PARC's Bridge and Question Answering System

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen

PARC

Proceedings of the GEAF 2007 Workshop

Tracy Holloway King and Emily M. Bender (Editors)

CSLI Studies in Computational Linguistics ONLINE

Ann Copestake (Series Editor)

2007

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

This paper describes the Bridge system, a system designed to robustly map from natural language sentences to abstract knowledge representations. The system runs on PARC's parser, generator, and ordered rewrite platform XLE. The Bridge system has been extended to include a type of light inference, based on an entailment and contradiction detection algorithm which also runs on XLE. The paper then describes a search and question answering application, Asker, which uses the Bridge system to create a semantic index of text passages and which allows a user to query the index in natural language.

1 Introduction

Bridge is a PARC system that robustly maps natural language sentences into a logical abstract knowledge representation language (AKR). Using this mapping, we have built an application, Asker, that supports high-precision question-answering of natural language queries from large document collections (e.g., the Wikipedia, newswire, financial reports). For example, if a collection includes the sentence *The reporter failed to discover that three young men were killed in the attack on Ryad*., then the system could answer the query *Did anyone die in the attack on Ryad?* with YES (perhaps indicating who died) and highlight the phrase in the document in the collection that contains this information.

The basic system components and their connection is shown in the diagrams in Figures 1–4. Natural language text is mapped into a first level of abstract knowledge representation (AKR0) (see section 2). Text passages are then passed through an expansion step to produce a representation with addtional inferrable facts (P-AKR). In contrast, queries are passed through a simplification step to produce a representation with fewer facts (Q-AKR), a smaller kernel from which the rest can be inferred. Asker uses the expanded passage to compute index terms that capture semantic roles in the representation (section 4.1). To retrieve potential answer passages from the collection, index terms from the query representation identify stored texts with corresponding semantic structure (section 4.2); as a backoff, texts are retrieved that share expanded, normalized keywords with the query. Entailment and contradiction detection (ECD) can be performed to determine subsumption relations between the passage and question and hence provide an answer (section 3). ECD can be used separately to check whether a given passage text entails or contradicts a given query/hypothesis text.

[†]This work was sponsored in part by DTO. Approved for Public Release; distribution unlimited. We thank the audience of GEAF for providing extensive feedback on the QA demo of the system. We also thank all of the people who have worked on the system over time. Ron Kaplan and Dick Crouch were central members of the team, and helped define the framework of the Bridge/Asker system. Dick Crouch was a major designer and implementor of key components. John T. Maxwell III is a major designer and implementor of the core XLE system. We also want to thank the interns and postdocs who contributed: Tina Bögel, Hannah Copperman, Liz Coppock, Olya Gurevich, Anubha Kothari, Xiaofei Lu, Johannes Neubarth, Matt Paden, Karl Pichotta, and Kiyoko Uchiyama.

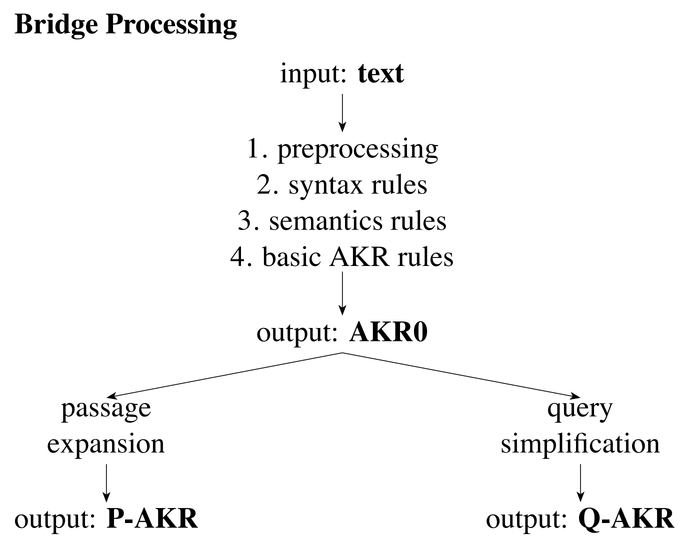

Figure 1: Syntactic Lexical-Function Grammar (LFG) rules and semantic and KR-specific ordered rewrite rules produce a basic knowledge representation for passage and query texts. Passages expand inferences based on linguistic properties. Queries are simplified to their core meaning to remove unnecessary structure.

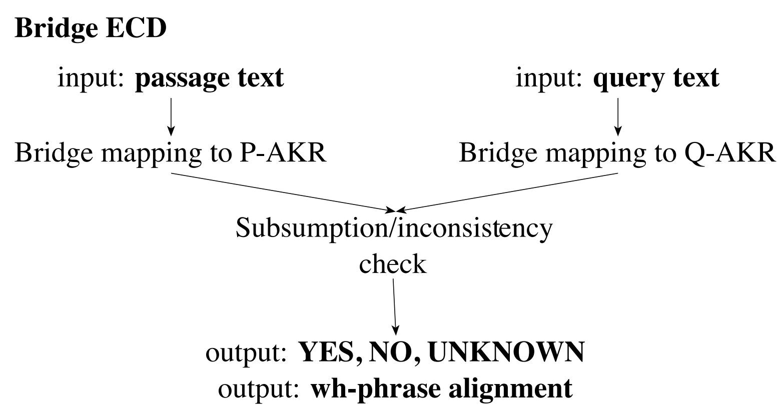

Figure 2: Expanded passage representations are compared using subsumption with simplified query representations to determine if the passage entails the query.

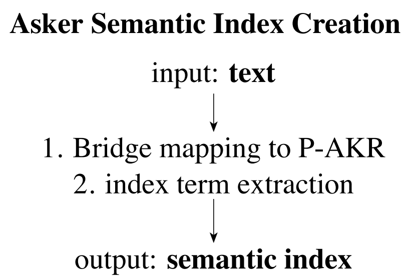

Figure 3: Index terms for each passage reflect the semantic roles of terms in a sentence.

