

[Kehler, et. al., 1998] A. Kehler, J. R. Hobbs, D. Appelt, J. Bear, M. Caywood, D. Israel, M. Kameyama, D. Martin, and C. Monteleoni. "Information Extraction Research and Applications: Current Progress and Future Directions." SRI International, http://citeseer.nj.nec.com/did/78141.

[Kim and Moldovan, 1993] J. Kim and D. Moldovan. "Acquisition of Semantic Patterns for Information Extraction from Corpora." Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications.

[Martin, 1989] C. Martin. "Case-based Parsing." C. Riesbeck and R. Schank, Inside Case-Based Reasoning, pp. 319-372, Lawrence Erlbaum Associates.

[Miller, 1995] G. Miller. "WordNet: a lexical database for English." Communications of the ACM, 38(11), 1995.

[Mitchell, 1997] T. Mitchell. Machine Learning. McGraw Hill.

[Muslea, 1999] I. Muslea. "Extraction Patterns for Information Extraction Tasks: A Survey." AAAI-99 Workshop on Machine Learning for Information Extraction.

[Norvig, 1989] P. Norvig. "Marker Passing as a Weak Method for Text Inferencing." *Cognitive Science*, Vol. 13: 569-620.

[Noy and Hafner, 1997] N. F. Noy and C. D. Hafner. "The State of the Art in Ontology Design: A Survey and Comparative Review." AI Magazine, Fall 1997, pp. 53-74, AAAI.

[Riloff, 1993] E. Riloff. "Automatically Constructing a Dictionary for Information Extraction Tasks." Proceedings of the Eleventh National Conference on Artificial Intelligence. AAAI Press.

[Riloff and Jones, 1999] E. Riloff and R. Jones. "Learning Dictionaries for Information Extraction Using Multi-level Boot-strapping." AAAI 1999, to appear.

[Roth, 1998] D. Roth. "Learning to Resolve Natural Language Ambiguities: A Unified Approach." Proceedings of AAAI-98.

[SAIC, 1998] Science Applications International Corporation. Message Understanding Conferences Web Site, http://www.muc.saic.com, Science Applications International Corporation.

[Schank, 1982] R. Schank. Dynamic memory: A theory of reminding and learning in computers and people. Cambridge University Press.

[Sleator and Temperley, 1993] D. Sleator and D. Temperley. "Parsing English with a Link Grammar." Third International Workshop on Parsing Technologies, August 1993.

[Soderland, 1999] S. Soderland. "Learning Information Extraction Rules for Semi-Structured and Free Text." Machine Learning, pp. 1-44, Kluwer Academic Publishers.

[Thompson, et. al., 1999] C. A. Thompson, M. E. Califf, and R. J. Mooney. "Active Learning for Natural Language Parsing and Information Extraction." Proceedings of the ICML-99.

[Wilks and Catizone, 1999] Y. Wilks and R. Catizone. "Can We Make Information Extraction More Adaptive?" In M. Pazienza (ed.) Proceedings of the SCIE99 Workshop. Springer-Verlag, Berlin. Rome.

[Woods, 2000] W. Woods. "Conceptual Indexing: Practical Large-Scale AI for Efficient Information Extraction." Proceedings of AAAI-2000, AAAI Press and the MIT Press.

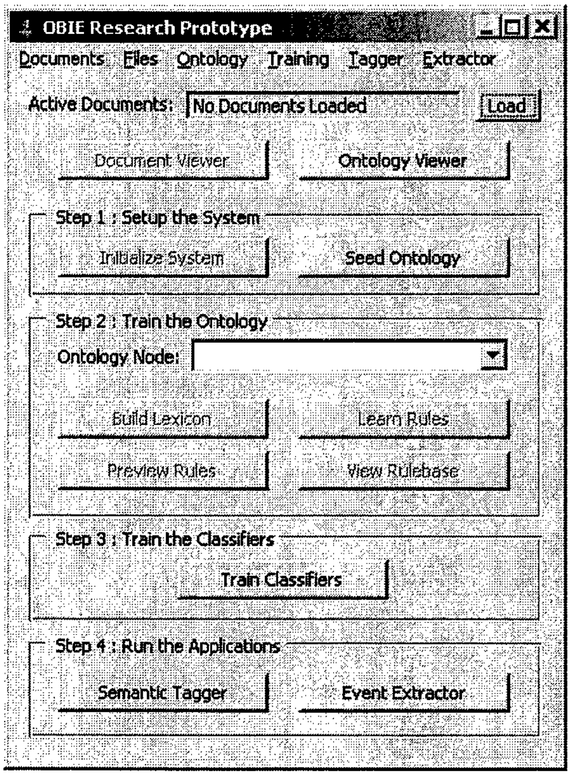
[Yangarber and Grishman, 1997] R. Yangarber and R. Grishman. "Customization of Information Extraction Systems." Proceedings of International Workshop on Lexically Driven Information Extraction, Frascati, Italy, July 16, 1997.

Appendix A – Demo Sequence

A twenty-five minute demo was prepared for the final presentation (see Section 2.1.9). The following demo script assumes that Allegro CL 5.0.1 has been installed on the system upon which the demo is to execute. The prototype runs within the Allegro CL environment – there is no executable. (Make sure that the Allegro build is current via sys:update-allegro. The CD install of 5.0.1 doesn't seem to handle packages properly.)

