A Proposal for Knowledge-Based Labeling of Semantic Relationships in English

Alicia Tribble

March 20, 2005

Contents

1 Introduction **3**

2 Thesis **5**
- 2.1 Linguistic and Semantic Scope . . . 5
- 2.2 Task-based Scope . . . 7
- 2.3 Summary . . . 8

3 The Scone Knowledge Representation **8**
- 3.1 Types, Individuals, and Roles . . . 9
- 3.2 Relations and Actions . . . 10

4 The Graph Approach **10**
- 4.1 A General Algorithm and Example . . . 10
- 4.2 Selecting Scone Elements . . . 12
- 4.3 Combining Semantic Substructures . . . 12

5 Semantic Distance Functions **13**
- 5.1 Static and Dynamic Functions . . . 13
- 5.2 Incremental and Non-incremental Functions . . . 14
- 5.3 Theory-based and Corpus-based Functions . . . 14
- 5.4 Training Data . . . 14

6 Developmental Plateaus and Evaluations **15**
- 6.1 Noun Compounds . . . 15
- 6.2 Semantic Role Labeling . . . 15
- 6.3 Restricted Queries and Sentences . . . 15
- 6.4 Advanced Topics . . . 16

7 Related Work **16**
- 7.1 Early NLU . . . 16
- 7.2 OntoSem . . . 18
- 7.3 Cyc . . . 18
- 7.4 Open Mind . . . 19

8 Timeline **19**

1 Introduction

This proposal introduces a method for producing meaning from natural language in a knowledge-based framework. Semantic analysis is a fundamental task for many applications, including machine translation, question answering, essay scoring, and tutoring. Systems that approach this problem without an underlying body of world knowledge may provide useful analysis for domain-specific tasks. But they can not provide an integrated model that reasons with new facts in the context of the world as we know it. In addition, a large, inference-capable knowledge base (KB) is perfectly suited to assist in some of the hard tasks of NLP: anaphora resolution, nonliteral usage, and lexical and structural ambiguity. Consider the sentence: "John said he can't make his appointment." Who is "he"? Which appointment? Is he unable to set a time, or will he simply be late for a meeting that is already scheduled? A persistent knowledge base, with representations for John and his activities, will help to interpret the statement correctly.

Our solution to the problem uses Scone (Fahlman, 2005), a knowledge representation and inference engine, to drive an analyzer of English statements and queries. Results from the system are grounded in the current world model, which is stored in a Scone knowledge base. Our research focuses on implementing an analysis engine that leverages semantics as much as possible. In addition, we we apply it to a range of NLP problems: reference resolution (anaphora and some quantification), metonymy, and unknown words. The system will be tested on higher-level tasks including knowledge base authoring. We also plan to compare the approach with statistical systems for semantic analysis.

There have been many efforts to represent statements and queries in terms of a background knowledge base. They include early database-accessors like BASEBALL (Green et al., 1961) and LUNAR (Woods, 1973), as well as NL interfaces to more complex inference engines like ThoughtTreasure (Mueller, 1998) and Cyc (Witbrock et al., 2003). Still, interesting problems remain in terms of how to use the KB most effectively. First, there is a general task of determining what "correct" semantic output should be. For a KB of non-trivial size, many reasonable representations of the same surface string may exist. It can be difficult even for humans to settle on the canonical semantic representation of a sentence. This entails identifying which KB concepts are present (as in word-sense disambiguation), as well as finding relationships between them that are licensed, according to the knowledge base.

One popular way to approach the problem is with syntactic parsing, augmented with semantic features that come from the knowledge base. The details of the grammar formalism may vary. KBNL (Barnett et al., 1990) used categorial grammars, while KBMT (Goodman and Nirenburg, 1992) and AUTOSEM (Rosé, 2000) use LFG-style grammars (lexical-functional grammar).

In our own preliminary work, we built a system of this type. We used the LCFLEX parser (Rosé and Lavie, 2001) with an LFG-style grammar of English. This work establishes the fitness of Scone for NLP tasks. The system communicates with Scone via call-out functions embedded in the grammar. A valid semantic structure is returned for each syntactic constituent. When no semantic value is found, the system may return NIL, or the user may be queried for more information.

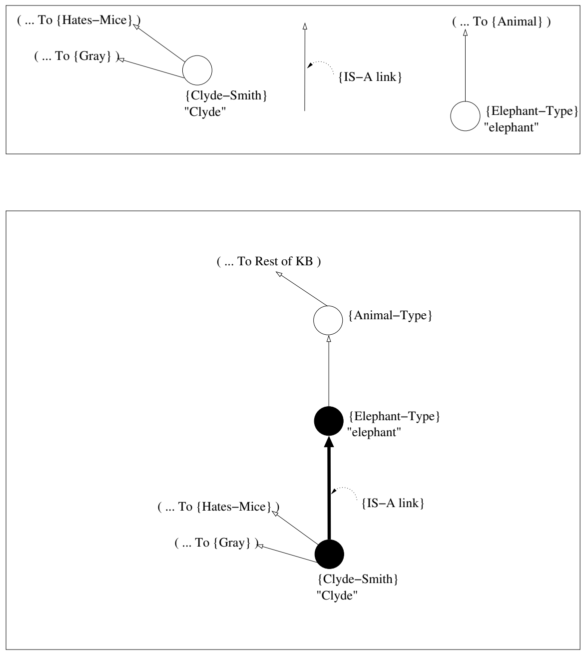
This type of solution is a starting point for general semantic analysis, but it has shortcomings. First, writing enough hand-written rules to cover general English is time-consuming. This is the case even when the syntactic rules are already in place, as they were in our system. Managing the interaction of syntactic rules with the knowledge base is difficult to do by hand. And while the sentences that are parsed do result in legal semantic forms, sentences which are not covered by the grammar produce no result at all [1].

Second, we fail to leverage the full power of the knowledge base. We would like to find a semantic link even when syntax is tenuous or missing. Given a sparse set of clues like "Thief. Careless. Prison.", we still want to find some connection between them. This motivates our choice to drive the analysis process with semantics, using syntactic structures to help when they are available.

We therefore propose an approach that frames the problem as a search for the lowest-cost semantic

[1] This is a shortcoming of the grammar rules, although not of syntactic parsers in general. Robustness features of parsers like LCFLEX include partial parses and word-skipping, in cases where the entire input is not covered by the grammar. However in cases where a complete semantic form is necessary but cannot be generated, NIL results may be preferable to these partial syntactic solutions.

Figure 1: Simplified analysis process for the English sentence “Clyde is an elephant.” First, a set of candidate concepts from the knowledge base is instantiated. We may consider them temporarily isolated from their connections to the rest of the KB, as in the top figure. In the lower figure, semantic analysis has placed them in context with respect to each other and to the rest of the knowledge base. The result is shown in bold.



structure, given an English sentence. Word meanings [2] are generated by instantiating concepts from the knowledge base. Phrase- and sentence-level meanings are built by searching Scone for a semantic path that connects these concepts. “Cost” is determined by assigning weights to each link between elements in the combined structure. The style is similar to case-frame parsing approaches like Conceptual Dependency (Schank and Tesler, 1969) and the Ontological Semantics systems (Nirenburg et al., 2003). This approach does maintain legal representations for each of the sub-parts of a parse, since combinations only occur when they are licensed by the KB. It is also driven by semantics, since new candidate meanings are generated through semantic operations. An small example of the process is shown in Figure 1. The diagram has been simplified for illustration.

One advantage over the other case-frame systems mentioned above is the semantic formalism itself. Scone is structured as a semantic network. Its desirable features include efficient methods for inference and search, and a structure for referring to statements recursively. An overview of the relevant parts of Scone are given in Section 3.