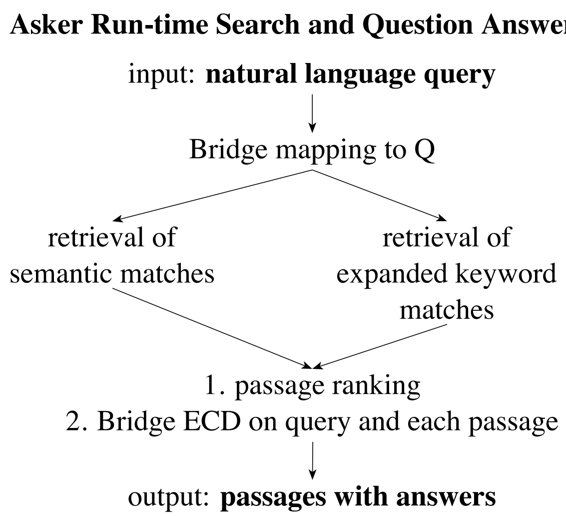

Figure 4: Use of index terms in the query supports more precise retrieval of relevant sentences. Keywords, expanded with WordNet synonym sets (synsets) and hypernyms, provide a backoff for recall.

The mapping from syntactic structures to (linguistic) semantics and then abstract knowledge representations (AKR) runs on the XLE platform (Maxwell and Kaplan, 1996; Crouch et al., 2007) and is described in Crouch and King (2006) and Crouch (2005). The logic of the representations has been described in Bobrow et al. (2005) and de Paiva et al. (2007). The linguistic, semantic rationale for the use of concepts in AKR was originally described in Condoravdi et al. (2001, 2003). Components of the system have been described in Crouch and King (2005), Gurevich et al. (2005), and Nairn et al. (2006). An earlier application to a collection of copier repair tips written by Xerox technicians is described in Crouch et al. (2002) and Everett et al. (2002). The more recent application to question-answering in the framework of the PASCAL-organized[1] competition Recognizing Textual Entailment (RTE) is described in Bobrow et al. (2007).

In this paper, we first describe the AKR language that our system uses (section 2). AKR is designed to meet two constraints that are somewhat in tension: a natural representation of language constructs on the one hand and a straightforward computation of direct inferential relations between two texts on the other. Our entailment and contradiction detection algorithm (section 3) implements this inference procedure between two possibly ambiguous texts without the need for disambiguation. Finally, we discuss the structure of the Asker repository which indexes sentences on the basis of their AKR representation in a large scale database (over 10^6 documents) and allows real-time semantic retrieval from this index (section 4).

2 Abstract Knowledge Representation (AKR)

We start our discussion of AKR representations with the sentence *John Smith discovered that three men died*. The full AKR is as in (1).

(1) **Conceptual Structure:**

```
subconcept(discover:2, [detect-1, ..., identify-5])
role(Theme, discover:2, ctx(die:5))
role(Agent, discover:2, Smith:1)
subconcept(Smith:1, [male-2])
alias(Smith:1, [John, Smith, John_Smith])
role(cardinality_restriction, Smith:1, sg)
subconcept(die:5, [die-1, die-2, ..., die-11])
role(Theme, die:5, man:4)
subconcept(man:4, [man-1, ..., world-8])
role(cardinality_restriction, man:4, 3)
```

Contextual Structure:

```
context(t)
context(ctx(die:5))
top_context(t)
context_lifting_relation(veridical, t, ctx(die:5))
context_relation(t, ctx(die:5), crel(Theme, discover:2))
instantiable(Smith:1, t)
instantiable(discover:2, t)
instantiable(die:5, ctx(die:5))
instantiable(man:4, ctx(die:5))
```

Temporal Structure:

```
temporalRel(startsAfterEndingOf, Now, discover:2)
temporalRel(startsAfterEndingOf, Now, die:5)
```

[1] See the PASCAL website: www.pascal-netword.org

The representation for this sentence has two contexts: the top context t, specifying what the author of the sentence is committed to as the true state of the world by virtue of uttering the sentence; and ctx(die:5), specifying what was discovered by John Smith, which is the proposition that three men died.

The verb *discover* carries a presupposition that what is described as being discovered is true according to the author of the sentence; that is, one can only be said to discover true facts. This is part of lexical knowledge and is captured in this example by context_lifting_relation(veridical, t, ctx(die:5)). Because of this veridical relation, in the expansion to P-AKR, the clauses:

(2) instantiable(die:5, t)
instantiable(man:4, t)

are added to the contextual structure. These instantiability statements capture existence commitments in our representation. As a result, the system will answer YES to the passage-query pair *John discovered that three men died. Did three men die?* In the top context t, we also have the instatiability claims:

(3) instantiable(Smith:1, t)
instantiable(discover:2, t)

Within the context of what was discovered by John Smith we have two concepts, the dying event die:5, and the concept man:4. For each of these, the representation has a subconcept expression. These expressions encode WordNet's representation of the verb *die* (a list of 11 synsets, corresponding to the 11 verb senses for *die* differentiated in WordNet) and the noun *man* (a list of 8 synsets):

(4) subconcept(die:5, [die-1, die-2, die-3, ..., die-11])
subconcept(man:4, [man-1, serviceman-1, ..., world-8])

We are using WordNet (Fellbaum, 1998) as a surrogate for the taxonomic part of an ontology because it is the largest available resource for mapping English words into an (approximate) abstraction hierarchy through WordNet's hypernyms. We have patched WordNet in places where omissions and extra entries became problems for the system. Since VerbNet, whose use is described below, links to WordNet, we have also made these two resources more consistent.