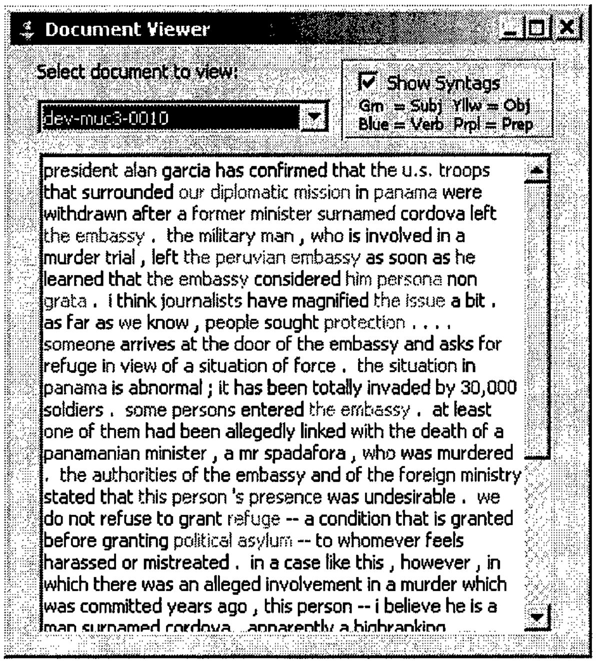
Step 1: Setup the System

1. **Load OBIE Project** – Start Allegro LISP 5.0.1 under the IDE environment. Open the **obie.lpr** project from the OBIE project directory. Close all Allegro CL IDE windows except the debug window.
2. **Run OBIE** – Choose Run Projects option. The prototype demo interface should appear, as pictured below. Click the **Load button** next to the Active Documents edit box (or choose Documents / Open Document File) and load the **muc-training-set-demo.obdoc** file. After that finishes, click the **Initialize System** button. This may take a couple of minutes to execute. The Debug Window will say "Done initializing system" when finished.



Active Documents: No Documents Loaded

3. **Note Initial Conditions** – No lexical knowledge exists in the system as yet. The domain is that of Latin American terrorism. Browse the training documents with annotated syntactic tags by clicking the **Document Viewer** button (or choose Documents / View Documents). Select to view **dev-muc3-0010**. Text with color-coded syntactic tags should appear.



Select document to view: dev-muc3-0010

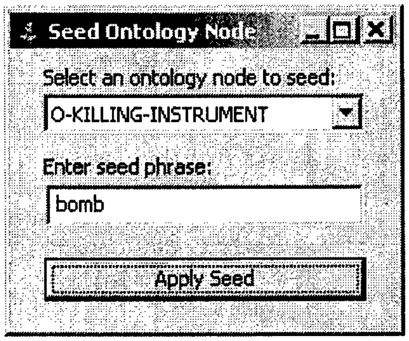
☑ Show Syntags
Grn = Subj Yllw = Obj
Blue = Verb Prpl = Prep

president alan garcia has confirmed that the u.s. troops that surrounded our diplomatic mission in panama were withdrawn after a former minister surnamed cordova left the embassy . the military man , who is involved in a murder trial , left the peruvian embassy as soon as he learned that the embassy considered him persona non grata . i think journalists have magnified the issue a bit . as far as we know , people sought protection . . . . someone arrives at the door of the embassy and asks for refuge in view of a situation of force . the situation in panama is abnormal ; it has been totally invaded by 30,000 soldiers . some persons entered the embassy . at least one of them had been allegedly linked with the death of a panamanian minister , a mr spadafora , who was murdered . the authorities of the embassy and of the foreign ministry stated that this person 's presence was undesirable . we do not refuse to grant refuge -- a condition that is granted before granting political asylum -- to whomever feels harassed or mistreated . in a case like this , however , in which there was an alleged involvement in a murder which was committed years ago , this person -- i believe he is a man surnamed cordova , apparently a highranking

Observe that the parser makes numerous errors, including a bad verb phrase ("cordova left") and a missed sentence "In a case like this…." While these errors diminish accuracy (especially recall), the techniques used by OBIE are robust and the parsing is sufficient for research purposes.

You may also wish to browse the structure of the ontology via the **Ontology Viewer** button. A graphical representation of the ontology will appear. Note the top-level division of nodes into functional knowledge (e.g., generation and activation functions), entities, events, and irrelevant (i.e., scaffolding) nodes. You may examine node contents by left-clicking on a node to turn it red and then right-clicking for a menu of options. Since all lexicons are currently empty, clustering will have no effect and node lexicons will have no entries. You may wish to view the packaging hierarchy of :o-bombing. When you have finished examining the ontology, close the ontology viewer window.

4. **Seed Nodes** – Under the premise that we are interested in extracting out examples of bomb explosions, seed the :o-killing-instrument node with the phrase "bomb" via the **Seed Ontology** button. When the seed dialog appears as at left, select :o-killing-instrument as the node to be seeded and type "bomb" into the text edit. Click the **Apply Seed** button. You will then be prompted to categorize instances of the seed term in text as being a member of the class of interest or not. Specialize the training examples to the :o-bomb node via the combo box as shown to the right. Since all bomb examples in the text correspond to actual bombs, accept all examples by checking the "Apply to all contexts" box and then click the Accept button. Next, seed the :o-killing-instrument node with the phrase "grenade". When classifying, specialize the node to :o-grenade and accept the example. Finally, seed the :o-killing-instrument node with the phrase "mine". When classifying, specialize the mine node to :o-mine. Since there are two semantic classes of mine represented in the training examples (ore mines and explosive mines), carefully accept instances of bombs in the text but reject instances of ore mines (via the Reject button). Close the Seed Ontology dialog if it is still open.



