AN ENGLISH TO LOGIC TRANSLATOR FOR ONTOLOGY-BASED KNOWLEDGE REPRESENTATION LANGUAGES

Adam Pease

278 Monroe Dr #30
Mountain View CA 94040
adampease@earthlink.net

William Murray

Teknowledge, 1800 Embarcadero
Palo Alto, CA 94303
wmurray@teknowledge.com

ABSTRACT

Ontologies provide advantages of knowledge reusability, sharing, and greater robustness when used to build large knowledge-based applications. Unfortunately, translating between English statements and a specific ontology requires skill in knowledge engineering and an understanding of formal logic and the ontology itself. A knowledge engineer must be familiar with the concepts in the ontology, the fine distinctions between terms, and the specific way the ontology conceptualizes the world.

We are developing a tool, CELT (Controlled English to Logic Translation), to enable non-programmers to add knowledge expressed in terms of an ontology. CELT is an automatic translation tool to convert controlled English to KIF formulas using ontologies built with the Suggested Upper Merged Ontology (SUMO). WordNet provides a base lexicon and a default preference for word senses. We do not expect CELT to obviate the need for knowledge engineers but to instead better leverage their time, as current machine translation tools assist professional human translators.

CELT uses Discourse Representation Theory to handle the translation of multiple sentences, the use of logical quantifiers, and the resolution of anaphoric referents. Individual sentences are parsed using a Definite Clause Grammar augmented with feature grammar extensions.

CELT is domain-independent but can be customized for particular domains by providing domain-specific ontologies and lexicons. The lexicons can specify both technical terms and domain-specific preferred word senses for common terms. CELT translates sentences to assertions and queries for a first-order logic theorem prover.

1 INTRODUCTION

Ontologies allow less brittle knowledge-based applications by providing a deeper grounding for all terms. They provide some of what is meant by 'common sense', in addition to facilitating sharing between knowledge-based applications and promoting reusability of knowledge.

Unfortunately, large ontologies are at least as difficult to learn as large class libraries as they define many different concepts, with fine distinctions between them, and a particular way of conceptualizing the world.

To encode domain-specific knowledge in an ontology is time-consuming and error-prone if the translations are performed manually. We have created a tool to automate the process in the same way that machine translation tools automate the process of translation for human linguists: the tool manages the bulk of the translation effort and a human (in our case a knowledge engineer) fine tunes the result and corrects any misinterpretations, such as where a wrong word sense was selected.

CELT automates the translation process and provides warnings when it applies heuristics to simplify later post-editing.

CELT generates expressions in the first order logic language of Knowledge Interchange Format [7]. The terms in the resulting KIF expressions come from SUMO [12]. WordNet [4] provides a large basic vocabulary. Domain-specific lexical and ontological extensions are required for new domain applications.

1.1 Use of Controlled English

CELT uses a controlled English grammar. The grammar is domain-independent and employs WordNet for a vocabulary of 100,000 word senses. Our research on CELT was originally inspired by the controlled English of ACE [6]. ACE was designed to provide a formal specification language for hardware and software systems in an expressive and easily readable subset of English. Each sentence has only one possible parse. Unlike ACE, CELT has a built-in lexicon of tens of thousands of words, imported from WordNet and manually mapped to SUMO concepts. So the following sentences are examples of typical CELT sentences:

Henry the Eighth rules England.
John eyes the pumpkin.
Mary's car runs into the river bank.

At present CELT only translates present indicative verbs and singular nouns into KIF. This choice, as well as other restrictions, allows us to perform a deterministic interpretation of the semantics of the sentence. We intend to lift these restrictions as we extend CELT. Currently, we use morphological processing rules, derived from the "Morphy" code of WordNet, to transform other verb tenses and plural verbs into the tense and number required. Thus CELT can also accept,

Henry the Eighth ruled England.
John eyed the pumpkins.
Mary's car ran into the river bank.

The key point here is that WordNet provides a very large initial vocabulary and CELT automates the process of mapping sentences with terms from WordNet to their more formal representations in KIF formulas and the Standard Upper Merged Ontology.

1.2 Translation into Discourse Representation Structures

CELT accepts multiple sentences such as

John sees the hamburger. He eats it. He is happy.

and uses Discourse Representation Theory (DRT) [10] to represent dialog context and to resolve anaphoric references.

DRT is also used to handle quantifiers in sentences. *Every farmer beats a donkey* is expressed in DRT as if it were the implication *If x is a farmer then x beats a donkey.*

DRT also specifies how anaphoric references are to be resolved. CELT generates warnings when it cannot find the antecedent reference according to the rules of DRT.

DRT uses a representation of the discourse context called a discourse representation structure (DRS). The DRS enumerates the discourse referents (objects that have been talked about) in the current context. Embedded DRSes can use anaphoric references to ancestor (surrounding) DRSes. DRT also provides many special purpose rules. For example, a consequent DRS can access the discourse referents in an antecedent DRS in an implication.

After sentences have been represented in DRSes these DRSes are further simplified by DRS reduction rules. These rules convert quantified sentences into implications, conjunctions into multiple DRSes, etc.

Individual sentences are parsed into a frame-slot representation at this point. We discuss sentence and query translation in the next section.

Code generation traverses the DRS structure until it reaches the simple sentences (no conjunctions, no quantifiers) contained within them. The code fragments generated for the simple sentences are combined according to the logical relations expressed among the DRSes that contain them.

