Textual Inference Logic: Take Two

V. de Paiva, D. G. Bobrow, C. Condoravdi, R. Crouch, L. Karttunen, T. H. King, R. Nairn, and A. Zaenen

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto CA 94304*

Abstract. This note describes a logical system based on concepts and contexts, whose aim is to serve as a representation language for meanings of natural language sentences. The logic is a theoretical description of the output of an evolving implemented system, the system Bridge, which we are developing at PARC, as part of the AQUAINT program. The note concentrates on the results of an experiment which changed the underlying ontology of the representation language from CYC to a version of WordNet/VerbNet.

1 Introduction

This note describes a second version of a logical system based on concepts and contexts, whose aim is to serve as a representation language for meanings of natural language sentences. This representation language is constrained in two different directions: on the one hand we want the *mapping* from English to this language to be as easy as possible, hence we want a very expressive logical language. On the other hand we want to do *reasoning* with this language, so we want to constrain its complexity as much as possible.

The first version of this logic of contexts and concepts was called TIL (for Textual Inference Logic) and was described in [1]. The logical system in this paper, which we call TIL2 (for Textual Inference Logic Two), the formalization of the implemented system Bridge that we are developing at PARC as part of the AQUAINT framework, shares with TIL its main characteristics: it is a logical system of concepts and contexts, where declaration of instantiability of an instance of a concept in a context specifies the truth of assertions concerning that concept in that context. Uninstantiability is the negation of instantiability. Some higher level discussion of the rationale behind the systems can be found in [6, 2, 4].

The main difference between TIL and TIL2 is the change of the underlying ontology from the CYC one to an in-house version of a merge of WordNet/VerbNet. Here we concentrate on the results of this experiment in the change of ontology. We also discuss briefly the problem of evaluating the quality of the representations produced by the system Bridge.

* This work was sponsored in part by DTO. Approved for Public Release; distribution unlimited.

For those interested, the system Bridge parses sentences in English using our industrial-strength parser XLE and a hand-crafted lexical functional (LFG) grammar[11]. Parsed sentences are mapped to f-structures and f-structures are then mapped to linguistic semantic structures. These are mapped to AKR (Abstract Knowledge Representation) structures using a robust rewriting system [6]. This layered approach to producing logic from text is useful and natural. We have discussed elsewhere the (perhaps less natural) main characteristics of our approach: the 'packing' of all these structures. By 'packing' we mean that instead of disambiguating structures (grammatical ones, semantic ones and knowledge representation ones) and pruning the less likely ones at each stage of the pipeline, our algorithms allow us to keep a condensed representation of all possibilities, effectively avoiding premature pruning of the correct choices.

Given that our logical AKR representations are intimately connected to the underlying ontology, one might expect that the change of ontology from CYC to WordNet/VerbNet would necessitate a total reworking of the system Bridge. This turned out not to be case, the re-architecture of the system was surprisingly easy and almost trouble-free. It is true that new trade-offs were made and these are some of the issues that we discuss here. But before discussing trade-offs we should explain *why* this change of ontology and *what* are the representations obtained in TIL2.

Our aim is to map free text to logical formulae based on a conceptual hierarchy that one can reason with. Our initial intuition was to try to use the concepts provided by the biggest knowledge base available CYC, and to take advantage of its reasoning component, which was familiar to some of us. Although we used CYC concepts for our first logic, we found it useful to map the text to an *abstract* form of knowledge representation (AKR), that could be realized as CYC or KM or any other knowledge representation formalism. The design of this AKR aimed for a sweet spot between ease of mapping from text to a formalism, and mapping from that formalism to standard logical representations. A happy surprise was our realization that the AKR representations were already good enough for some important classes of textual inferences that we wanted to concentrate on. In general, the inferences we wanted to concentrate on were immediate, almost simple-minded, but necessary for the understanding of the text. For example, if the text says that "John managed to close the door" then we can safely infer that "John closed the door" and this kind of immediate inference is absolutely necessary to answer questions, based on snippets of text, as is the case in our primary application. Furthermore, these inferences did not seem to depend crucially on the particular ontology; they were much more dependent on the articulation of inference patterns surrounding the use of particular classes of words which appear quite often in open texts.