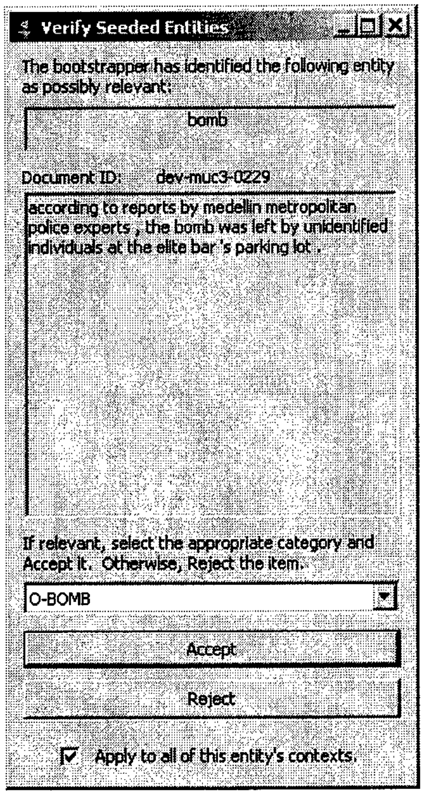
Seed Ontology Node

Select an ontology node to seed:

O-KILLING-INSTRUMENT

Enter seed phrase:

bomb



Verify Seeded Entities

The bootstrapper has identified the following entity as possibly relevant:

bomb

Document ID: dev-muc3-0229

according to reports by medellin metropolitan police experts , the bomb was left by unidentified individuals at the elite bar 's parking lot .

If relevant, select the appropriate category and Accept it. Otherwise, Reject the item.

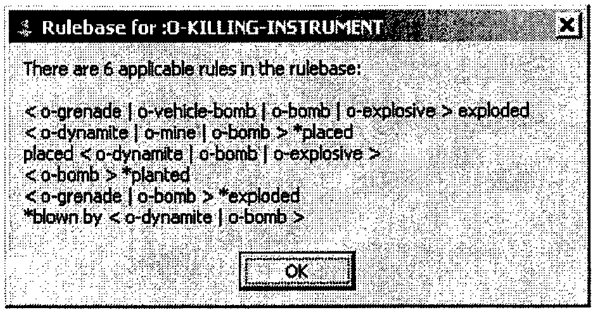
O-BOMB

☑ Apply to all of this entity's contexts.

Step 2: Train the Ontology

5. **Train Killing Instruments** – Select :o-killing-instrument in the Ontology Node combo of the main interface and click on the **Build Lexicon** button. OBIE will display a series of lexically similar concepts. You will accept all positive examples and reject all negative examples. The system will make a best guess as to the appropriate level of the ontology to which the node should be specialized. The only time you will have to change the category for the node is with car bombs (e.g., "other car bombs"): change the category to :o-

vehicle-bomb. The only entities you will have to reject are those of the irrelevant semantic sense of mines (e.g., "peruvian mines"). If you run across a long sequence of contexts for a phrase (e.g., for the phrase "bombs"), you may wish to click the "Apply to all contexts" check-box to accept them all at once (assuming they are all likely to be relevant). Next, OBIE bootstraps new entities that may or may not be weapons and presents them for classification. Reject all contexts for "kg", specialize "device" to :o-explosive and accept, specialize "dynamite charge" to :o-dynamite and accept all, accept relevant examples of "charge" and reject irrelevant examples (all but "...charge of dynamite..." are irrelevant), reject all of "people", "late-model white monza", "savings", and "checkpoints", accept all of "explosive devices", and then reject all the remaining items. Hit the **Preview Rules** button to see what the currently best-ranked rules are. Now click on the **Learn Rules** button. You will be asked to classify more training examples. Accept both of the "explosive charge" instances. Click on **View Rulebase** to see the actual rules learned; there should be six rules shown as at left. Notice that they have been specialized to all of the ontology nodes to which they are applicable based on the training data.

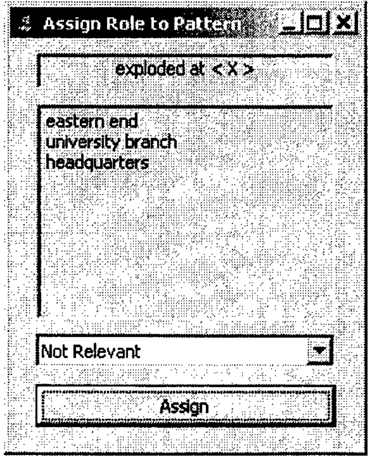


Rulebase for :O-KILLING-INSTRUMENT

There are 6 applicable rules in the rulebase:

```
< o-grenade | o-vehicle-bomb | o-bomb | o-explosive > exploded
< o-dynamite | o-mine | o-bomb > *placed
placed < o-dynamite | o-bomb | o-explosive >
< o-bomb > *planted
< o-grenade | o-bomb > *exploded
*blown by < o-dynamite | o-bomb >
```