To capture the fact that the number of dying men is three, the representation includes a cardinality restriction on the concept man:4.[2] The dying event is related to its undergoer participants via role(Theme, die:5, man:4). In the top context we have two more concepts, the concept for John Smith and the discovering event discover:2, a subconcept of WordNet's synsets for the verb *discover*.

[2]Our representations deal with quantifiers in general through a combination of instantiability statements, contexts and cardinality restriction clauses.

While WordNet knows about some words used as names,[3] it does not list every man named John in history, nor does it list every masculine name. The English morphology associated with the system's syntactic grammar knows that *John* is a man's name, and the semantics uses this information to create a subconcept structure based on WordNet: subconcept(Smith:1, [male-2]). The name itself is captured in an alias fact. Incorporated into the system is a theory of when two aliases can refer to the same individual. So *John Smith* can be mentioned later as *John*, *Smith*, or *John Smith*. These three possiblities are included in the alias fact. Given a passage-query pair like *John Smith arrived and John Bowler left. Did Bowler leave?* the system will answer YES. Moreover, to the passage-query pair *John Smith arrived and John Bowler left. Did John leave?* the system will answer YES: [John Bowler], since at least one of the people named *John* in the passage did leave.

Finally, the concept discover:2 is restricted to have Smith:1 as its agent role (role(Agent, discover:2, Smith:1)) and the context specifying what John discovered as its theme role (role(Theme, discover:2, ctx(die:5))).

The temporal relations capture the relative time ordering of the events described with respect to the time of utterance or writing of the sentence. Now (the time of utterance) is after the discovering, and the dying, as represented by:

(5) temporalRel(startsAfterEndingOf, Now, discover:2)
temporalRel(startsAfterEndingOf, Now, die:5)

As indicated by this example, AKR representations can express the content of beliefs, possible states of the world, counterfactuals, etc.

2.1 Existence and Restrictions

Terms like die:5 and man:4 do not refer to individuals, but to concepts (or types). When the AKR makes reference to a subconcept man:4 of the kind [man-1, serviceman-1, man-3, ..., world-8] restricted to be a kind of man that died, the AKR does not make a commitment that there are any instances of this subconcept in the world being described by the author of a sentence. For example, the sentence *John imagined that three men died.*, has in the AKR an embedded context representing what is being imagined. Because this embedded context is not veridical with respect to the top context, there is no commitment (by the author or in the representation) about there actually being any dead men.

The instantiable assertions represent the existence of the kinds of objects described. In the top-level context t, there is a commitment to an instance of a male individual with the name *John Smith* and of a discover event discover:2 made by him. While the three men and the dying event occur in the context of what was discovered by John Smith, they become instantiable at the top context because *discover* with a *that* complement is marked as a factive verb (Nairn et al., 2006).

[3] For example, WordNet has synsets for the evangelist John and the English King John who signed the Magna Carta. There are also entries for the common noun *john*.

Compared to traditional first order logic with complex quantifiers, AKR separates the descriptions of types of events and objects (in the conceptual block) from the commitments to existence (in the contextual block). The conceptual block includes subconcept assertions, role restrictions and cardinality constraints. The contextual block includes (un)instantiability of these concepts in contexts, and relations between contexts, including context-lifting rules similar in spirit to those in McCarthy's context logic (McCarthy, 1993). The use of contexts to capture a collection of statements true in a context and the use of contexts as arguments (reifying the collection of statements) makes AKR technically not first order, but the reasoning in the system preserves many first order properties locally.

2.2 Lexical resources

Mapping to AKR and textual inference depend crucially on words and ontological relations between the concepts they map to. We have integrated a number of existing lexical resources into a Unified Lexicon (UL) (Crouch and King, 2005), adding new annotations to classes of words to support desired inferences. The basic size of the UL is shown in (6).

(6) **Unified Lexicon: Part of Speech of Entries**

POS	Number of Entries
verbs	42,675
nouns	14,293
adjectives	8,537
deverbal adjectives	1,291
adverbs	13

Note that many words have no UL entry because their behavior in the mapping to AKR is predictable from their syntactic structure (e.g., most nouns, adjectives, and adverbs). In addition, adjectives and nouns that are predictably derived from verbs (e.g, *the hopping frog*, *the defeated champion*, *the writing of the book*) do not need entries in the UL to trigger the appropriate mapping rules.

2.2.1 Basic Concept and Role Lookup

The mapping rules and the UL use WordNet synsets and hypernyms. The system maps the words recognized by WordNet into the associated synsets directly via the WordNet API; a copy of WordNet is not included in the UL. Words not in WordNet are mapped to, generally singleton, synsets based on information from the XLE morphology and syntax (e.g., the treatment of person names discussed above). Initially all synsets for a given word are retrieved; this list is then trimmed to a subset of the WordNet concepts if additional information is available, for example from VerbNet or from the context of the text. Noun-noun compounds (e.g., *theme park*) and adjective-noun compounds (e.g., *high school*) known to WordNet are assigned the appropriate WordNet synsets.

VerbNet (Kipper et al., 2000) is used to map from syntactic predicate-argument structures to event structures with named roles, occasionally simplified by collapsing certain role distinctions. These role and event structures have been heuristically augmented to cover all of the verb-subcategorization frame pairs in the XLE syntactic lexicon (e.g., the role assignments from verbs known to VerbNet can be used to provide roles for other verbs in their WordNet synset with the same subcategorization frames). This results in significant expansion of the coverage of VerbNet: of the ∼42,000 verb entries in the UL, ∼25,000 are not directly from VerbNet. Examples of the VerbNet roles can be seen in the AKRs in examples such as (1).

2.2.2 Lexical Marking for Rule Triggering

In addition to these basic resources, the UL incorporates information about lexical items that is needed to trigger mapping rules that affect the contextual facts, especially those involving relations between contexts (Nairn et al., 2006). These lexical classes are shown in (7).