In addition, we provide a measure of semantic relatedness (i.e. the cost of a candidate semantic structure) that can be applied in a consistent way across many levels of analysis. In real-world data, inconsistencies occur often between natural language utterances and the KB representations used to model them. The work in (Fan et al., 2003) describes the effort required just to transform a naïve semantic encoding – one that is intuitive and nearly correct – into one that actually conforms to the structural requirements of a specific knowledge representation. This mismatch between naïve and correct encodings affects the interpretation of queries as well as statements.

[2] Including multiword expressions and idioms

Although we can not solve every case of mismatch, we can provide some flexibility by training our search parameters on sample data (when searching for the best meaning). Training will adjust the priority of certain semantic network links according to what is most prevalent in the current domain. A similar approach, called OntoSearch, is described in (Onyshkevych, 1997). In comparison, we propose a more detailed investigation of these priorities and weighting schemes for them. We also broaden the range of problems solved by our ontological-semantic search.

2 Thesis

The exposition above describes our motivation for building a general, knowledge-driven semantic analysis engine. Our hypothesis is that such an engine can provide analysis of simple statements and queries that help knowledge engineers make changes to the Scone knowledge base, improving their comfort and efficiency as compared to low-level lisp calls. In addition, the knowledge can itself be a tool for solving NLP problems including reference resolution, metonymy resolution, and unknown word handling. We propose that dealing with these problems can contribute to performance on higher-level tasks like entity tracking in email, among others.

The proposed work is bound along two dimensions: the range of English linguistic and semantic phenomena that we plan to cover, and performance on higher-level tasks, like entity tracking in text, question-answering, or user dialogue, that we will support and test on.

2.1 Linguistic and Semantic Scope

Our goal is to explore the capabilities of this style of analysis. Because our focus is on driving the process semantically, we begin with a case where syntax has a limited role in the meaning of the phrase: noun-noun compound resolution. Nominal compounds have been explored in several computational frameworks, including NETL, the precursor to Scone. (McDonald, 1982), presents a rule-based search for noun compound meaning. More recently the work of (Fan et al., 2003) reported that some levels of an ontology [3] are more important than others for a knowledge-based approach to this task. Our own completed work on this topic indicates that a search through the ontology can be feasibly executed, even with few syntactic clues for pruning.

When more syntax is available, more engineering is required to put that knowledge to use. A general sentence-level analyzer will be developed in several phases. First, we will focus on simple queries to the KB. These queries are restricted to those with the following properties:

Semantic Restrictions

- All vocabulary is known and concepts are available to the current KB. However, vocabulary may be ambiguous.
- Unambiguous concept names from the KB may be given in curly-brace notation, e.g. {eat-action-1}.
- Sets or classes must refer to defined types in the KB, not subsets of those types. This corresponds to a syntactic restriction on demonstrative pronouns and quantifiers.
  (*Allowed:* Do elephants eat grass? *Disallowed:* Do those elephants eat grass?)
- Temporal ordering is not represented in the semantic analysis. All actions are assumed to occur in the present.
- All semantic role-filling entities are assumed to appear in the current phrase – semantic ellipsis is not allowed.

[3] Our knowledge base supports multiple inheritance in addition to taxonomic information. For the purposes of this document we will use "ontology" and "knowledge base" interchangeably.

Figure 2: Restricted Queries

Does a business transaction have participants?
What are the roles of a negotiation?
What are the sub-types of animal?

Figure 3: Restricted Statements

A driver is also a person.
Oregon is a state.
The owner of a pet is a pet-owner.

Syntactic Restrictions

- Queries are limited syntactically to one clause. Later experiments will explore linking multiple clauses from a single sentence.
- Nonliteral usage, including metonymy, is not allowed.
- Quantification (non-existential) and comparison do not appear.
(*Allowed:* Are there bears in Oregon? *Disallowed:* Do bears eat more than people do? *Disallowed:* Does it rain a lot in Oregon?)
- Anaphora must be resolved inside the current clause.
(*Allowed:* John tied his shoe. *Disallowed:* John ate it.)
- Coordination is restricted to subject noun phrases. This constraint is related to the single-clause constraint and to the quantification constraint above.
(*Allowed:* Joe and Sue went to Panama. *Disallowed:* Joe and Sue took money and cigars to Panama and Spain)

These queries, although limited in scope, are still useful to knowledge engineers. The queries provide access to the current state of the KB. This is important in the case of a large knowledge base that is being augmented by hand. Simple queries can help engineers to add knowledge that is consistent with the existing model. Some examples of queries that are covered in this stage are given in Table 2. In addition, the constraints still allow for word sense ambiguity, prepositional attachment ambiguity, and syntactic ellipses. The basic system must handle these issues before more complicated problems can be approached. An important milestone of the thesis is to prove that our general algorithm for semantic analysis can cope with these NLP issues. The algorithm itself is described in more detail in Section 4.

The natural extension of this stage is to include statements (as opposed to only queries) which abide by the same restrictions given above. Examples of such statements are given in Table 3. They allow users to make small changes to the current KB, perhaps in response to mistakes they detect using queries. Restricted queries and statements comprise the coverage goal for a basic implementation of the system. Once this work is in place, we can use it to explore harder NLP problems.

One of the main benefits of using a knowledge-based approach for semantic analysis is that the resulting meaning is linked to a larger world model. Getting this link right is one of the most important issues we face. For that reason, we will address some linguistic topics in detail that can contribute to better performance on reference resolution. The authors of (Beale et al., 2004) outline some of the issues involved in resolving referring expressions with an ontology. They note that entities known to the system may be referenced by indirect means, including *descriptions* and *pointers*. Resolving such expressions should be a strength of our knowledge-based semantics approach and will be a focus of the thesis experiments.

Specifically, we will expand our coverage of quantification, nonliteral usage, and unknown words. Each of these phenomena was restricted in the examples given above. Of these, quantification is perhaps the most difficult but also the most important. Certain quantifiers and determiners play an important role in referring expressions: *Some of the members, Those people*. Quantification outside of referring expressions will not be explored in detail because the problem is so large, but it is a good example of a topic for future work.

Metonymy is a type of nonliteral usage where part of an entity stands in for the whole, or vice-versa (ex: *The U.N. voted on the resolution*, where *U.N.* stands for *the people who are representatives at the U.N.*). Because the mismatch is related to a type hierarchy, several groups have applied ontological reasoning to the problem of metonymy resolution, including (Onyshkevych, 1997) and (Barnett et al., 1990). Because of its frequency and relative simplicity, it should also be addressed by the proposed system.

The third topic we plan to address in detail is constructing partial meaning structures for phrases with unknown concepts. Again, this should be a strength of a knowledge-based system. How can we use the KB to find likely properties of unknown words, and to link them to existing definitions when appropriate? At minimum, we may use the KB to guess the type of an object based on selectional restrictions: given a sentence like *Joe ate a kumquat*, we might guess that *kumquat* is a *food*. Can we do any better than this? How will it affect the ability of the system to cope with new vocabulary?

These topics alone define a set of problems we plan to explore in detail after the basic system is in place. Several of the other restrictions we placed on queries and statements will not be addressed in the space of this thesis; namely, temporal and tense issues, deep topics in quantification, and multiple coordinated phrases, among others.

2.2 Task-based Scope

Linguistic issues limit the range of English language phenomena we expect to cover in this proposal. But the contribution of the system can also be described in terms of higher-level tasks. These are the cases where we expect our system to produce helpful output that supports other activities like entity tracking in text corpora. The number of tasks will grow over time, expanding in parallel with the linguistic coverage.