At the same time, we were having serious difficulties completing mappings from open texts when using the CYC system. CYC's mappings from word to CYC concepts are very sparse, as might be expected from a knowledge base not built to model language. We realized that having good information, very deep, about some concepts and nothing at all about others was worse than having superficial information about most words. Thus we decided to move to a WordNet/VerbNet ontology, or more precisely, to the projection of WordNet/VerbNet obtained from our own Unified Lexicon[5].We considered trying to extend the mappings from WordNet to CYC; however, we found that concepts implicit in WordNet covered a broader range than those in CYC, and we found no automatic way of extending the mappings from WordNet senses to CYC concepts.

As in the previous version of the system the logic is based on the notion of *events* expressed in a neo-Davidsonian style [10]. We use the neo-Davidsonian notation because it supports easy handling of optional/missing arguments. We couple this with use of *contexts* based on McCarthy's ideas [8]. McCarthy's contexts have two properties that we cash out in our system. The first is that within a context, reasoning can be done locally. So for example, if *Ed leaves Berlin.* then whether this is in a hypothetical/counterfactual, or real-world context, in that context one can conclude that Ed was in Berlin. The second property of McCarthy's contexts is the existence of context-lifting rules that relate statements in one context to ones in nested contexts. We show how linguistic structures provide a framework for different classes of such context-lifting rules.

We first describe the logical system, using several examples. Rather than listing all kinds of relations between contexts and concepts that the implemented system produces, we aim to give a feel for how our representations look like. Then we discuss the changes, gains and losses, caused by the change of ontology from CYC to WordNet/VerbNet. Finally we discuss methods and criteria for evaluating the coverage of the logical system obtained. We close with some ideas for further work.

2 TIL2 via examples

It is traditional for logics of Knowledge Representation to be fragments of first-order logic (FOL). It is traditional for logics for natural language semantics to be higher-order intensional logics. Our logic has concepts, which make it look like "description logics", that is, fragments of FOL, but it also has contexts, a possible-worlds-like construct that, we hope, is expressive enough for the needs of natural language.

Concepts, the way we conceive them, come from both neo-Davidsonian event semantics and, somewhat independently, from description logics. Some of our reasons for using a concept denoting analysis instead of an individual denoting analysis when mapping noun phrases to logic are discussed in [3]. The main reasons are being able to deal with non-existent entities (for example when mapping *"Negotiations prevented a strike"* we do not want to say that there exists negotiations N and there exists a strike S and $\mathit{prevented}(N, S)$, as the prevented strike does not really exist in the actual world).

One of the main differences between TIL and TIL2 consists in the type of concepts that are used. While in the previous logic TIL the basic ontology was the CYC ontology, for TIL2 the basic ontology is WordNet/VerbNet. But whatever the basic ontology, concepts in our logic are of two very different kinds: the first kind of concepts are given a priori, sitting in a established hierarchy, based on the hierarchy underlying either CYC or the synsets of WordNet, considered as a taxonomy. The second kind of concepts are dynamic, created by the implemented system Bridge when we feed it an English sentence. The dynamic concepts are created and placed in the hierarchy in use, as best as we can, at run time.

For example, when using the CYC ontology, for the sentence *A zebra slept*, we use two CYC concepts `Zebra` and `Sleeping` and two dynamic concepts $zebra : 1$, a subconcept of the CYC concept `Zebra` and $sleep : 11$, a subconcept of the CYC concept `Sleeping`. Now when the same sentence is analyzed in the WordNet/VerbNet version of the system, the dynamic concept $zebra : 1$ will be mapped to a subconcept of the WordNet synset corresponding to the zebra animal, but the dynamic concept corresponding to the zebra's sleeping, $sleep : 11$ will be mapped to **two** different static concepts in WordNet, one corresponding to the WordNet meaning of animal sleep, the other corresponding to "accommodate", as in the sentence *The tent sleeps six*.

The concepts in WordNet are treated by Bridge, following WordNet convention, using the synset numbers. These are not very easy to read, hence the system pretty-prints it as a head word of the synset, followed by a number. The dynamic concepts are written as the word colon a number, showing that this is simply a Skolem constant. For example a clause like $subconcept(sleep : 11, [sleep-1, sleep-2])$ means that the dynamic subconcept of the zebra sleep ($sleep : 11$) is either a subconcept of $sleep-1$ or of $sleep-2$.