(7) **Unified Lexicon: Lexical Marking for Rule Triggering**

Lexical Class	Number	Example
factives	230	John discovered that Mary left.
implicatives	192	John managed to leave.
propositional attitude	762	John abhors that Mary left.
neutral	33	John sought a unicorn.[4]
temporal relation	721	John longs to leave.
temporal: forward shift	301	John authorized Mary to leave.
temporal: simultaneous	70	John attempted to leave.
sentential adverbs	13	Obviously John left.

For example, the factivity of the verb *discover* when used with a *that* complement is marked. This marking indicates that *discover*'s Theme context argument is veridical with respect to its immediately higher context, enabling the lifting of instantiability from the lower context to the higher one, as described in (1).

2.2.3 Lexical Marking for Normalization

Lexical resources are also used in the normalization of representations. Relevant lexical classes are shown in (8). A canonical example of this type of normalization is the mapping of eventive nominal expressions into equivalent verbal counterparts (e.g., *Rome's destruction of Carthage* is mapped to the same representation as *Rome destroyed Carthage*.) (Gurevich et al., 2005). The UL contains related noun-verb

[4]This marking is meant for intensional predicates with respect to an argument position, distinguishing between *seek* and *find*, for instance. It results in having no instantiability assertion, capturing an existential commitment, for the term corresponding to the relevant argument of the predicate (in the case of *seek* the direct object). By default there is an instantiability assertion for every argument of a predicate in the context of predication.

pairs which are used by the rules to map nouns and their associated phrases into their verbal, eventive counterparts with appropriate arguments. These entries not only include the pairings (e.g. *destruction-destroy*, *employer-employ*) but also classification information. Some of this information involves the mapping of arguments; for example, agentive nominals like *employer* refer to the agent of the event, while *-ee* nominals like *employee* refer to the patient. Other information involves the degree of lexicalization; this determines whether the mapping to the eventive representation is obligatory or optional. These rules, in conjunction with the lexical class information, capture ambiguity in the language; for example, *Rome's destruction* can mean either that Rome is the patient of the destroying event or the agent.

(8) **Unified Lexicon: Lexical Marking for Normalization**

Lexical Class	Number	Example
deverbal nouns	5,484	Rome's destruction of Carthage
become adjective	51	The child sickened.
become more adjective	121	John darkened the room.
pertainyms	289	Japanese children
conditional verb	29	John wasted the chance to leave.
ability nouns	11	John had the choice to leave.
asset nouns	15	John had the money to leave.
bravery nouns	16	John had the strength to leave.
chance nouns	19	John had the chance to leave.
effort nouns	13	John took the trouble to leave.
certainty adjectives	3	John is sure to leave.
consider verb	4	John considered the boy foolish.

The mapping of texts to AKR involves changes of representation to aid inference. Among these are the representation of linguistic paraphrases and idioms which fall into classes that are lexicalized appropriately. For example, the "become adjective" verbs like *redden* are rewritten to an AKR similar to that of *become red*. Phrases such as *take a turn for the worse* are mapped to the representation for *worsen*. An additional, related large class of items are light verbs such as *take*, where the meaning of the verb phrase depends on an argument of the verb. Some examples of light verb use include *take a flight* and *use a hammer* that can be transformed into *fly* and *hammer*. Some verbs are marked as conditionally implicative because they form implicative constructions with a particular class of nouns. For example, *have the foresight to X* is semantically the same type as *manage to X*. As the best representation for the output of these rules is still being explored, there are only a few lexicalizations for each class currently implemented.

As mentioned above, many noun-noun compounds are known to WordNet and hence are given the appropriate WordNet synset. However, many such compounds, especially the less-lexicalized ones, are not in WordNet. The AKR mapping rules define noun-noun relations based on the meaning of the head noun and the meaning of its modifier, where the meanings are (upper level) WordNet synsets. For example, a food solid modifying a type of tableware (e.g, *meat plate*) creates a *for* relation. These rules allow multiple mappings to reflect the multiple readings of many noun-noun compounds (e.g., *a wood box* can mean either a box made of wood or a box for holding wood).

Not all normalization is triggered by lexical classes that are encoded in the UL: the structure of the representations is often sufficient to determine how to map them into AKR. Our general approach is to capture the similar content of alternative linguistic expressions by normalizing their AKR to a common representation. This normalization occurs at many levels. For example, the syntax abstracts away from word order and localizes dependencies (e.g. in *John wants to leave.*, John is localized as the subject of both *want* and *leave*), the semantics canonicalizes passives to actives (*The cake was eaten by John.* becomes *John ate the cake.*)[5] and negative quantifiers on subjects (*No boy left.* introduces a sentential negation similar to *not*). Lexically-based inferences provide further information. One significant type of such inferences is associated with verbs of change, such verbs of change of location (e.g. from *John left Athens.* one concludes that John was in Athens before the departure and was not there at least for a while afterwards). The information about pre- and post-conditions of events described by verbs of change such as *leave* is productively extracted from the VerbNet event structure into the UL and then used by the mapping rules.

2.2.4 Lexical Marking for Expansion of Representation

Some mappings expand the representation instead of, or in addition to, normalizing it. Most of these mappings expand just the passages and not the queries. Sample lexical classes of this type are shown in (9).

(9) **Unified Lexicon: Lexical Marking for Expansion of Representation**

Lexical Class	Number	Example
lethal cause verbs	29	John strangled his victim.
symmetric nouns	2	John is Mary's partner.

Such expansions are sometimes specific enough that they are done exclusively in the rules and are not currently in the UL. For example, in a text, *buy* is inferred from *sell*, with appropriate role substitutions, and vice versa. As a result, a query about a buying event can match against a passage described in the text as a selling event. These are done as relatively constrained lexical classes in order to correctly map the arguments of one event to those of the other (e.g. *win-lose* maps its surface arguments differently from *buy-sell*).[6] Family relations such as *husband-wife*

[5] The choice to have the active-passive correspondence dealt with in the mapping component rather than the UL reduces the size of the UL. The active-passive correspondence could, alternatively, be encoded in the UL by matching every transitive verb entry with an entry for its passive counterpart, thus substantially increasing the size of the UL.