The basic system that supports restricted queries and statements can already be a useful tool for interacting with the KB. The goal of the system at this stage is to aid users who are somewhat familiar with the current knowledge base; for example, knowledge engineers who are in the process of adding facts to the KB. The system will be evaluated on how well it covers the queries and statements such users make. In addition, comparisons can be made between users who access Scone through the English interface and users who rely on the lisp calls provided with the Scone engine.

We can reasonably expect even the restricted natural language interface to be helpful as an accessor to the KB. Controlled language has already been used successfully in knowledge-based NLP systems like KANT (Nyberg and Mitamura, 1992) and KANTOO (Nyberg and Mitamura, 2000) for interlingual Machine Translation (MT). The KANT controlled language (CL) has restrictions similar to ours, including known vocabulary and simplified syntax (for details on the KANTOO CL, see (Nyberg and Mitamura, 1995)). This sets an encouraging precedent for our work.

The next plateau in system development will be added support for specific phenomena like indirect reference resolution and unknown words, mentioned above. The task-based goal of this phase is to provide semantic analysis of user input to the RADAR project (CMU-RADAR, 2004) at Carnegie Mellon University. RADAR is a software personal assistant that includes intelligent tools for email management, personal scheduling, and resource allocation (i.e. meeting rooms and other facilities). A deep repository of contextual information is absolutely necessary for a machine to respond intelligently in these tasks. Our vision for the Scone knowledge base in RADAR is that it will store persistent user-specific world models. The English language analyzer we propose here will provide direct access to these models based on user input.

The English analyzer can be evaluated in this context based on how well it identifies entities that appear in emails and in transcribed user dialogues with the RADAR tool. Precision and recall on reference resolution, as well as linguistic coverage of the queries and statements that appear there, will be measured and reported. Other tasks within RADAR may include acquisition of knowledge from user statements, or question-answering in response to queries. The RADAR project is currently under development, and planning and testing the system on these additional tasks may be topics for future work (outside the thesis) in coordination with the RADAR team.

Finally, we would like to compare our system to current empirical systems for semantic analysis. Automatic labeling of semantic roles, as introduced in (Gildea and Jurafsky, 2002), is an empirical analog to the knowledge-based analysis we propose. Semantic roles (semroles) are labels for the relationships between a verb and its arguments. They usually include *Agent*, *Patient*, *Theme*, *Location*, and *Goal*. Large labeled corpora like PropBank (Palmer et al., 2003), VerbNet (Kipper et al., 2000), and NomBank (Meyers et al., 2004) have helped drive research in this area. Semrole labeling in this style has also been the focus of several recent competitions and workshops, including the shared task for CoNLL-2004 (Carreras and Màrques, 2004) and CoNLL-2005 (upcoming).

The work on semrole labeling is an important piece of the contemporary landscape in semantic analysis. Comparing it with our own system will provide insight into the value of deep knowledge from Scone, versus generalization power from training on large corpora. Ideally, we will discover ways to leverage both. For example, we have already prepared experiments that use Scone representation to encode training data from NomBank. In tests, our system is given a head word and an argument. Our goal is to instantiate Scone concepts corresponding to these words, and to find the best path through the Scone KB that connects them. The experiment requires hand-editing of the knowledge base to accommodate new concepts from the training data. This knowledge then guides the search for a connection between two concepts. The connection is meaningful and grounded in the world model of the KB. In contrast, a statistical model would classify the connection between concepts using a small vocabulary of semrole labels. However the statistical model requires no hand-editing. We would like to determine whether the knowledge-based result (i.e. the full Scone connection between two concepts) can contribute to higher accuracy in the task than statistics alone. In addition, as the background knowledge base grows, the incremental cost of adding concepts for a new domain may decrease. This would make the knowledge-based approach even more widely applicable.

2.3 Summary

In summary, our plan is to build a knowledge-based semantic analysis system that can address restricted queries and statements. We will evaluate the performance of the system based on English coverage as well as by testing its contribution to higher-level tasks like entity tracking in email. We will also compare the system to empirical systems for semantic role labeling.

In the remainder of the proposal, we give implementation details of the system. The next section gives an overview of the relevant parts of Scone. Development and maintenance of the Scone inference engine lies outside the scope of this proposal. However understanding the formalism is important for framing our work. Section 4 describes an approach based on graph-style traversal of the Scone semantic network. This is the general algorithm which we will use in our evaluations. Section 5 describes the space of semantic distance functions which we plan to explore. We expect these functions to play a major role in the success of the general system. Finally, we describe the task of semantic role labeling. Although performance evaluations on RADAR tasks are also an important part of our proposed work, the details of these tests will have to be planned when the RADAR system itself is more fully developed. This should be no later than one year from the proposal date.

3 The Scone Knowledge Representation

The Scone representation system comprises the Scone representation language and the Scone inference engine. The engine implements low-level commands, like concept instantiation and look-up. In addition, the full representation system may include Scone knowledge bases (KBs), which are text files that make calls to the engine functions. The KBs are written by knowledge engineers and store the actual ontologies or world models.

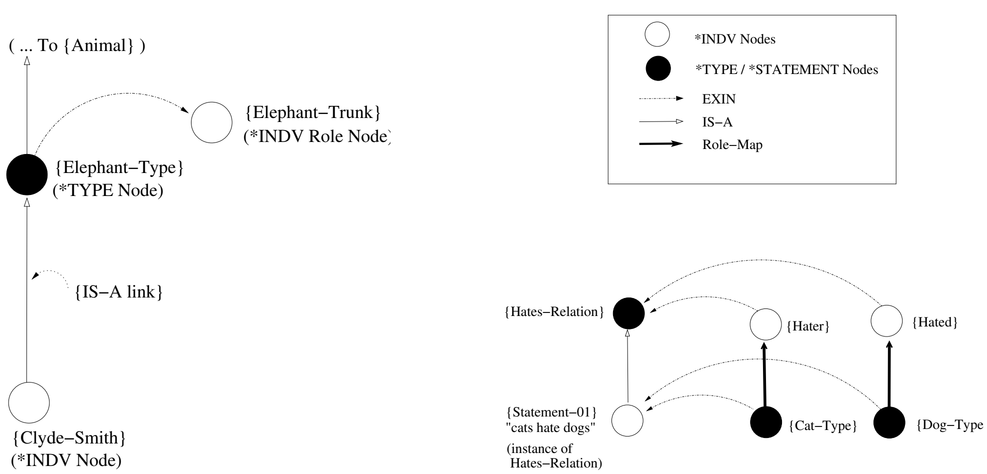
Scone represents the links and nodes of the semantic graph with a data structure called an element. Each element has a unique identifier or system-internal name (iname). Inames are represented in the Scone syntax with curly braces; we adopt the same convention in this paper. Each element is also associated with a list of non-unique external names. The external names may be English or other natural-language strings, and a single string may be associated with multiple concepts. For example, an element {clyde-smith} may appear in text as "clyde," "smith," or "clyde smith." These strings will all go into the list of external names associated with the {clyde-smith} element. If another "clyde" appears in the knowledge base, the string will become ambiguous, but the inames remain unique. This many-to-many mapping mirrors the lexical ambiguity problem inherent in natural language. Semantic graph edges in Scone represent relations between concepts. They can be taxonomic in nature ("A is-a B") or may describe a non-structural relationship between concepts ("A likes B").

3.1 Types, Individuals, and Roles

An example Scone structure is given in Figure 4. Elements of three different types occur here: a *TYPE node represents the typical member of a class of objects ("Elephants"); an *INDV node represents a discrete object ("Clyde") or other entity ("the War"); a *ROLE node is a place holder that links a *TYPE or *INDV to its owner ("Elephant's Trunk"). Directed links of two types appear: IS-A links appear as solid lines and EXIN links ("exists-in" or "context" links, linking roles to their owners) appear as dash-dot lines. This kind of diagram is the most intuitive way to give a snapshot of a Scone meaning representation. Later examples will use the same diagram conventions for node and link types. These are simplified examples; the facts stored in an actual Scone knowledge base will depend on what the author of that KB writes there.