The most underspecified concept in the WordNet hierarchy is **entity**, which corresponds to the concept `Thing` in CYC. All our concepts are subconcepts of the most underspecified concept. We assume that that there are no circularities nor inconsistencies[1] in the given initial hierarchy, be that CYC or WordNet.

The second main difference between TIL and TIL2 has to do with how the concepts are related, when expressing propositions. In both systems concepts are related via "role" assertions, but the kinds of roles available are different. Thus continuing on with the same example *"The zebra slept"* when using the CYC ontology, we were able to use the CYC role `bodilyDoer` to connect the sleeping event concept to the zebra concept, so the representation ends up with the two subconcept clauses plus a clause for $role(bodilyDoer, sleep : 11, zebra : 1)$, while the representation using the WordNet/VerbNet ontology is very similar, but has instead $role(Agent, sleep : 11, zebra : 1)$ The CYC role `bodilyDoer` is much more specific than the role `Agent` from the much more limited collection of VerbNet roles. Our unified lexicon ([5]) provides a mapping from the grammatical relations produced by our XLE/LFG parser to the concept and role structure based on the information in VerbNet. While many of our roles resemble linguistic "thematic roles", the view here is more general and we have many roles that do not correspond to thematic roles, see below. Roles are written as ternary relations, in a prefix notation, i.e. $role(t_1, t_2, t_3)$ where t_1 is the name of the role and t_2 and t_3 are the concepts in the binary relation named by t_1. Thus the intuitive meaning of $role(Agent, sleep : 11, zebra : 1)$ is that in a particular

[1] This is a big assumption, but we hope others are working on the problem.

context there is an sleeping event (a generic sub-concept of the sleeping concept) and there is a zebra (a generic sub-concept of the concept of zebra) such that the relation *Agent* relates this zebra and this sleeping event.

The logical system described so far looks like a description logic. We have concepts Concept with their own partial order (written as $subconcept(t_1, t_2)$) and roles Role, which are binary relations on the set of concepts Concept. We write *clauses* that either relate concepts via subconcept relations or relate roles to pairs of concepts, like $role(Agent, sleep : 11, zebra : 1)$. And we write collections of clauses that correspond to representations of natural language sentences and hence correspond to propositions.

But our simple logic has contexts Context, as well as concepts. There is a first initial context (written as t) that corresponds roughly to what we take the world to be like, as far the author of the sentence is committed to. But since this circumlocution is awkward, we will usually talk about this top level context as the 'true context'.

Contexts in our logic were conceived as syntactic ways of dealing with intensional phenomena, including negation and non-existent entities. They support making existential statements about the existence and non-existence in specified possible worlds of entities that satisfy the intensional descriptions specified by our concepts. The possible worlds reflect the worlds implicitly (partially) described by the author of a text. Authors statements of propositional attitudes clearly require use of intensional terms, since no existence in the real world can be implied by such descriptions. It is clear that intensional notions are required when dealing with the representation in logic of *propositional attitudes*. We use propositional attitudes as an example of our use of contexts.

Propositional attitudes predicates relate contexts and concepts in our logic. Thus a concept like 'knowing' or 'believing' or 'denying' introduces a context that represents the proposition that is known, believed or denied. For example, if we want to represent the sentence *Ed denied that the diplomat arrived*, we will need concepts for the arriving event, for the denying event, for the diplomat and for Ed. And we will need roles that describe how these concepts relate to each other. Thus we need to say who did the 'denying' and 'what was denied' and who did the arriving. The content of what was denied in the denying event is the proposition corresponding to *The diplomat arrived*. The role corresponding to 'what was denied' relates a dynamic concept, the concept of the denying event (written as $deny : 4$), to (the contents of) a new context. To name this new context we use its 'context head'. The context head is the arriving event, so the new context is called $context(ctx(arrive : 4))$ ('contex-head' is one of the many roles in the system that is not a thematic role).

Contexts allow us to localize reasoning: the existence of the denying event and of Ed are supposed to happen in the true world, but the existence of the arrival of the diplomat is only supposed to happen in the world of the things denied by Ed. In particular the arrival event could be considered as not happening, if Ed is known as a reliable source. (The system takes no position as to the instantiability or not of the arrival event in the top context: the instantiability of the arriving is only stated in the context of the things denied by Ed.) In some cases (for example if the sentence was *Ed knew that the diplomat arrived*) we can percolate up the truth of assertions in inner contexts up to the outside context. In many cases we cannot. The happening or not of events is dealt with by the *instantiability/uninstantiability* predicate that relates concepts and contexts.