6. **Train the Bomb Explosion Event** – The :o-bomb-explosion event has two roles, one corresponding to the geographic location in which the event occurred and one corresponding to the weapon used. Since we have trained up the node packaged by the weapon role (:o-killing-instrument), OBIE can use that work to seed the explosion event training process. We will thus bootstrap up bomb explosion events *without seeding*. **Select :o-bomb-explosion** in the Ontology Node combo box. Hit **Preview Rules** to see what patterns are currently best ranked as indicators of the :o-bomb-explosion event based on the weapons training that we have done thus far. Train :o-bomb-explosion by hitting the **Build Lexicon** button. Since we want to accept only instances where an explosion occurred, reject all examples of "must be place", "have been place", "can be placed", "were placed", and "was placed". Accept all examples of "had exploded", "exploded", and "has exploded". Hit **Learn Rules** to learn the event rulebase. A role assignment dialog will appear as shown to the left. This dialog will present a list of entities associated with a given pattern. Two roles will be available in the combo box for this event, one for weapon and one for geographic location (defined for our purposes as specific cities, countries, and neighborhoods). For each role assignment dialog, scan the list of entities. If any entity is a relevant location or weapon, select the appropriate role in the combo and hit assign. Otherwise, select "Not Relevant" and hit assign. Because the entities in the dialog to the left are neither weapons nor specific geographic locations, choose the "Not Relevant" option and hit the assign button. You will also assign as "Not Relevant" the patterns *exploded before <x>* and *exploded <x>*. When the dialog for *exploded in <x>* appears, notice that the list of entities does include specific geographic names (medellin, bogota, etc.) as shown to the

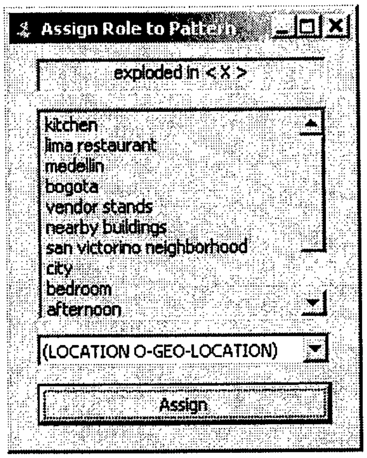


Assign Role to Pattern

exploded at < X >

eastern end
university branch
headquarters

Not Relevant



exploded in <x>

kitchen
lima restaurant
medellin
bogota
vendor stands
nearby buildings
san victorino neighborhood
city
bedroom
afternoon

(LOCATION O-GEO-LOCATION)

left. Choose the Location role and click assign. You will then need to classify each of the entities as being valid locations or not. Reject "kitchen" and "lima restaurant". Accept "medellin" and "bogota". Reject "vendor stands" and "new buildings". Accept "san victorino neighborhood" and "city". Reject all of the rest. Assign as "Not Relevant" the patterns *exploded under* <x> and *exploded of* <x>. Assign to the Location role the pattern *exploded near* <x>. Accept the one entity. Assign as "Not Relevant" the pattern *exploded inside* <x> and *exploded on* <x>. Assign to the Weapon role the pattern <x> *exploded.* Reject the two entities that are given for classification. (The remaining items are already in the weapons lexicon and do not need to be reclassified.) With training complete, click on the **View Rulebase** button and verify the presence of three extraction rules for this event.

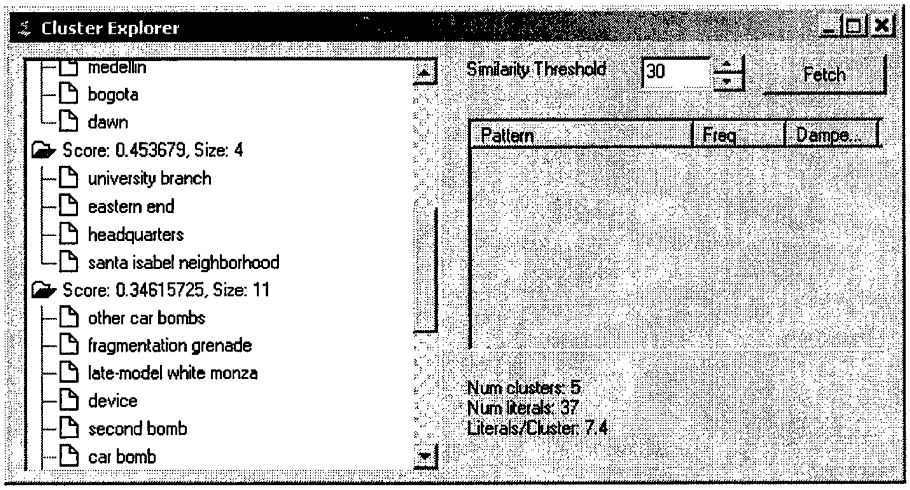
7. **Train the Geo-Location Entity –** Just as entity training can automatically seed the event bootstrapping process, the work we just performed in training the bomb explosion event can automatically seed the bootstrapping of geographic locations. We will now bootstrap that entity *without requiring user seeding.* Select :o-geo-location in the Ontology Node combo box. Hit the **Build Lexicon** button. Accept all of the suggested lexically similar items. When the contextually similar entities appear, Reject all of "motion" and "agriculture livestock ministry". Accept all of "el congo" and "cali". Reject all of "alvarez", "office", "no arrests", "attention", "high-ranking government officials", "judges", "political leaders", "citizens", and "public officials". Accept all of "el salvador". Because the list of entities is pretty long, we will now discontinue this process by clicking on the X button (close window) of the classification dialog. Click the **Learn Rules** button. Accept all of "rochela". Reject all of "regiment". Accept all of "12th block". View the rulebase via the **View Rulebase** button and verify the presence of six rules. Note that some of them don't have much to do with bombings, but they do reliably predict locations in general.