Figure 4 represents the free-standing individual {Clyde} and shows that he is an {Elephant}. He inherits all the roles of his parent class, in this case {Elephant-Trunk}. Because Scone inheritance is transitive, {Clyde} may also inherit things like a heart, lungs, and nervous system from the class Animal or weight and volume from the class Physical Object. An important feature of the Scone engine is that it includes functions for listing and traversing these role-relations. This facilitates concise search for nodes with nested descriptions, for example "The tires of Clyde's mother's car."

Figure 4: Scone representation of "Clyde the elephant" and "Cats hate dogs"

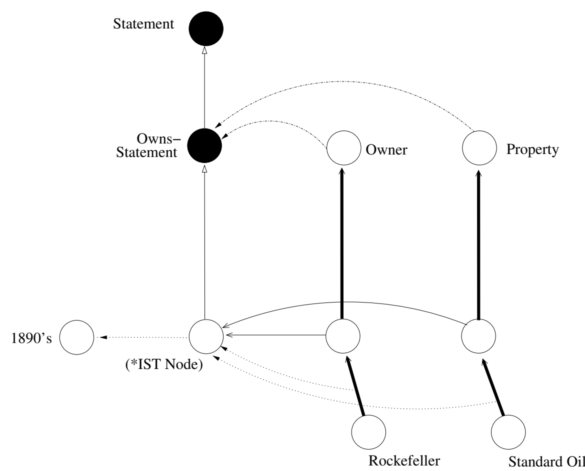


3.2 Relations and Actions

Actions and temporally-bound states are represented with *RELATION nodes and *STATEMENT links. Relations can be defined by the user and must have restrictions on the element types of their domains and ranges. A statement instantiates a relation between two type or indv nodes. Figure 4 shows a Scone representation of "Cats hate dogs." This representation would allow the inference engine to reply "True" to the query "Does Fluffy hate Fido?".

Statements and actions can also be scoped in time or in location. Figure 5, adapted from (Fahlman, 1979), shows the statement "Rockefeller owned Standard Oil in the 1890's." The *IST node is an individual node descended from the *STATEMENT node type.

Figure 5: "Rockefeller owned Standard Oil in the 1890's"



These examples cover the basics of the Scone representation. More complex issues for the meaning representation will be addressed in terms of their effect on our analysis algorithm, in Section 4.

4 The Graph Approach

4.1 A General Algorithm and Example

Given the semantics described above, we want to design a semantic structure-building algorithm that uses the graphical nature of the ontology to advantage. A general algorithm along these lines proceeds as follows: first the sentence is scanned for strings that trigger the instantiation of ontology concepts. Each of these instances is linked to the ontology itself via its parent concept. Next a search procedure looks for the least-cost path through the ontology that links enough instantiated concepts to cover the input sentence. This path corresponds to the meaning that best satisfies the constraints already present in the knowledge base. To implement this approach we need two main components: a control structure that keeps track of token coverage in the input along with partial meaning structures, and an internal function that gives the cost of a single path in the semantic net.

Figure 6 gives a small example ontology with three instances, seven types, two individual roles, and links labeled IS-A or ROLE. Figure 7 shows the set of elements from this ontology which will be instantiated for the compound noun phrase "glass wine glass" (notation for nodes and link types is consistent with Figure 4). In this example ontology, simplified for illustration, the word "glass" may refer to a drinking container or to a substance. The word "wine" has only one meaning, referring to the liquid. The actual ontology may have many more links. In addition, each meaning for "glass" is instantiated twice, since each appearance of

Figure 6: Example Scone KB

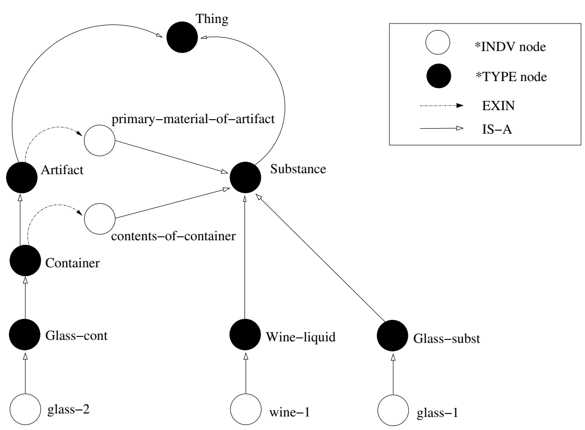
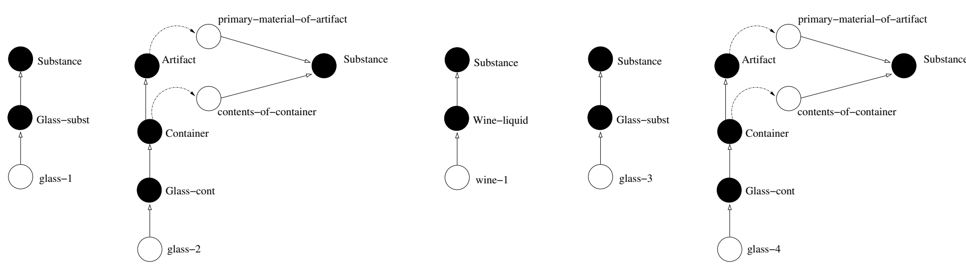


Figure 7: KB instantiations for the NP *glass wine glass*



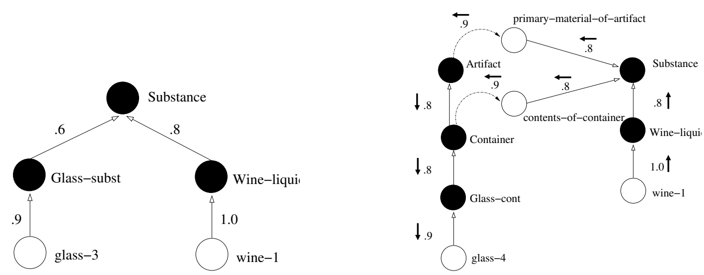
the word in the test phrase triggers a new instantiation. During processing these instances may be collapsed if, for example, they are found to refer to the same entity.

Using the syntactic cue that the head of a noun compound is usually the rightmost token, we first try to link "wine" with the final "glass" by searching for the least costly path between them through the ontology. Figure 8 compares the path from element *wine-1* to element *glass-3* with the path from *wine-1* to *glass-4*. The search compares multiple paths using the word sense "glass-container" as well as the path using the word sense "glass-substance". The weight associated with each edge in this example is interpreted as a closeness score on the interval of [0.0, 1.0]. These scores were assigned by hand for illustration. They depend on the direction of travel (from filler to constraint), the identity of the current edge (i.e. the ontological link), and previous edges in the current path. The cumulative cost for a path is given by the product of these scores.

This example was treated in earlier work on resolving noun compounds with NETL (McDonald, 1982). The approach used in that work relied on hand-coded rules for preferring one semantic link over another. The system eventually covered a reasonable range of noun compound relationships but sometimes failed entirely, in cases where no rule could apply. The weighted search used in our approach should allow us to return some answer for every test query, although some will be very unlikely.

One of the main objectives of the thesis is to explore the space of control structures and path weighting functions for this kind of approach, and to implement a working system based on our findings. Details on our plans for the weighting function are given in Section 5. Below we describe the control structure, which includes two main processes: a procedure for selecting the Scone elements that will contribute to the meaning

Figure 8: Weighted paths for *wine glass-3* and *wine glass-4*



of a sentence, and a combination procedure for building larger semantic structures from these pieces.