[6] With appropriate, more complex lexical markings, such correspondences could be encoded in the UL. Mapping rules would then be used to generate terms and role restrictions for the member of the pair not explicit in the input sentence.

are also expanded in the passages to allow them to match with queries using the converse relation.

A related aspect of our approach is to make information in the structure of certain phrases explicit. For example, date expressions (e.g., *May 1, 2007*) and location expressions (e.g., *Boise, Idaho*) are decomposed into subfacts that allow basic inferencing in conjunction with the rest of the representation. For example, making explicit that Boise is in Idaho, not just part of the name of the place, makes it possible to conclude from the fact that John lives in Boise, Idaho, that John lives in Idaho.

As seen by the wide range of examples in this section, lexical resources are a vital component of the Bridge system. The system incorporates existing resources, such as VerbNet, as well as resources created especially for the system. Each set of resources is used by the AKR mapping rules to create appropriate representations of natural language texts. The efficacy of these resources and their implementation is demonstrated by the ability of the system to use the resulting representations in applications such as the Asker search and question answering system.

2.3 Ambiguity Management

A hallmark of our computational approach to syntax, semantics, and knowledge mapping has been the ability to manage ambiguity by combining alternative interpretations into a single packed structure that can be further processed without the typically exponential cost of unpacking (Maxwell and Kaplan, 1991). For the traditional example of *John saw a girl with a telescope*, the packed representation compactly represents two interpretations: one where the seeing was done with a telescope and the alternative where the girl was carrying a telescope. In the packed representation, the common elements of both interpretations are represented only once, and only the alternative connections need to be expressed. The packed AKR representation is shown in (10). The alternate connections are shown in the lines labeled A1 and A2.

(10) **Choice Space:**
```
  xor(A1, A2) iff 1
```
Conceptual Structure:
```
  subconcept(see:2, [see-1, ..., interpret-1])
A1: role(prep(with), see:2, telescope:9)
  role(Stimulus, see:2, girl:6)
  role(Experiencer, see:2, John:1)
  subconcept(John:1, [male-2])
  alias(John:1, [John])
  role(cardinality_restriction, John:1, sg)
  subconcept(girl:6, [girl-1, ..., girl-5])
A2: role(prep(with), girl:6, telescope:9)
```

```
role(cardinality_restriction, girl:6, sg)
subconcept(telescope:9, [telescope-1])
role(cardinality_restriction, telescope:9, sg)
```

Contextual Structure:

```
context(t)
top_context(t)
instantiable(John:1, t)
instantiable(girl:6, t)
instantiable(see:2, t)
instantiable(telescope:9, t)
```

Temporal Structure:

```
temporalRel(startsAfterEndingOf, Now, see:2)
```

The two distinct readings are labeled by A1 and A2, which are a disjoint partition of the top level choice 1 (`xor(A1, A2) iff 1`). In reading A1, the seeing concept is further restricted to be a seeing with a telescope, whereas in A2, the girl is restricted to be a girl with a telescope.

The mapping from text to AKR via the syntactic and semantic representations and the entailment and contradiction detection take advantage of the same ambiguity management system, thereby gaining full efficiency by never unpacking.

Each level of representation provides possible sources of additional ambiguity. Sometimes it is useful to choose a subset of the interpretations for efficiency reasons or to interface with non-ambiguity-enabled modules and applications. Stochastic models are used to order the interpretations by probability in the XLE system (Riezler et al., 2002). In addition, rule-based optimality marks allow low probability interpretations through only if there is no more optimal interpretation available (Frank et al., 2001). This mechanism is used, for example, to apply VerbNet's sortal restrictions on roles so that the subconcept associated with a verb's arguments can be further constrained, thereby increasing precision and decreasing ambiguity. The optimality mechanism treats these sortal restrictions as soft constraints. If in an ambiguous, packed representation one solution satisfies the sortal restrictions and one does not, only the one that satisfies them appears in the final representation. However, if all the solutions violate the sortal restrictions, the ones which violate the fewest restrictions are used. The combination of efficient processing of packed ambiguous structures with stochastic and rule-based methods for selecting among these representations supports practical, robust analysis of natural language texts.

3 Entailment and Contradiction Detection (ECD)

So far we have described how the Bridge system produces AKR logical forms. These are used for light reasoning, that we call entailment and contradiction detection. It follows the form of the "textual inference" challenge problems that have been part of the PASCAL initiative. The task of the challenge is: given two sentences, P (for passage or premise) and Q (for query or question), determine whether P provides an intuitive answer for Q as judged by a competent user of the language without any special knowledge. Thus the goal is to decide whether Q follows from P plus some background knowledge, according to the intuitions of an intelligent human reader. This decision is supposed to be based simply on the language involved, factoring out world knowledge, but this distinction is difficult to characterize precisely and has become the topic of much current research.

We have developed a collection of algorithms for efficiently detecting entailment and contradiction relations holding between AKRs for queries and AKRs for candidate answer texts. We have taken a very strict approach, not including plausible inferences. Thus we deal only with a circumscribed set of textual inferences, but ones that must be handled by any system aiming for the larger task. Our approach is to expand the passage texts by using the linguistic inference patterns described earlier. The system tests entailment and contradiction through a subsumption process described below. Some special case reasoners support identification of named objects, comparison of specificity of WordNet synsets, and compatibility of cardinality restrictions. We call our strict form of textual inference "precision-focused textual inference"; our approach and results are described in Bobrow et al. (2007).

As a simple example consider how we conclude from *John saw a happy girl.* that *A child was seen.* The representations are shown in (11) and (12) respectively.