8. **Train the People Harmed Event –** To demonstrate event seeding, we will now learn to recognize indicators for people getting harmed by providing one seed term. Click on the **Seed Ontology** button. Choose :o-people-harmed in the node combo and type the seed term "died". Accept all of the classification entities. Close the seed dialog. Select :o-people-harmed in the Ontology Node combo of the main interface. Click on the **Build Lexicon** button. Accept all of the lexically similar "had died", "have died", and "has died" verb phrases. Accept all of the contextually similar "were wounded" and "was wounded" phrases. Reject the "may have been wounded" phrase since this doesn't definitively indicate a harming has occurred. Accept all of the "were wounded" and "was wounded" phrases. (These are actually different phrases than the previous versions though they look identical.) Now click the **Learn Rules** button. Note that this event has only one role, that of victim, associated with it. Assign as "Not Relevant" the patterns *died on* <x>, *died as* <x>, *died from* <x>, *died at* <x>, *died since* <x>, and *died during* <x>. Assign to the Victim role *died* <x> since the entities listed are indeed people. Accept both entities. Assign as "Not Relevant" the pattern *died in* <x>. Assign to Victim <x> *died.* To expedite the demo,

accept all entities as relevant (one or two irrelevant ones may slip in, but that is ok as the classifiers are robust). Assign as "Not Relevant" * *wounded by* <*x*>, * *wounded in* <*x*>, **wounded as* <*x*>, and * *wounded to* <*x*>. Assign as Victim <*x*> * *wounded.* Again in the interests of time, accept all entities. Finally, assign as "Not Relevant" * *wounded during* <*x*>. Press the **View Rulebase** button and verify that three rules exist.

9. **Learn the People Entity** – In the interests of time, we will skip bootstrapping of the :o-people node and directly learn from the training instances captured during event bootstrapping in the last step. Select :o-people in the Ontology Node combo and click the **Learn Rules** button. Accept the given entity. Click the **View Rulebase** button and verify the presence of four rules.

10. **Observe Bomb Explosion Clusters** – To suggest how event structure might be induced semi-automatically from text, we'll run a clustering algorithm on the :o-bomb-explosion event. Click on the **Ontology Viewer** button of the main interface. Left click on the :o-bomb-explosion node in the atomic event hierarchy, turning it red. Right click on the same node and choose the **Cluster Literals** option. Browse the clusters and notice that weapons and locations tend to separate fairly well. Structural targets and locations also segregate somewhat. (These clusters become better with more training of the ontology.) This suggests that the structure of an event may be inducible directly from text, perhaps under user-guidance. OBIE could thus assist in the ontology definition process. Close the Cluster Explorer and Ontology Viewer windows.



medellin
bogota
dawn
Score: 0.453679, Size: 4
university branch
eastern end
headquarters
santa isabel neighborhood
Score: 0.34615725, Size: 11
other car bombs
fragmentation grenade
late-model white monza
device
second bomb
car bomb

Similarity Threshold 30

Num clusters: 5
Num literals: 37
Literals/Cluster: 7.4

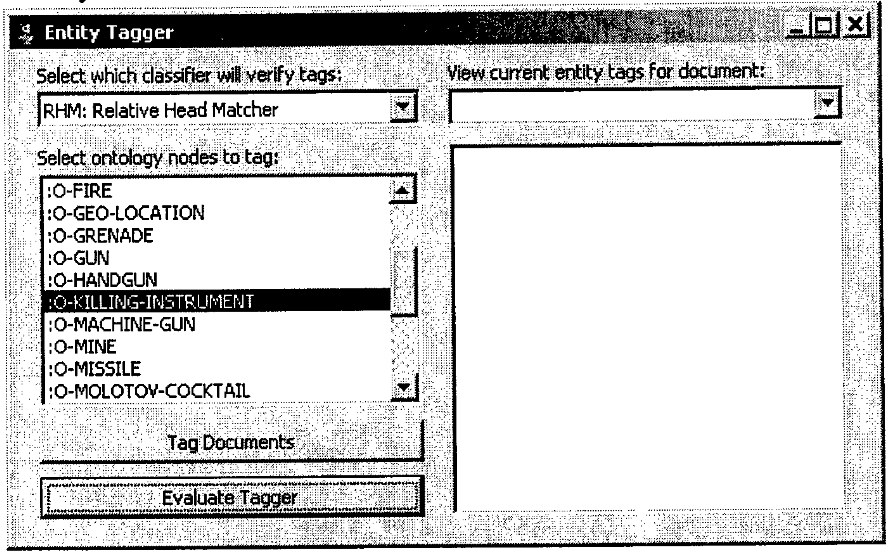
Step 3: Train the Classifiers

11. **Train the Classifiers** – Click the **Train Classifiers** button on the main interface. This will allow the naïve-bayes and k-nearest-neighbor classifier suites to initialize themselves based on the training data captured during the bootstrapping process.