4.2 Selecting Scone Elements

A Scone element can be triggered by an English term that appears in the element's :english tag. It can also be triggered by a link from an external lexicon. Curly brace notation is supported so that users who do know the internal name of a concept may specify it unambiguously. Idiomatic expressions are handled in the lexicon. This machinery for triggering Scone elements in response to English strings is the same one used in our early grammar-based system (Section 1). It is part of our completed work that can be re-used in the new approach.

4.3 Combining Semantic Substructures

Once the set of Scone elements is instantiated, we need to execute a search for the least-cost connected path among them. One architecture option for this process would be a heuristic graph-building search similar to the searches that drive syntactic parsers (although with a different style of constituent). In this case, attention must be paid to the tangled hierarchy structure of Scone: it includes cycles and crossing paths. A control structure along these lines was used in (Barnett et al., 1990). A two-dimensional syntactic chart was augmented with semantic information, creating a three-dimensional space for best-first search.

As an alternative, we could use the Scone inference engine itself to control the search process. This approach takes advantage of efficient graph traversal functions (called scans) that are built into the representation itself. We are able to scan up or down the ontology from a semantic node, evaluating the links and properties of nodes that we find as we go. In order to convert such a traversal into a cost estimate for the combination of two semantic sub-structures, we still accumulate edge weights as we proceed from a slot in one structure to a candidate filler in another. Distances are compared at the end of an entire ontology scan.

A third style of control architecture for this process is presented in (Beale et al., 1996). It was used for the analysis module of the Mikrokosmos MT system (Beale et al., 1995). To find the meaning of an input sentence, the algorithm first uses constraints that come from the ontology to prune the search space of meaning combinations. Next, a solution synthesis technique is used in place of search to find the best combined meaning. This approach is more efficient than best-first search and could be a useful framework for us to explore.

In developing a full system, we will start with the scan-based control structure, using a static weighting function for the path costs as a placeholder. Once this control structure is in place we will have a framework for exploring the next major component of the system, the training procedures for ontological distance functions.

5 Semantic Distance Functions

In the example given in Figure 8, the least costly solution is a combined semantic structure where the element *wine-1* fills the role of *contents-of-container*, which *glass-4* inherits from *Container*. The example demonstrates the fact that the number of links in a path alone is not sufficient for determining the best meaning; the path weights make the algorithm successful. One of the major tasks of the thesis is to investigate functions for assigning these weights, along with the most appropriate training procedures for learning weight functions from data. These functions represent semantic distance according to the current ontology. We will use the terms “ontological distance function” and “weighting scheme” or “path weighting scheme” all to refer to the same process: the process by which a cost is assigned to a path through the knowledge base from one given concept to another. In the section below, we will describe the types of distance functions we plan to explore. Note that for any of these functions, a path or partial path is provided by Scone, and a score is returned. There are no permanent weights attached to the Scone element structure.

5.1 Static and Dynamic Functions

Measures of semantic network distance can be classified according to how they assign the weight of each link that lies on the current path. They may be static, meaning that the same link (or class of links) in the knowledge base receives the same weight every time it is encountered. This type of function includes simple link-counting, which assigns the same weight to every edge [see (Rada et al., 1989) for an example]. They may also be dynamic, meaning that the function can assign a different weight to a link each time it is seen. An example of a static weighting function might be a lookup table like the one in Table 1. Such a function gives the of a path link, conditioned only on the name of the link. A dynamic function would condition the weight on the link name and some additional features of the path. Table 2 gives an example, where the weight is conditioned on link name and the number of times that link has appeared so far in the path.

Table 1: Static weights

IS-A	1.0
INV-IS-A	0.8
HAS-A	0.6
INV-HAS-A	0.4
ROLE-OF	0.9
INV-ROLE	0.4

Table 2: Dynamic weights (dependent on number previously seen, in parens.)

IS-A	(0)	1.0
IS-A	(1)	1.0
IS-A	(>1)	1.0
INV-IS-A	(0)	0.8
INV-IS-A	(1)	0.6
INV-IS-A	(>1)	0.4
...		

Our hypothesis is that static weighting functions will be insufficient for good performance on the overall task of ranking candidate meanings. The earlier work by Onyshkevych in (Onyshkevych, 1997) supports this claim. Although some static weighting schemes will be tried for comparison, we will focus on the dynamic class.

5.2 Incremental and Non-incremental Functions

In the class of dynamic weighting functions, we can further separate incremental functions from non-incremental ones. An incremental weighting function assigns a link weight conditioned only on the links that precede it in the path. Table 2 is incremental, since the number of times a link name has been previously seen is a feature of the preceding path. In contrast, we could assign a link weight based on the number of times the link was seen in the entire path, before and after the current link. This would be non-incremental, since we can't assign a weight until the whole path is complete. Such a weighting function would still be useful for ranking full paths against each other, which is the real goal of the distance function.

Other examples of non-incremental semantic distance functions include the Hirst-St-Onge metric (Hirst and St-Onge, 1998), which conditions the distance on the path length as well as the number of times the path "changes direction". Direction change can be seen in Figure 8, where the links ascend the type hierarchy, then cross role relations, and finally descend the type hierarchy. The Hirst-St-Onge metric and other metrics for semantic distance were compared in (Budanitsky and Hirst, 2001). This task-based comparison was made in the context of detecting spelling errors in text. But it introduced metrics from a range of categories, including two that were empirically trained.

5.3 Theory-based and Corpus-based Functions

A researcher can implement the distance functions we have discussed so far without ever looking at an example of the text he wants to analyze. These functions require an inventory of the link names that will appear, so that a weight can be assigned to each one. But these weights can be assigned by hand. They may be based on some intuition or theory about which links are more important than others in the ontology. For this reason we will refer to functions that do not require training data as "theory-based".

More interesting functions are ones that reflect some statistical knowledge about the kind of English text we want to analyze. These corpus-based functions will be the main focus of our experiments. This is what places our work in a hybrid category between statistical and purely rule-based NLU systems. Our goal is to take advantage of the broad coverage of the former and the indispensable world knowledge encoded in the latter. Corpus-based functions for ontological distance were introduced by Resnik in (Resnik, 1995). His distance function used the path between two concepts (actually a lowest-common-ancestor feature, but we consider this to be path-based) along with concept-frequency counts from a training corpus. Some corpus-based functions were also explored in (Onyshkevych, 1997). This work was applied to metonymy resolution, rather than as a general tool for semantic analysis. The weighting functions relied on hand-created classes to reduce the number of weights that had to be trained.

Our work will be to extend these approaches and test them in our semantic analysis system with Scone. We mentioned earlier that the control structure will be built as a first step. Exploring the performance of corpus-based semantic distance functions within that framework will be another important research and implementation task for the thesis.

5.4 Training Data

An side effect of any of these data-driven strategies is that they transfer the human effort that was focused on grammar writing in phase one to the task of labeling data. This is arguably a much more efficient use of human knowledge, as the emergence of empirical methods in all fields of NLP seems to indicate. In order to propose any corpus-based solution, it is critical to understand what kind of data will be required for training it.

Data for our purposes consists of hand-labeled paths through the Scone ontology that indicate the preferred interpretation of a phrase or sentence seen in training. A sample generated from the training phrase “glass wine glass” is given in Table 3. This data can then be used to train parameters of a variety of weighting functions.