While we may be prepared to make the simplifying assumption that if 'X is known' than 'X is true', we certainly do not want to make the assumption that if 'X is said' than 'X is true'. We say that the context introduced by a knowing event is *veridical* with respect to the initial context t, while the context introduced by a saying event is *averidical* with respect to the initial context. Negation introduces a context that is *anti-veridical* with respect to the original context. Thus we have a fairly general mechanism of contexts (these can clearly be iterated), which can represent some positive and some negative information. Similarly to McCarthy's logic we also have 'context lifting rules' that allow us to transfer veridicality statements between contexts, in a recursive way.

Our representations also have a (preliminary) layer of temporal representation on them. The idea is to order events according to their times of happening and with respect to some generic time 'Now'.

A few words on related work: Clearly our goals and motivations are very similar to the SNePS project[12]. We share the use of intensional notions and of contexts, with a logic approach that strives for the right amount of expressivity. But the differences are overt: we do not feel the need for belief revision. We deal with snapshots of the author's world, not with systems of beliefs. Our basic logic system is constructive, not relevant and paraconsistent. While both logics have been (and are being) designed to support natural language processing and commonsense reasoning, they are implemented very differently.

3 Changing the Ontology

Changing the ontology allowed us to talk about ambiguity-enabled or packed representations. While we could, in principle, do the same with the CYC ontology and we did so, to a limited extent, in practice we simply didn't have the different concepts for each word. For many words we did not have a single concept associated to it, for very few we did have more than one. So we were restricted to what the ontologists in CYC thought the meaning of a given word was. (Of course we are now constrained to the meanings that the lexicographers at WordNet think one should have, but the pool is much bigger. So we do not have the problem of "missing concept for skolem", by and large). Thus a sentence like "Ed arrived at the bank" will not be assigned simply one of possible meanings of "bank" (river bank or financial institution). Actually it will map to any of the ten possible meanings of bank in WordNet. Also "arrive" will be mapped to two different meanings, the physical reaching of a destination and the somewhat metaphoric, succeed in a big way. But instead of having twenty different representations for the meaning of the sentence, sharing the concepts 'Ed', 'arrive', 'bank' and the

VerbNet roles for 'arrive', we have a single representation packing all of this as

$$subconcept(arrive:4, [arrive-1, arrive-2])$$
$$role(Experiencer, arrive:4, Ed:1)$$
$$role(Cause, arrive:4, bank:15)$$
$$subconcept(Ed:1, [male-2]])$$
$$subconcept(bank:15, [bank-1, \ldots bank-10])$$

One bad side of this is that we are forced (to begin with, at least) to use the very uninformative VerbNet roles. Thus in the example above we end up with one sensibly named role $role(Experiencer, arrive:4, Ed:1)$ and one not so sensibly named $role(Cause, arrive:4, bank:15)$. We have discussed ways of augmenting the number of roles of VerbNet (from less than twenty) to a reasonable number, presumably much less than the 400 that CYC has, but have found that a daunting task, so we are still exploring possibilities. Roles in our system are supposed to support inference and at the same time are supposed to make the mapping from language feasible. For the latter purpose (mapping from language feasibly) VerbNet roles are well-suited, but they are too underspecified to help with inference. The quest is on to find a collection of roles that keeps feasibility of the mapping, but improves the inferential capabilities.

While the mechanism that implements the packing of representations could be used with the CYC ontology too, the actual details of the previous implementation, which looked at noun concepts before verb concepts (given CYC's more extensive coverage of nouns) made packed representations the exception rather than than the norm. In any case packing makes more sense when using WordNet/VerbNet where we do have many concepts for each word.

Another feature of our use of the new ontology is that it does not enforce "sortal restrictions". Using CYC we could make sure that in the sentence *Ed fired the boy* the verb 'fire' was used with the meaning of what CYC calls `DischargeWithPrejudice`, while in *Ed fired the cannon* it was used with a `ShootingAGun` meaning. With the new ontology we do not weed out even the worst clashes of meanings. But a single representation covers a multitude of meanings. We take this as a shortcoming that we plan to address in future work.