(11) John saw a happy girl.
Conceptual Structure:
subconcept(happy:12, [happy-1, felicitous-2, glad-2, happy-4])
subconcept(see:6, [see-1, understand-2, witness-2, , see-23])
role(Stimulus, see:6, girl:18)
role(Experiencer, see:6, John:1)
subconcept(John:1, [male-2])
alias(John:1, [John])
role(cardinality_restriction, John:1, sg)
subconcept(girl:18, [girl-1, female_child-1, ... girl-5])
role(cardinality_restriction, girl:18, sg)
role(subsective, girl:18, happy:12)
Contextual Structure:
context(t)
top_context(t)
instantiable(John:1, t)
instantiable(girl:18, t)
instantiable(see:6, t)
Temporal Structure:
temporalRel(startsAfterEndingOf, Now, see:6)

(12) A child was seen.
Conceptual Structure:
subconcept(see:13, [see-1, understand-2, witness-2, ... see-23])

role(Stimulus, see:13, child:3)
subconcept(child:3, [child-1, child-2, ... child-4])
role(cardinality_restriction, child:3, sg)
Contextual Structure:
context(t)
top_context(t)
instantiable(see:13, t)
instantiable(child:3, t)
Temporal Structure:
temporalRel(startsAfterEndingOf, Now, see:13)

ECD works on texts that have been analyzed into AKRs. Passage AKRs are expanded to encode linguistically based inferences (none in (11)). The AKR for concept and context denoting terms are aligned across the passage and question representations, and rules defining a calculus of entailment and contradiction are applied.

Before determining specificity relations between terms in the premise and conclusion AKRs, it is necessary to align these terms: alignments are not always obvious. They are computed by a heuristic algorithm that considers all plausible alignments where there is sufficient conceptual overlap between terms. This may result in multiple possible alignments with different likelihood scores. Term alignments with wh-terms (*who*, *what*, *where*, etc.) provide the answers to wh-questions when an entailment is detected. In the above example, the two seeing events are aligned, as are the skolems for girl:18 and child:3.

We check each possible term alignment to see if there is an entailment or contradiction between the premise and conclusion representations. The technique detects an entailment or contradiction if any interpretation of a premise entails or contradicts any interpretation of the conclusion.

The detection mechanism is implemented using XLE's packed rewrite system. The core idea behind using the rewrite system is that if the premise representation entails part of the conclusion representation, then that part of the conclusion can be deleted (i.e. rewritten to nil). A conclusion is entailed if all of its component parts have been removed. Hence, if there is a choice in which all of the conclusion representation has been removed, then there is some interpretation of the premise and the conclusion under which the premise entails the conclusion. Contradictions are detected via rules that add a contradiction flag whenever there is a choice of premise and conclusion interpretations such that parts of the representations conflict.

As a preliminary to deleting entailed conclusion facts or flagging contradictions, rules are first applied to make explicit the subsumption and specificity relations holding between concept terms in the premise and conclusion.

The next set of rules explores the consequences of these specificity relations on instantiability claims. For an upward monotone environment, instantiability of a specific concept entails instantiability of a more general concept and uninstantiability of a general concept entails uninstantiability of a more specific concept. For downward monotone environments, the relations are reversed. This captures the pattern that if *A little girl hopped*. then we know that *A girl hopped*., since *little girl* is more specific than *girl*. From *Girls hopped*. we cannot infer that *Little girls hopped*., as it is possible that all the hopping girls are big girls, but from *All girls hopped*. we can infer that *All little girls hopped*., as the quantifier creates a specificity reversing situation.

To return to our example, it is determined from WordNet that a girl is a kind of child. A happy girl (girl with the role subsective happy) is yet more specific. Hence the seeing event in the passage is more specific than that in the hypothesis and hence instantiable(see:6, t) entails instantiable(see:13, t). Instantiability statements in t are existence statements, and the existence of an instance of a more specific concept implies the existence of its generalizations (if there is a happy girl, there is a girl, which means there is a child, and similarly for *see*).

The ECD algorithm separates the task of structure alignment from the task of detecting logical relations between the representations. This separation makes the method more robust than many graph alignment and matching approaches (Braz et al., 2006) and is applicable to packed representations without full graph matching. This implements a verifiable calculus of entailment and contradiction, which in theory corresponds (closely) to Natural Logic entailment (van Benthem, 1986; MacCartney and Manning, 2007). The differences reside in the introductions of contexts and of packed representations. We believe that the ECD algorithm combines the best of the inference-based and graph-matching approaches. Term alignment is robust to variations in the input structures and the absence of precisely formulated axioms. The entailment calculus rules can be sensitive to non-local aspects of structure and thus deal with more global constraints on entailment or contradiction. In addition, since the approach is ambiguity-enabled, the system can detect whether any one of the possible interpretations of the putative answer answers any one of the possible interpretations of the question.

Given this ability to determine entailment and contradiction between a passage and a query, the Asker system builds up a semantic index of AKRs for passages and then at run-time produces AKRs for queries. These query AKRs are used to retrieve possible answer passages and then ECD can be applied to provide answers to the original query. This process is described in the next section.

4 Indexing and Retrieval

Asker is a search and question answering system. In order to retrieve relevant passages and documents from a large corpus, a specialized search index is constructed that encodes the information from the AKR for each sentence. Typical keyword search indices map words (or their stems) to their document occurrences, along with word offset information and other metadata. The Asker semantic index contains this information, but also maps each word's synonyms and hypernyms to the passages containing them, along with their semantic roles and relations. The index scales to retrieve semantically related passages from very large corpora (millions of documents from a single server) in a second or less. The results correspond to the semantic structure of the query, enabling much higher precision than free text searches. The results of the semantic search can be evaluated by the ECD algorithms to test for entailment or contradiction and hence answer the query.