Table 3: Training Data for Corpus-Based Distance Metrics

Origin	Destination	Path
glass-1	glass-2	glass-1; IS-A; Glass-subs; IS-A; Substance; INV-IS-A; primary-material; INV-ROLE; Artifact; INV-IS-A; Container; Glass-cont; INV-IS-A; glass-2
wine-1	glass-2	wine-1; IS-A; Wine-liquid; IS-A; Substance; INV-IS-A; contents-of; INV-ROLE; Container; INV-IS-A; Container; INV-IS-A; Glass-cont; INV-IS-A; glass-2

6 Developmental Plateaus and Evaluations

The scope of the proposal in Section 2 describes an incremental expansion of linguistic coverage. As the coverage expands, we want to assess whether the system performance improving on higher-level tasks. In some cases this requires dropping improvements into an existing system. In others it requires data collection and evaluation design.

6.1 Noun Compounds

We already have some completed work implementing the rule-based search introduced by in (McDonald, 1982). Test material from that work, along with a knowledge base that covers the test concepts, already exist. This makes the noun compounding task a good starting platform. First, we need to build up the control structures for path search and weighting. Once the infrastructure is in place, we can test our accuracy on finding the correct path through the KB and assigning it the lowest cost. This includes the task of labeling a training set with correct paths.

6.2 Semantic Role Labeling

The CoNLL-2005 is sponsoring a shared task on semantic role labeling, starting with the release of training data this month (February, 2005). A system that uses external resources, like the Scone knowledge base, would fall into the “open” track of this competition. Semantic role labeling is a topic of ongoing interest that we will explore throughout the work of the thesis. But because the competition is being held this Spring, we will push this topic to the top of the research plan, after the baseline path-weighting code is in place.

Preparing a system for this competition will involve several steps: Import the frame vocabulary given in the CoNLL training data into a Scone knowledge base, take a sample of the training data and hand-label paths for this data through the Scone KB, examine the paths that result for correlation with semrole labels. If the Scone paths are predictive, then we have some chance of contributing to performance on this task. We can test this much of the system even before implementing the control structure that finds all connections for a full English phrase.

6.3 Restricted Queries and Sentences

Our target for covering restricted queries and sentences is to help knowledge base authors. First, we need to discover the range of sentences used in this task. We need a collection of user data for one of the target domains in RADAR: appointment scheduling, resource allocation, or event planning. Given the English text, we can examine the coverage of the system in terms of KB concepts, English vocabulary, and phrase-level constructs.

The implementation goal for this phase is to explore control structures. Code for finding the connection between two Scone concepts is developed for the noun compound system in Section 6.1. Given longer phrases, we want to re-introduce syntactic cues where appropriate. The control structure gives us a framework for choosing which concepts to connect next. A case-frame parsing approach like the one given in (Carbonell and Hayes, 1992) will be a starting point.

6.4 Advanced Topics

The next plateau in system development will be added support for specific phenomena like indirect reference resolution and unknown words.

We first need to identify cases where these issues appear in the target RADAR domain. This includes emails as well as sentences collected above which don't meet the restrictions we placed on simple queries and statements. Specific strategies for adapting the system are still under development and will have to be refined once the basic system is in place.

7 Related Work

The work described here applies principles of Natural Language Understanding in a new architecture. We extend previous work on semantic distance to improve the quality of our analysis. We also apply our system to the task of semantic role labeling in a hybrid approach for that task, using empirical training along with human-engineered knowledge. These three contributions are grounded in traditional NLU techniques, but they explore contemporary problems and compete with systems on the forefront of current research.

7.1 Early NLU

(Carbonell and Hayes, 1992) gives an review of the work in natural language understanding (NLU) from an AI perspective. The authors identify four classes that cover the majority of NLU systems: pattern matching, semantic grammars, syntactically driven parsing, and case frame instantiation. These four technologies have appeared in NL interfaces for a number of applications, including database query, question-answering, story understanding, and information retrieval.

The most famous of the early pattern-matching systems is ELIZA (Weizenbaum, 1966), which played the role of a therapist in a human-computer dialogue. This approach is computationally the simplest of the four. Exact strings or regular expressions are matched against user input, and a lookup table gives the system response. Nonetheless, early pattern-matching systems fostered interest in the NLU problem in general. Other systems in this class include STUDENT (Bobrow, 1968) and PARRY (Parkison et al., 1977).

Semantic grammar systems use similar techniques to syntactically driven parsing. In both cases, a sentence is given some hierarchical structure, based on units that are identified in a set of grammar rules. In syntactic parsing, the units are usually based on phrase categories like noun, verb, and adjective. In contrast, semantic grammars recognize syntax and semantics at the same time. For example, a semantic grammar might have a category "parts of the body," which would recognize words like "leg," "arm," and "hand." In a syntactic grammar, these would all be considered nouns and would be indistinguishable from another noun like "table." This property makes semantic grammars good for domain-limited analysis systems. The first semantic grammars appeared in 1975 in the SOPHIE system (Brown and Burton, 1975). SOPHIE operated in the domain of electronic circuitry. The LADDER system (Sacerdoti, 1977) is another example from this class. It generated database queries in response to user questions about ships (e.g. "What is the beam of the Kennedy?"). The more recent NESPOLE! Machine Translation Project (Lavie et al., 2001) used a semantic grammar approach for both analysis and generation, in the domain of travel planning.

For general NLU, the narrowness of semantic grammars is a weakness, and pattern-matching is too naive for rich meaning representation. More general understanding systems have come from the last two classes mentioned earlier: systems using syntactically-driven parsing, and case-frame systems. Our work includes components from both.

Our own grammar-based system, mentioned in Section 1 belongs to the class of syntactically-driven NLU systems. These are systems which generate a syntactic parse first, then build a semantic representation using the result. They vary widely in the details of their syntactic analysis and in the semantic representation they use. SAD-SAM (Lindsay, 1963) was an early example of this approach. It used a limited English vocabulary and a context-free grammar. Semantic analysis operated on the completed parse tree for a test sentence. This separation of syntactic knowledge (in the rules that generate a parse tree) from semantic knowledge (in the rules that convert a parse tree to a meaning representation) is a property that makes syntactically-driven analyzers more general than semantic grammars. But it also means that the syntactic analyzer may waste time parsing sentences that will never be assigned a coherent meaning. "The man bit into the sandwich," for example, is a legitimate sentence. But it is syntactically similar to "The sandwich bit into the man," which makes much less sense. Semantic grammars, in fact, were developed in response to this overly-broad property of syntactic systems.

The most relevant systems to our work are those that try to combine the generality of syntactically-driven parsing with pruning power from a semantic representation. The RUS system (Bobrow, 1978) interleaved syntactic and semantic analysis to do just that. It used a framework based on Cascaded Augmented Transition Networks (CATNs). Attribute grammars (Knuth, 1968) and unification grammars in the tradition of (Kay, 1982) allow the analysis process to be driven by syntactic rules, but the grammars also incorporate semantic features. A description of the basic algorithm for unification-based parsing is given in (Norvig, 1992). In this way, the meaning of each syntactic constituent can be built incrementally using the meaning (i.e. the semantic features) of its sub-constituents. This compositional approach to semantic analysis has become a standard paradigm. It is also referred to as Montague semantics, described in (Dowty et al., 1981).

A number of systems that use this model have been developed, each with a unique research focus. PATR-II (Shieber, 1984) was an early system that established unification grammars as a feasible approach to natural language analysis. KBNL (Barnett et al., 1990) was a system for understanding English text that focused on the separation of syntactic from semantic knowledge sources and on broad-domain text analysis. AUTOSEM (Rosé, 2000) focused on robustness in the analysis of student essays. The system described in (Aleven et al., 2003) deals with student input in a tutoring session.