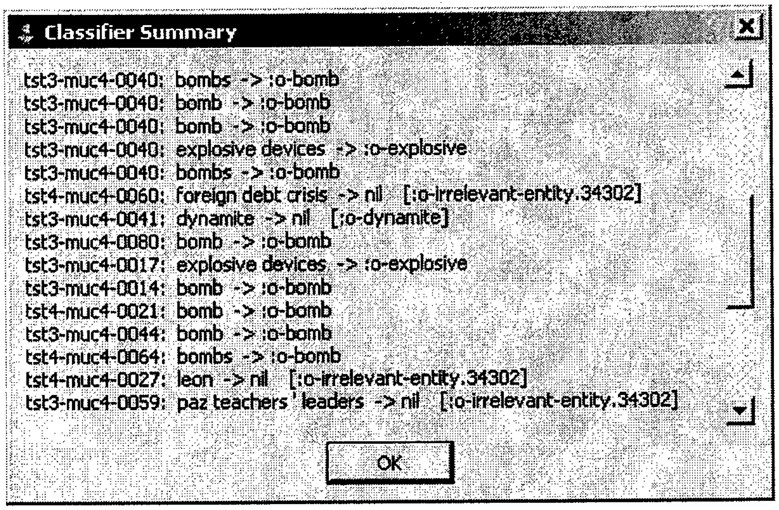
Step 4: Run the Applications

12. **Load the Test Documents** – Click the **Load Documents** button on the main interface and select **muc-testing-set.obdoc**. This will load 200 test documents.

13. **Run the Semantic Tagger** – Click on the **Semantic Tagger** button. The semantic tagger interface will appear. This interface is used to tag entities from the ontology in the most recently loaded document set (the test documents in this case). Note the multiple classifier types available to be used during semantic tagging. Choose **RHM: Relative Head Matcher** in the classifier combo box. This classifier uses direct dictionary lookup to try and classify each entity extracted by the node rulebases used by the tagger. Select the :o-killing-instrument node in the entity list box. Click the **Tag Documents** button. Then click the **Evaluate Tagger** button. When the file load dialog appears, choose the **demo-killing-instrument-key.obsem** file. (This contains an answer key for the classifiers to use in determining their accuracy characteristics.) OBIE will now show a list of all entities extracted from the test documents and the classification of each entity by the classifier, as well as various accuracy metrics based on the answer key. Notice that "foreign debt crisis" and "dynamite" could not be classified by dictionary lookup because "crisis" and "dynamite" (the head nouns of each phrase) never appeared in the training texts. The node name in square brackets after misclassified entities is the correct answer from the key. Close the Classifier Summary window and now select the **RHM + LPS + NB + AI Committee** classifier. Hit the **Tag Documents** and **Evaluate Entities** buttons again. Note this time that the classifiers correctly identify "foreign debt crisis" as irrelevant (i.e., not a killing instrument) and "dynamite" as dynamite based on contextual

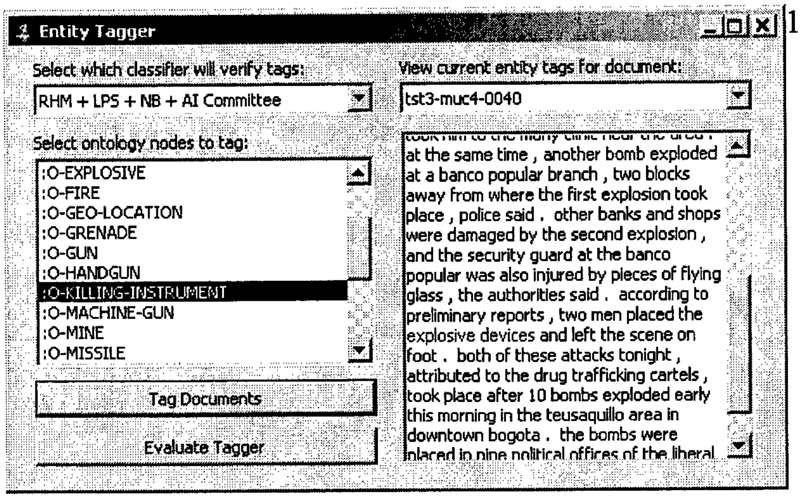


Select which classifier will verify tags: RHM: Relative Head Matcher



```
tst3-muc4-0040: bombs -> :o-bomb
tst3-muc4-0040: bomb -> :o-bomb
tst3-muc4-0040: bomb -> :o-bomb
tst3-muc4-0040: explosive devices -> :o-explosive
tst3-muc4-0040: bombs -> :o-bomb
tst4-muc4-0060: foreign debt crisis -> nil  [:o-irrelevant-entity.34302]
tst3-muc4-0041: dynamite -> nil  [;o-dynamite]
tst3-muc4-0080: bomb -> :o-bomb
tst3-muc4-0017: explosive devices -> :o-explosive
tst3-muc4-0014: bomb -> :o-bomb
tst4-muc4-0021: bomb -> :o-bomb
tst3-muc4-0044: bomb -> :o-bomb
tst4-muc4-0064: bombs -> :o-bomb
tst4-muc4-0027: leon -> nil  [:o-irrelevant-entity.34302]
tst3-muc4-0059: paz teachers' leaders -> nil  [:o-irrelevant-entity.34302]
```

similarities. Also note that "leon" and "paz teachers' leaders" are incorrectly classified. As more training is performed on the ontology, the classifiers become more accurate and make fewer such mistakes. Close the Classifier Summary window.