4.1 Indexing

Each document in the corpus is broken into sentences, and the AKR for each sentence is fed into the indexer (see Fig. 3). The lexical identifiers (literal strings or identifiers from linguistic resources) for each word in the AKR are combined with information about their semantic function in the passage to create a set of index term strings. These strings are then associated with occurrence information, which records the relationship of the identifier to the actual word (i.e., alias, synonym, or hypernym and the number of levels in the ontology from the word to the hypernym), along with the document, sentence, and predication containing the word, indicators for montonicity, byte positions of the words in the sentence, and other information.

For example, the AKR for the sentence *Ramazi knows Legrande.* contains the semantic roles Agent and Theme, describing the knowing relation between Ramazi and Legrande. These semantic relations are encoded into index terms by combining the term with the role and its position in the relation, e.g. know:Agent:1, Ramazi:Agent:2 and know:Theme:1 and Legrande:Theme:2. These index terms are associated in the search index with the information about how they occur in the passage.

By looking up know:Agent:1, the system will find all occurrences of any agent knowing anything, and Ramazi:Agent:2 will retrieve occurrences where Ramazi is the agent of some event. By taking the intersection of these occurrence lists, the system finds passages where Ramazi knows something. Likewise, the system finds the intersection of occurrences where the Theme involves knowing Legrande. The occurrence information specifies the predication containing these relations, so the system can find those passages containing references to Ramazi knowing Legrande.

The actual index terms are generated using WordNet concept IDs and alias information, rather than the string. So in this example the term *know* is associated with a number of WordNet IDs (synonyms and hypernyms of the term), and each of these IDs is stored separately in the index. Thus, rather than know:Agent:1, the actual terms stored would be 587430:Agent:1, 588355:Agent:1, 588050:Agent:1, etc. The passages associated with these index terms will be retrieved for any term in a query mapping to the same Wordnet concept.

Finally, this information is inverted to enable efficient lookup of occurrences by index term. The index format is designed to store all of this information for each index term in a highly compressed encoding and to permit lookup with little or no degradation of performance as the corpus grows. The occurrence lists (known as postings) are arranged to take advantage of regularities in the occurrence data, using variable-length integers and delta-encoding for compression (as well as bitvectors for the most frequently occurring terms) and the data is localized using skip-lists to enable efficient disk reads. Each sentence is associated with its containing document, and an arbitrary amount of metadata can be stored for each document.

4.2 Retrieval

At query time (see Fig. 1), the semantic search module receives an AKR for the natural language query. A set of index terms is generated from the query AKR in the same manner used for indexing of passage AKRs, only with simplification of the facts instead of augmentation. The postings for the index terms are retrieved, and the data is processed to find documents containing the appropriately tagged terms occurring in predications that correspond to those in query.

The system attempts to match each semantic fact in the query with each result passage, checking to see that the terms align in corresponding predications. For example, for a query like *Does Ramazi know Legrande?*, the results would include the passage *Legrand is known by Ramazi.*, but it would not include *Ramazi knows Hassan, Hussein knows Legrande, but no one knows the cell leader.*, where both Ramazi and Legrande play roles in a knowing relationship, but not to each other.

This strategy results in high-precision retrieval. As a back-off strategy, the system uses extended key word and key word search techniques. Extended key word search takes advantage of the stemming of word forms and the mapping into WordNet concepts and, for proper nouns, alias facts in order to increase recall of standard key word search. The results of these key word searches are presented separately from the full retrieval results and are not input to ECD.

The retrieval process does not test for strict entailment and contradiction, however. For example, a query of *Did Cheney go to Baghdad?* Might return the passage *Cheney was believed to have gone to Baghdad.*, even though it is not entailed by the query. To check for entailment and contradiction, the results of indexed search can be filtered through the ECD component (section 3) to eliminate false positives for answering the question.

5 Discussion and Conclusions

The Bridge system is a complexly engineered combination of linguistic formalisms based on theoretical criteria. It provides the basis for applications, including entailment and contradiction detection (ECD) and semantic retrieval (Asker). Because the system is under development by a significant number of people working in parallel, it requires a support environment to ensure that changes improve the system in the intended directions, without losing efficiency or accuracy. Some tools supporting this system development are described in Chatzichrisafis et al. (2007).

The architecture provides a layered set of transformations of English text to an abstract knowledge representation. The LFG-based syntactic parsing system produces a dependency structure. The semantics module produces a flattened representation that normalizes these functional dependency structures. It maps grammatical functions into semantic roles and further normalizes the syntactic dependencies, e.g., transforming deverbal nouns and adjectives into their underlying verbal form. Criteria for the syntactic and semantic representations include capturing linguistic generalizations and parallelisms cross-linguistically (Butt et al., 1999, 2002).

The mapping rules for knowledge representation produce descriptions of the concepts under discussion, and a contextual structure that captures the nested structure of contexts. They also specify for each concept whether it is instantiable in the relevant contexts or not. Passage expansion rules add linguistically supported inferences to the representation, to make the import of the sentence explicit in the representation. The criteria for the AKR include natural representation of the distinct meanings of a text, the ability to be transformed into an (extended) first order form for use by other logical reasoners, and support for applications, especially Asker.

The architecture is supported by a collection of linguistic resources, some of which were developed specifically for this system. The broad-coverage English grammar, morphology, and lexicon were developed over many years for a range of applications. The semantics module uses WordNet for its linguistic taxonomic ontology and VerbNet as a resource for transforming grammatical roles into semantic roles. These resources have been extended using syntactic resources from the XLE grammar to produce a Unified Lexicon (UL). In addition, the UL includes lexical markings needed to support normalization, paraphrase, lexical inference, and structural inference. Classes of words that support specific extensions to the initial AKR are lexicalized in the UL. For example, we have identified and categorized over 300 verbs that support pre-suppositional and implicative inference.