Our own grammar-based system, mentioned in Section 1, uses the same parser as these last two systems. But it implements a new layer of interaction with the semantics of Scone. This allows us to take advantage of the improved constraint-checking and inference mechanisms that are available in our knowledge base. It is groundwork for the other approaches we develop in the thesis, and it shares some drawbacks with the other systems from this class. Rules must be written manually, and for the general domain the tradeoff between a general grammar and an ambiguous one is difficult to manage by hand. In addition, these systems have little or no support for ranking the options available to them.

These problems are both addressed in our graph-based approach to semantic analysis, which falls roughly into the final class: case-frame systems. Case frames were introduced in (Fillmore, 1968). Frames, like Scone elements, represent concepts with associated roles that link them to other concepts. In the generic algorithm for frame-based analysis (Carbonell and Hayes, 1992), a set of frames is selected based on input words that trigger them. Next a recursive procedure attempts to connect the selected frames (or a subset of them) to each other. Frames can be connected by linking semantic roles with their fillers.

Because the basic semantic structures are similar, this recognition algorithm for frames is similar to the one we use for recognizing Scone structures. A critical difference is that case frames, in their original form, contain syntactic as well as semantic information. For example, the frame *Break*, as in "Susan broke the window," may indicate that the *Agent* case will appear in the SUBJECT position. Scone representations are purely semantic. Syntactic knowledge is stored in the Lexicon that associates Scone structures with strings. The argument for separating semantic from syntactic information was made in citebarnett-kb-nlp:1990. It addresses the problem we describe above for semantic grammars: the rules are too specific to scale up to a broad domain. The same argument can be made about the CD parser, below.

Roger Schank's Conceptual Dependency theory (CD) (Schank, 1975) defines a set of case frames for primitive actions like transferring property (e.g., *give*), changing position (*go*), speaking, eating, and others. Complex events are represented by combining these basic elements. The CD representation and the CD parser

(Schank and Tesler, 1969) have been used in a number of semantic analysis systems. MARGIE (Schank et al., 1973) was an understanding and inference system, and QUALM (Lehnert, 1977) was described as a question-answering system, although its task was closer to story understanding.

Other NLU systems that use case-frames include database query interpreters (McDermott, 1982), Information Retrieval systems [(Katz, 1988), (Lewis et al., 1989), (Mauldin, 1991)], and Machine Translation systems (Beale et al., 1995). The distinguishing feature of true case-frame systems is that semantic structures, rather than syntactic ones, drive the parsing process.

In the remainder of this section, we will present three modern systems that also leverage common-sense knowledge for NLP, and describe the differences in our work.

7.2 OntoSem

Ontological Semantics (OntoSem) is a semantic interpretation project at the University of Maryland. It is a continuation of an approach that started in the Mikrokosmos project (Beale et al., 1995), and is the most comparable system algorithmically to the work we propose here. In contrast to the systems described above, both of our approaches provide a ranking of candidate meanings based on a "cost" that comes from a knowledge base.

The semantic representation they use is based on frame-like structures called Text Meaning Representations (TMRs). The analysis routine they use involves two layers of constraint satisfaction. The control structure is based on the work of (Beale et al., 1996). Frames are selected in a similar manner to our own lexical routines: triggers from the input sentence cause a number of semantic structures to be activated. The subsequent procedures attempt to create a single TMR from these candidate pieces.

Ontological distance is used in the OntoSem project for word sense disambiguation and metonymy resolution. These are critical issues for NLU, and it is the aim of this work to apply better empirical techniques to these tasks and to ontology-based semantic analysis in general. The authors themselves allude to the need for this kind of work: "Supporting semantic analysis in this way [word sense disambiguation with corpus-based heuristic procedures] should become an important direction of work in corpus-oriented computational linguistics" (Nirenburg et al., 2003). We also see the opportunities for incorporating better corpus-based information in a knowledge-based (for us, Scone-based) semantic analyzer. The trained semantic distance approaches described in our proposal attack this important topic.

7.3 Cyc

The Cyc project was introduced in 1984 with the goal of collecting the first repository of common-sense knowledge that would be comprehensive enough for general reasoning. The knowledge base itself now stores close to two hundred thousand terms (Cycorp, 2004). The meaning representation they use is based on first-order predicate calculus. All the knowledge about a particular concept is represented as the set of logical predicates which have that concept as a participant.

In terms of natural language understanding, Cyc can play the role of an ontological knowledge source for knowledge-based approaches. CycL, the formalism used to store facts in Cyc, was in fact the target representation for the KBNL system (Barnett et al., 1990). In addition, researchers internal to the Cyc project have developed a natural language interface for CycL (Witbrock et al., 2003). This system is driven by a syntactic grammar, with semantic processing performed later.

In (Mahesh et al., 1996), the authors comment on the appropriateness of Cyc for knowledge-based NLP applications like ours. They concluded that Cyc was, at that time, not readily applicable to NLP for reasons that included "a lack of selectional constraints on relations between concepts" and the intractability of conducting path-based queries on the logical predicate representation. Scone, in contrast, has a rich representation for selectional constraints that can include pointers to other Scone concepts. The graph-based representation of Scone also supports path-based queries very well. Using the English interface we describe here, for example, one could get an answer to the question "What is the relationship between *bullet* and *gun*?" These type of open-ended queries are difficult or impossible to answer in the Cyc representation.

In addition, Cyc at the time of the critique we mentioned had little support for ranking candidate meanings when an interpretation was ambiguous. With adapted training data, some of the semantic distance techniques we propose here could plausibly be ported to Cyc or other knowledge bases. Although the details of the formalism may differ, positive results from our experiments could set an example for systems like Cyc.

7.4 Open Mind

Open Mind is another contemporary project whose goal is to acquire a truly general base of common-sense knowledge. Their focus has been on gathering this knowledge from untrained users (Singh et al., 2002). In their early collection efforts, they attempted to analyze free text entered by users on the Open Mind website (Mind, 2004). The data collected by this project is freely available and provides examples of human attempts to impart knowledge to a machine. Eventually the NLU for text from the world wide web proved uneconomical for the Open Mind effort. The team switched to analysis of input text via templates and web forms. As a result, the field is still open for automatic analysis of this data. We will use some of it for testing our system, particularly while the test dialogues from RADAR are still being determined.

8 Timeline

- March-May 2005

  Implement baseline Scone path search.

  Label training data for noun compound test.

  Evaluate accuracy on noun compounding task.

  February 28: CoNLL Shared Task Training Data Release

  Import frames to Scone

  Label training examples

  Address task-specific issues related to CoNLL data: format of output, adjusting training procedures, etc.

  Implement SVM-based system for comparison – using template from existing system (Moschitti, 2004)

  April 22: CoNLL Shared Task Paper Submission Deadline

  May 06: CoNLL Shared Task Test Data Release

  May 15: CoNLL Shared Task Results Deadline

- August-December 2005

  Write thesis section on noun compounds and semantic role labeling

  Conduct data collection for evaluating restricted-query coverage.

  Continue to investigate path search: non-baseline weighting schema.

  Implement control structures for phrase-level semantics.

  Evaluate on restricted queries and statements.

  Write thesis section on restricted queries and statements.

- January-May 2006

  Expand training data to include advanced topics: quantification, metonymy, unknown words

  Analyze data and plan the approaches for reference resolution in more detail.

  Baseline test on reference resolution and entity tracking.

Develop and test improvements to the baseline.

Evaluate performance on entity tracking in RADAR email.

- August-December 2006

  Continue work on advanced topics.

  Finish writing.

References

V. Aleven, K. R. Koedinger, and O. Popescu. 2003. A tutorial dialog system to support self-explanation: Evaluation and open questions. In *Proceedings of AI-ED the 11th International Conference on Artificial Intelligence in Education*, pages 39–46, Amsterdam.