Entity Tagger

Select which classifier will verify tags: RHM + LPS + NB + AI Committee

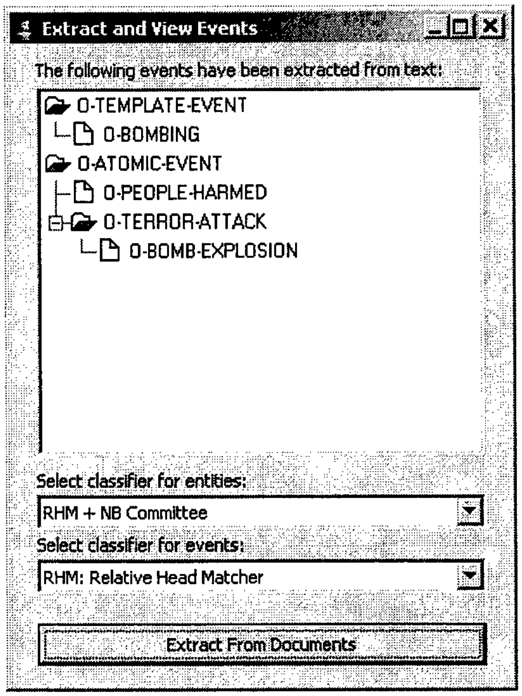
View current entity tags for document: tst3-muc4-0040

Select ontology nodes to tag:
:O-EXPLOSIVE
:O-FIRE
:O-GEO-LOCATION
:O-GRENADE
:O-GUN
:O-HANDGUN
:O-KILLING-INSTRUMENT
:O-MACHINE-GUN
:O-MINE
:O-MISSILE

at the same time , another bomb exploded at a banco popular branch , two blocks away from where the first explosion took place , police said . other banks and shops were damaged by the second explosion , and the security guard at the banco popular was also injured by pieces of flying glass , the authorities said . according to preliminary reports , two men placed the explosive devices and left the scene on foot . both of these attacks tonight , attributed to the drug trafficking cartels , took place after 10 bombs exploded early this morning in the teusaquillo area in downtown bogota . the bombs were placed in nine political offices of the liberal

14. **View Document Tags** – Select document **tst3-muc4-0040** in the document selection combo box of the Entity Tagger interface. The text for that document will be displayed with all classified killing instruments highlighted in red. Note that the tagger missed one, the phrase "10 bombs". A visual inspection of the syntactic tags for that document (viewable via the document viewer if you are interested) reveals that the syntactic parser incorrectly parsed that sentence. Because "exploded" was ignored by the parser completely, no relevant extraction rules could pick the entity up. To see how any level of the ontology can be tagged, select the node :o-bomb for tagging in the node selection combo and hit the **Tag Documents** button. Note that "explosive devices" is no longer tagged in the document because it is an example of an :o-explosive node, at one level of generality above the :o-bomb node. Close the Entity Tagger interface window.

15. **Run the Event Extractor Application** – Click on the **Event Extractor** button. An interface window will appear listing all of the template events and all of the atomic events currently in the system. No instances should yet exist in the system. Choose the classifier suites to be used for event and entity classification during the extraction task. Select **RHM + NB Committee** for the entity classifier combo and **RHM: Relative Head Matcher** for the event classifier combo. (The use of RHM + NB will cause some irrelevant extractions to occur as that classifier is overly permissive, but the errors are somewhat instructive.) Click on the **Extract From Documents** button to run the extractor. This application will apply all of the entity and event rulebases to extract out the entities and events related to bombings according to the training performed thus far, and will correlate the extracted information into instantiated ontology nodes. Once the extractor has run, the interface will display all of the instantiated nodes that were extracted from the test

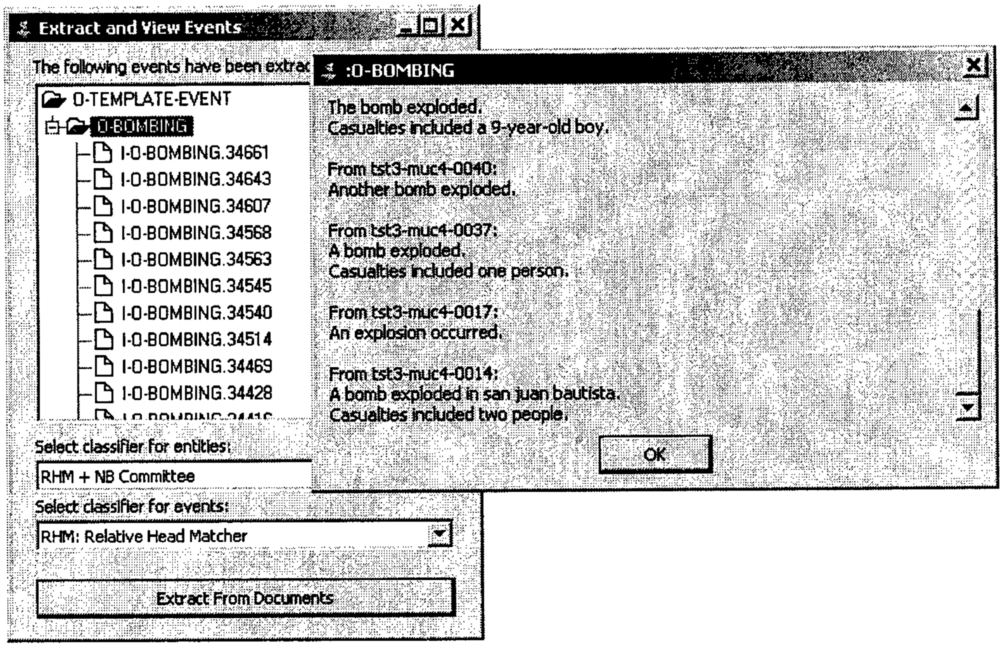


Extract and View Events

The following events have been extracted from text:
O-TEMPLATE-EVENT
└ O-BOMBING
O-ATOMIC-EVENT
├ O-PEOPLE-HARMED
└ O-TERROR-ATTACK
  └ O-BOMB-EXPLOSION