Our question answering architecture exploits the AKR representation of sentences. The use of AKR structural components as index terms has significantly improved precision of retrieval from our semantically indexed repository. ECD can be used as a mechanism to answer questions, not just retrieve relevant passages.

References

Bobrow, Daniel G., Condoravdi, Cleo, Crouch, Richard, Kaplan, Ron, Karttunen, Lauri, King, Tracy Holloway, de Paiva, Valeria and Zaenen, Annie. 2005. A Basic Logic for Textual Inference. In *Proceedings of the AAAI Workshop on Inference for Textual Question Answering*.

Bobrow, Daniel G., Condoravdi, Cleo, de Paiva, Valeria, Karttunen, Lauri, King, Tracy Holloway, Nairn, Rowan, Price, Charlotte and Zaenen, Annie. 2007. Precision-focused Textual Inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Braz, Rodrigo, Girju, Roxana, Punyakanok, Vasin, Roth, Dan and Sammons, Mark. 2006. An Inference Model for Semantic Entailment in Natural Language. *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Ob-*

ject Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop pages 261–286.

Butt, Miriam, Dyvik, Helge, King, Tracy Holloway, Masuichi, Hiroshi and Rohrer, Christian. 2002. The Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Butt, Miriam, King, Tracy Holloway, Niño, María-Eugenia and Segond, Frédérique. 1999. *A Grammar Writer's Cookbook*. CSLI Publications.

Chatzichrisafis, Nikos, Crouch, Dick, King, Tracy Holloway, Nairn, Rowan, Rayner, Manny and Santaholma, Marianne. 2007. Regression Testing For Grammar-Based Systems. In Tracy Holloway King and Emily M. Bender (eds.), *Proceedings of the GEAF 2007 Workshop*, CSLI Publications.

Condoravdi, Cleo, Crouch, Dick, Stolle, Rienhard, de Paiva, Valeria and Bobrow, Daniel G. 2003. Entailment, Intensionality and Text Understanding. In *Proceedings Human Language Technology Conference, Workshop on Text Meaning*.

Condoravdi, Cleo, Crouch, Richard, van der Berg, Martin, Everett, John O., Stolle, Reinhard, de Paiva, Valeria and Bobrow, Daniel G. 2001. Preventing Existence. In *Proceedings of the Conference on Formal Ontologies in Information Systems*.

Crouch, Dick, Condoravdi, Cleo, Stolle, Reinhard, King, Tracy Holloway, de Paiva, Valeria, Everett, John O. and Bobrow, Daniel G. 2002. Scalability of redundancy detection in focused document collections. In *Proceedings First International Workshop on Scalable Natural Language Understanding*.

Crouch, Dick, Dalrymple, Mary, Kaplan, Ron, King, Tracy Holloway, Maxwell, John and Newman, Paula. 2007. XLE Documentation, www2.parc.com/isl/groups/nltt/xle/doc/.

Crouch, Dick and King, Tracy Holloway. 2006. Semantics via F-Structure Rewriting. In *Proceedings of LFG06*, pages 145–165, CSLI On-line publications.

Crouch, Richard. 2005. Packed Rewriting for Mapping Semantics to KR. In *Proceedings Sixth International Workshop on Computational Semantics*.

Crouch, Richard and King, Tracy Holloway. 2005. Unifying Lexical Resources. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

de Paiva, Valeria, Bobrow, Daniel G., Condoravdi, Cleo, Crouch, Richard, Kaplan, Ron, Karttunen, Lauri, King, Tracy Holloway, Nairn, Rowan and Zaenen, Annie. 2007. Textual inference logic: Take two. In *Proceedings of the Workshop on Contexts and Ontologies: Representation and Reasoning, Workshop associated with the 6th International Conference on Modeling and Using Context*.

Everett, John O., Bobrow, Daniel G., Stolle, Reinhard, Crouch, Dick, Condoravdi, Cleo, de Paiva, Valeria, van den Berg, Martin and Polanyi, Livia. 2002. Making ontologies work for resolving redundancies across documents. In *Communications of the ACM*, volume 45, pages 55–60.

Fellbaum, Christiane (ed.). 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Frank, Anette, King, Tracy Holloway, Kuhn, Jonas and Maxwell, John T. 2001. Optimality Theory Style Constraint Ranking in Large-scale LFG Grammars. In Peter Sells (ed.), *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 367–397, CSLI Publications.

Gurevich, Olga, Crouch, Richard, King, Tracy Holloway and de Paiva, Valeria. 2005. Deverbals in Knowledge Representation. In *Proceedings of FLAIRS'06*.

Kipper, Karin, Dang, Hoa Trang and Palmer, Martha. 2000. Class-based Construction of a Verb Lexicon. In *AAAI-2000 17th National Conference on Artificial Intelligence*.

MacCartney, Bill and Manning, Christopher D. 2007. Natural Logic for Textual Inference. In *Proceedings of the ACL 2007 Workshop on Textual Entailment and Paraphrasing*.

Maxwell, John and Kaplan, Ron. 1991. A Method for Disjunctive Constraint Satisfaction. In Masaru Tomita (ed.), *Current Issues in Parsing Technologies*, pages 173–190, Kluwer.

Maxwell, John and Kaplan, Ron. 1996. An Efficient Parser for LFG. In *Proceedings of the First LFG Conference*, CSLI Publications.

McCarthy, John. 1993. Notes on Formalizing Context. In *Proceedings of the 13th Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 555–560.

Nairn, Rowan, Condoravdi, Cleo and Karttunen, Lauri. 2006. Computing Relative Polarity for Textual Inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics)*.

Riezler, Stefan, King, Tracy Holloway, Kaplan, Ronald M., Crouch, Richard, Maxwell, John T. and Johnson, Mark. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of ACL02*.

van Benthem, Johan. 1986. *Essays in Logical Semantics*. Reidel.