Jim Barnett, Kevin Knight, Inderjeet Mani, and Elaine Rich. 1990. Knowledge and natural language processing. *Communications of the ACM*, 33(8).

S. Beale, S. Nirenburg, and K. Mahesh. 1995. Semantic analysis in the mikrokosmos machine translation project. In *Proceedings of the 2nd Symposium on Natural Language Processing*, pages 297–307, Bangkok, Thailand.

S. Beale, S. Nirenburg, and K. Mahesh. 1996. Hunter-gatherer: Three search techniques integrated for natural language semantics. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI96)*, Portland, Oregon.

S. Beale, M. McShane, and S. Nirenburg. 2004. Ontological semantic reference resolution: Setting the stage.

D. G. Bobrow. 1968. Natural language input for a computer problem-solving system. In *Semantic Information Processing*, pages 146–226. MIT Press, Cambridge, Mass.

R.J. Bobrow. 1978. The RUS system. Technical Report 3878, BBN, Cambridge, MA.

J.S. Brown and R.R. Burton. 1975. Multiple representations of knowledge for tutorial reasoning. In D.G. Bobrow and A. Collins, editors, *Representation and Understanding*, pages 311–349. Academic Press, New York.

A. Budanitsky and G. Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of NAACL-2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, June.

Jaime Carbonell and Philip J. Hayes. 1992. Natural language understanding. In Stuart Shapiro, editor, *the Encyclopedia of Artificial Intelligence*. John Wiley, New York.

Xavier Carreras and Lluís Màrques. 2004. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL-2004*, pages 89–97. Boston, MA, USA.

CMU-RADAR. 2004.

Cycorp. 2004. http://www.cyc.com.

D.R. Dowty, R. Wall, and S. Peters. 1981. *Introduction to Montague Semantics*. Reidel, Dordrecht.

S. E. Fahlman. 1979. *NETL: A System for Representing Real World Knowledge*. MIT Press, Cambridge, MA.

S. E. Fahlman. 2005. Scone user's manual.

James Fan, Ken Barker, and Bruce W. Porter. 2003. The knowledge required to interpret noun compounds. In *IJCAI 2003*, pages 1483–1485.

C. Fillmore. 1968. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–90. Holt, Rinehart, and Winston, New York.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28:3:245–288.

Kenneth Goodman and Sergei Nirenburg, editors. 1992. *KBMT-89: A Case Study in Knowledge-Based Machine Translation*. Morgan Kaufmann, San Mateo.

B.F. Green, A.K. Wolf, C. Chomsky, and K.Laughery. 1961. Baseball: An automatic question answerer. In *Proceedings of the Western Joint Computer Conference 19*, pages 219–224.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Boris Katz. 1988. Using english for indexing and retrieving. In *1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*.

M. Kay. 1982. Parsing in functional unification grammar. In D.R. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, pages 251–278. Cambridge University Press, New York.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX, July.

Donald E. Knuth. 1968. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145.

A. Lavie, Langley C., Waibel A., Lazzari G., Pianesi F., Coletti P., Balducci F., and Taddei L. 2001. Architecture and design considerations in nespole!: a speech translation system for e-commerce applications. In *Proceedings of HLT 2001 Human Language Technology Conference*, San Diego, California, March 18–21.

Wendy Lehnert. 1977. A conceptual theory of question answering. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 651–657.

D. D. Lewis, W. B. Croft, and N. Bhandaru. 1989. Language-oriented information retrieval. *International Journal of Intelligent Systems*, 4(3), Fall.

Robert K. Lindsay. 1963. Inferential memory as the basis of machines which understand natural language. In E. Feigenbaum and J. Feldman, editors, *Computers and Thought*. MIT Press, Cambridge, MA.

K. Mahesh, S. Nirenburg, J. Cowie, and D. Farwell. 1996. An assessment of cyc for natural language processing. Technical Report Memoranda in Computer and Cognitive Sciences MCCS-96-296, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

Michael L. Mauldin. 1991. Retrieval performance in FERRET, a conceptual information retrieval system. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–355, Chicago, Illinois.

J. McDermott. 1982. XSEL: A computer salesperson's assistant. In J. Hayes, D. Michie, and Y-H. Pao, editors, *Machine Intelligence*, pages 325–337. Ellis Horwood and Wiley, New York.

David B. McDonald. 1982. Understanding noun compounds. Technical Report CMU-CS-82-102, Department of Computer Science, Carnegie Mellon University.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. Annotating Noun Argument Structure for NomBank. In *Proceedings of LREC-2004*, Lisbon, Portugal.

Open Mind. 2004.

Alessandro Moschitti. 2004. A study on convolution kernel for shallow semantic parsing. In *proceedings of the 42nd Conference of the Association for Computational Linguistics (ACL-2004)*, Barcelona, Spain.

Erik T Mueller. 1998. *Natural Language Processing with ThoughtTreasure*. Signiform, New York.

Sergei Nirenburg, Marjorie McShane, and Stephen Beale. 2003. Operative strategies in ontological semantics. In *Proceedings of HLT-NAACL-03 Workshop on Text Meaning*, June.

Peter Norvig. 1992. *Paradigms of Artificial Intelligence*. Morgan Kaufmann Publishers, San Francisco, CA.

Eric Nyberg and Teruko Mitamura. 1992. The kant system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of COLING-92*.

Eric Nyberg and Teruko Mitamura. 1995. Controlled english for knowledge-based mt: Experience with the kant system. In *Proceedings of TMI-95*.

Eric Nyberg and Teruko Mitamura. 2000. The KANTOO machine translation environment. In *Proceedings of AMTA 2000*.

Boyan Onyshkevych. 1997. An ontological-semantic framework for text analysis. Technical Report CMU-LTI-97-148, Language Technologies Institute of Carnegie Mellon University.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2003. The proposition bank: An annotated corpus of semantic roles. *submitted to Computational Linguistics*, December.

R.C. Parkison, K.M. Colby, and W.S. Faught. 1977. Conversational language comprehension using integrated pattern-matching and parsing. *Artificial Intelligence*, pages 111–134.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

Philip Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal.

C.P. Rosé and A. Lavie. 2001. Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In J. C. Junqua and G. Van Noord, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Press.

C.P. Rosé. 2000. A framework for robust semantic interpretation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

E.D. Sacerdoti. 1977. Language access to distributed data with error recovery. In *Proceedings of the Fifth International Joint Conference on Artificail Intelligence*, pages 196–202, Cambridge, MA.

Roger Schank and Larry Tesler. 1969. A conceptual dependency parser for natural language. In *International Conference on Computational Linguistics (COLING)*.

R. C. Schank, N. Goldman, C. Reiger, and C. Riesbeck. 1973. Margie: Memory, analysis, response generation, and inference in english. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 255–261, SRI, Menlo Park, California.

Roger C. Schank. 1975. *Conceptual Information Processing*. North-Holland Publishing Co., Amsterdam.

S.M. Shieber. 1984. The design of a computer language for linguistic information. In *Proceedings of COLING*, pages 362–366.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In Robert Meersman and Zahir Tari, editors, *Lecture Notes in Computer Science: Vol. 2519*. Springer-Verlag, Heidelberg.

J. Weizenbaum. 1966. Eliza – a computer program for the study of natural language communication between man and machine. *CACM*, 9(1):36–45, January.

M. Witbrock, D. Baxter, J. Curtis, D. Schneider, R. Kahlert, P. Miraglia, G. Matthews, and A. Vizedom. 2003. An interactive dialogue system for knowledge acquisition in cyc. In *IJCAI-03*, pages 138–145, Acapulco, Mexico, August.

W. Woods. 1973. Progress in natural language understanding - an application to lunar geology. In *Proceedings of AFIPS*, volume 42, pages 441–450.