Select classifier for entities: RHM + NB Committee

Select classifier for events: RHM: Relative Head Matcher documents. Double-clicking on any node (an instance node or a category node) will display all of the extractions in English in a pop-up dialog. The nodes are generated into English via a simple canned expression by functionality attached to the abstract entity and event nodes in the ontology. Double-click on the :o-bombing entry. 16 extractions are summarized into English and are displayed. Notice that a couple of incorrect extractions are performed, mainly "the missile exploded in the air" and "The foreign debt crisis exploded in andean countries". These occur because the RHM + NB classifier is overly permissive in its entity classifications. Notice also that some of the extractions occur across sentence boundaries. For example, if you inspect the document **tst3-muc4-0044** via the document viewer, you will see that the information correlated together into the summary "A bomb exploded. Casualties included a woman." spans multiple sentences. Finally, notice that all types of information trained during the bootstrapping phase are represented in the extractions, including both the explosion and people harmed events, and the people, location, and weapon entities. When you have finished examining the extractions, close all of the windows and exit the LISP environment. That concludes the demo sequence.



The bomb exploded.
Casualties included a 9-year-old boy.

From tst3-muc4-0040:
Another bomb exploded.

From tst3-muc4-0037:
A bomb exploded.
Casualties included one person.

From tst3-muc4-0017:
An explosion occurred.

From tst3-muc4-0014:
A bomb exploded in san juan bautista.
Casualties included two people.

Select classifier for entities: RHM + NB Committee

Select classifier for events: RHM: Relative Head Matcher

REPORT OF INVENTIONS AND SUBCONTRACTS

(Pursuant to "Patent Rights" Contract Clause) (See Instructions on Reverse Side.)

Form Approved
OMB No. 0704-0240
Expire Sep 30, 1988

1a. NAME OF CONTRACTOR / SUBCONTRACTOR Stottler Henke Associates, Inc.	c. CONTRACT NUMBER DAAH01-00-C-R113	**2a. NAME OF GOVERNMENT PRIME CONTRACTOR** Same	c. CONTRACT NUMBER Same	**3. TYPE OF REPORT** *(X one)* ☐ a. INTERIM ☒ b. FINAL
b. ADDRESS *(Include ZIP Code)* 1660 So. Amphlett Blvd., STE #350 San Mateo, CA 94402	d. AWARD DATE *(YYMMDD)* 00/03/30	b. ADDRESS *(Include ZIP Code)* Same	d. AWARD DATE *(YYMMDD)* Same	**4. REPORTING PERIOD** *(YYMMDD)* a. FROM 00/03/30 b. TO 00/09/30

SECTION I - SUBJECT INVENTIONS

5. "SUBJECT INVENTIONS" REQUIRED TO BE REPORTED BY CONTRACTOR / SUBCONTRACTOR *(If "None," so state)*

a. NAME(S) OF INVENTOR(S) *(Last, First, MI)*	b. TITLE OF INVENTION(S)	c. DISCLOSURE NO., PATENT APPLICATION SERIAL NO. OR PATENT NO.	d. ELECTION TO FILE PATENT APPLICATIONS				e. CONFIRMATORY INSTRUMENT OR ASSIGNMENT FORWARDED TO CONTRACTING OFFICER	
			(1) United States		(2) Foreign			
			(a) Yes	(b) No	(a) Yes	(b) No	(1) Yes	(2) No
None								

f. EMPLOYER OF INVENTOR(S) NOT EMPLOYED BY CONTRACTOR / SUBCONTRACTOR

(1)(a) Name of Inventor *(Last, First, MI)*

(b) Name of Employer

(c) Address of Employer *(Include ZIP Code)*

(2)(a) Name of Inventor *(Last, First, MI)*

(b) Name of Employer

(c) Address of Employer *(Include ZIP Code)*

g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED

(1) Title of Invention

(2) Foreign Countries of Patent Application

SECTION II - SUBCONTRACTS *(Containing a "Patent Rights" clause)*

6. SUBCONTRACTS AWARDED BY CONTRACTOR / SUBCONTRACTOR *(If "None," so state)*

a. NAME OF SUBCONTRACTOR(S)	b. ADDRESS *(Include ZIP Code)*	c. SUBCONTRACT NO.(S)	d. DFAR "PATENT RIGHTS"		e. DESCRIPTION OF WORK TO BE PERFORMED UNDER SUBCONTRACT(S)	f. SUBCONTRACT DATES *(YYMMDD)*	
			(1) Clause Number	(2) Date *(YYMM)*		(1) Award	(2) Estimated Completion
None							

SECTION III - CERTIFICATION

7. CERTIFICATION OF REPORT BY CONTRACTOR / SUBCONTRACTOR *(Not required if* ☒ *Small Business or* ☐ *Non-Profit organization.) (X appropriate box)*

a. NAME OF AUTHORIZED CONTRACTOR / SUBCONTRACTOR OFFICIAL *(Last, First, MI)*

c. I certify that the reporting party has procedures for prompt identification and timely disclosure of "Subject Inventions," that such procedures have been followed and that all "Subject Inventions" have been reported.

b. TITLE

d. SIGNATURE

e. DATE SIGNED

Previous editions are obsolete.