4 Inferences and Design Decisions

The reason for introducing event concepts was the fact that they make some inferences that can be complicated in other semantical traditions very easy. For example it is obvious how to obtain *Ed arrived in the city* from the sentence *Ed arrived in the city by bus.* This inference corresponds simply to conjunction dropping in our logic. But of course there is much more to textual inference than simply dropping conjuncts.

To test textual inference our system provides a method for detecting entailment/contradictions, called "qa" for the application in question answering. When given two passages "qa" tells us whether the second passage is entailed by the first one or not. Simple subconcept/superconcept reasoning is handled.

In addition we support some pre and post condition reasoning. So *Ed arrived in the city* does entail that *A person arrived in the city*, since Ed is a person. Similarly *Ed arrived in Rome* should entail that *Ed arrived in a city*, as 'Rome' is a city, but given that the proper names in WordNet are somewhat sketchy, we do not use this facility.

Note that the clauses we construct satisfy the usual monotonicity patterns, both in positive and in negative form. Thus *Ed arrived in the city by bus* entails that *Ed arrived in the city.* But *Ed did not arrive in the city* entails that *Ed did not arrive in the city by bus*, while *Ed did not arrive in the city by bus* does **not** entail that *Ed did not arrive in the city.*

We have also implemented the transformation of nominal deverbals with their respective arguments into verb-argument structures. The work is described in [7]. It allows us to conclude from a sentence like *Alexander's destruction of the city happened in 332 B.C.* that the sentence *Alexander destroyed the city in 332 B.C.* follows.

We have done significant work on exploring how certain linguistic expressions support classes of context-lifting rules. Using the context structure of our logic we support inferences associated with kinds of verbs with implicative behavior. In our unified lexicon, the classes of such behavior are marked. This work is discussed in [9]. Here we simply give an example of each one of the classes of "implication signatures" or implicative behavior described by Nairn, Condoravdi and Karttunen. There are nine such classes, depending on whether positive environments are taken to positive or negative ones. Thus for example the verb "manage" takes positive predicates (e.g "Ed managed to close the door" → "Ed closed the door") to positive predicates and negative ones ("Ed didn't manage to close the door" → "Ed didn't close the door"). By contrast the verb "forget (to)" inverts the polarities: "Ed forgot to close the door" → "Ed didn't close the door" and "Ed didn't forget to close the door" → "Ed closed the door".

More complicated are the verbs that only show their implicative behavior either in positive or negative situations. For example we have positive implicatives like the verb "force (to)" takes positive polarities and produces positive polarities (e.g "Ed forced Mary to paint" → "Mary painted"), but if "Ed didn't force Mary to paint" we cannot tell whether Mary painted or not. While "refuse (to)" only works to produce negative polarity (e.g. 'Mary refused to sing" → "Mary did not sing"). There are also negative implicatives like "attempt (to)" and "hesitate (to)" which again only work for a negative polarity, but produce a positive one ("Ed didn't hesitate to leave" → "Ed left", but if "Ed hesitated to leave" we cannot tell whether he left or not).

Finally we have factives and counterfactives, examples being "forget (that)" ("Ed forgot that Mary left" → "Mary left" and "Ed didn't forget that Mary left" → "Mary left" and "pretend that" ("Ed pretended that Mary left" → "Mary didn't leave" and "Ed didn't pretend that Mary left" → "Mary left"). And the neutral class, where we cannot say anything about the veridicity of the complement ("Ed said/expected that Mary left"). Further work is in progress to mark implicative behavior of verbs that do not take sentential complements.

One of our difficult design decisions was over the treatment of copula. It was clear that one needed to have trivial inferences like "Ed is a clown" contradicts "Ed is not a clown". But the mechanism used to infer that should be also able to cope with answering yes to "Ed, the clown, slept" implies that "A clown slept" and several other similar and not so similar inferences.

5 Towards Evaluation

From the beginning we faced the problem of measuring the 'quality' of our representations. One can try to measure that by manually inspecting the representations themselves and checking that the arguments provided by the system correspond to our intuitions. But this is not very efficient nor objective. We can also try to measure the faithfulness of the representations by checking whether the system can answer correctly questions, using these representations. We have tried this indirect method in the AQUAINT pilot KB Evaluation and the measuring is quite difficult because question/answer pairs usually have to deal with several logic-linguistic issues at once. We devised pairs of question/answers that try to focus on a particular specific problem at a time. Thus we have test-suites checking mostly deverbal nouns, or anaphora resolution or coordination of sentences, etc. But besides being time consuming and laborious, it is not clear that this would measure adequacy or faithfulness of the representations in a fair way. At the moment, it seems to us that the best that can be done is to try to look at textual entailment, as originally proposed by the PASCAL RTE but modify it to deal with the issues that we consider important, like the implicative behavior of lexical items and especially the need to distinguish between entailment of a negation from not being able to draw a conclusion [13].

6 Conclusion

This note only starts the discussion of the kinds of inferences that we expect to be able to make using our logic of concepts and contexts. On the positive side we have an implemented system Bridge that it is easy to modify as it relies on a heavy duty rewriting system (the transfer system[6]) capable of packing efficiently large amounts of representations, be they f-structures or AKR-structures. This system proved to be robust enough to cope with a very radical change of ontologies. Moreover, the abstract description of the system needed almost no modification.

On the negative side, much work remains to be done to get the system working as well as we want it to. First we still have a long way to go as far as improving the representations is concerned. Amongst the issues we have not discussed here are how to deal with noun-noun compounds, how to deal with contexts introduced by adjectives and adverbs and how to deal with temporal modifiers and temporal interpretation in general. We have done some work on these problems and hope to describe that work elsewhere.

We have said nothing about how to deal with lexical entailments such as *Ed snored* implies that *Ed slept*. We are not sure whether this problem should be addressed by creating enriched representations (maybe the concept of 'snoring' must include a "concurrent" necessary condition of 'sleeping') , or whether such inferences should be handled in the entailment/contradiction algorithm. So we are back to the previous trade-off between easiness of mapping and easiness of reasoning.

References

1. D. Bobrow, C. Condoravdi, R. Crouch, R. Kaplan, L. Karttunen, T. H. King, V. de Paiva, and A. Zaenen. A basic logic for textual inference. In *Proceedings of the AAAI Workshop on Inference for Textual Question Answering, Pittsburg, PA*, 2005.
2. C. Condoravdi, D. Crouch, R. Stolle, V. de Paiva, and D. Bobrow. Entailment, intensionality and text understanding. In *Proceedings Human Language Technology Conference, Workshop on Text Meaning, Edmonton, Canada*, 2003.
3. C. Condoravdi, R. Crouch, M. van der Berg, , J. Everett, R. Stolle, V. de Paiva, and D. Bobrow. Preventing existence. In *Proceedings of the Conference on Formal Ontologies in Information Systems (FOIS). Ogunquit, Maine*, 2001.
4. D. Crouch, C. Condoravdi, R. Stolle, T. King, V. de Paiva, J. O.Everett, and D. Bobrow. Scalability of redundancy detection in focused document collections. In *Proceedings First International Workshop on Scalable Natural Language Understanding (ScaNaLU-2002), Heidelberg, Germany*, 2002.
5. R. Crouch and T. H. King. Unifying lexical resources. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes, Saarbruecken, Germany.*, 2005.
6. Richard Crouch. Packed rewriting for mapping semantics to KR. In *Proceedings Sixth International Workshop on Computational Semantics, Tilburg, The Netherlands*, 2005.
7. O. Gurevich, R. Crouch, T. H. King, and V. de Paiva. Deverbals in knowledge representation. In *Proceedings of FLAIRS'06, Melbourne Beach, FA*, 2005.
8. J. McCarthy. Notes on formalizing context. In *Proceedings of the 13th Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 555–560, 1993.
9. R. Nairn, C. Condoravdi, and L. Karttunen. Computing relative polarity for textual inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics). April 20-21, 2006. Buxton, UK*, 2006.
10. T. Parsons. Underlying eventualities and narrative progression. In *Linguistics and Philosophy, 681-99.*, 2002.
11. S. Riezler, T. H. King, R. Kaplan, R. Crouch, J. T. Maxwell, and M. Johnson. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia, PA*, 2002.
12. S. C. Shapiro. SNePS: A logic for natural language understanding and commonsense reasoning. In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language, L. Iwanska, S. Shapiro (editors), AAAI Press/MIT Press*, 2000.
13. A. Zaenen, L. Karttunen, and R. Crouch. Local textual inference: can it be defined or cicumscribed? In *Proceedings of the AC 2005 Workshop on Empirical Modelling of Semantic Equivalence and Entailment, Ann Arbor, MI*, 2005.