1.3 Translation into Frame-Slot Sentence Parses

Individual sentences are parsed into a frame-slot representation. For example, the slots of a noun phrase represent the grammatical features of the noun (case, gender, etc.) along with semantic information to aid the translation, such as the SUMO term the noun maps to, or a lambda-expression containing SUMO terms if the mapping is not one-to-one.

CELT is implemented in SWI-Prolog and its grammatical rules are expressed in a Definite Clause Grammar (DCG). The DCG formalism is extended with the feature grammar extension of GULP 3.1 [3]. Thus CELT's grammar rules form a unification grammar.

No special parsing procedures are applied; instead, the translations of the DCG rules are sufficient. Most parses take less than a second real time on 500 MHz PCs.

```
Entity
   Physical
      Object
         SelfConnectedObject
            CorpuscularObject
             OrganicObject
             Artifact
         Collection
         Agent
         Region
      Process
      DualObjectProcess
         IntentionalProcess
         Motion
      InternalChange
   ShapeChange
   Abstract
      Quantity
         Number
         PhysicalQuantity
      Attribute
      SetOrClass
      Relation
      Proposition
Graph
GraphElement
```

1.4 Ontology

CELT uses the Suggested Upper Merged Ontology [12] as the vocabulary for its logical output. SUMO contains roughly 1000 terms and 4000 axioms including 750 rules. It has been mapped by hand to all 100,000 word senses in WordNet 1.6.

The figure at left shows the top of the SUMO taxonomy. SUMO provides the structure needed for logical reasoning, as well as the vocabulary for output. For example, if CELT is told that a person is dead, it can infer that the person will not play the role of agent in any future action. Such a conclusion is derivable from the logical rules in SUMO.

SUMO is extended with a number of domain ontologies that together with SUMO total 20,000 terms and 60,000 axioms. A mid-level ontology has also been created and 5000 of the most frequently used WordNet synsets have been mapped to those more specific terms.

2 CELT TRANSLATION EXAMPLE

We provide examples of the translation process below. Assume the input to be translated is:

John eats every hamburger that he sees. He observes a hamburger at Wendys.

Next we could ask: *Who eats what?* CELT would answer *John eats a hamburger in Wendys* using a First-Order Logic theorem prover such as SNARK [16] or OTTER [10].

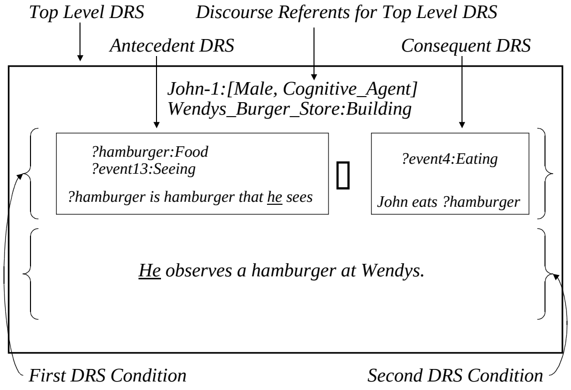
We go through the translations below.

2.1 DRS Representation

Initially we start with one top-level DRS. CELT adds the two sentences given as part of the DRS. The two sentences are first separated into two separate DRS conditions, both of which must be true for the DRS to be true.

So far there is just one DRS. But the next DRS reduction rule that is applied transforms the quantified sentence to an implication which has two parts, an antecedent and a consequent, each wrapped in its own DRS and embedded in the top-level DRS. This transform essentially rewrites the first sentence to ***if*** *x is a hamburger that he sees* ***then*** *John eats x* (for all x).

The second sentence *'He observes a hamburger at Wendys*.' remains part of the top-level DRS. It is a DRS condition that follows the implication. The implication itself is a DRS condition represented by the implication DRS and the two antecedent-consequent DRSes further embedded in it.


Top Level DRS
Discourse Referents for Top Level DRS
Antecedent DRS
Consequent DRS
John-1:[Male, Cognitive_Agent]
Wendys_Burger_Store:Building
?hamburger:Food
?event13:Seeing
?hamburger is hamburger that he sees
▯
?event4:Eating
John eats ?hamburger
He observes a hamburger at Wendys.
First DRS Condition
Second DRS Condition

According to DRT proper nouns are added to the top-level DRS, so both 'John' and 'Wendys' appear as discourse referents in the top-level DRS.

The other discourse referent is 'he'. CELT determines that 'he' refers to 'John' as both are male, whereas 'Wendys' refers to a building.

At this point CELT would generate the following KIF code if the input were only *John eats every hamburger that he sees.*

```
(exists
  (?event4)
  (forall
    (?hamburger ?event13)
    (implies
      (and
        (instance ?hamburger Food)
        (patient ?event13 ?hamburger)
        (instance ?hamburger Physical)
        (experiencer ?event13 John-1)
        (attribute John-1 Male)
        (instance ?event13 Seeing))
      (and
        (instance ?event4 Eating)
        (attribute John-1 FullyFormed)
        (attribute John-1 Male)
        (instance John-1 Human)
        (agent ?event4 John-1)
        (patient ?event4 ?hamburger)))))
```

2.2 Noun Phrase and Sentence Parsing

The antecedent DRS is a noun phrase, in this case one with a relative sentence modifying it. The consequent is a complete sentence. CELT uses the same grammar for parsing both, but starting at different nonterminals (NP versus Sentence). The NP parse in turn calls for parsing a relative sentence ('that he sees'), which is handled as an embedded sentence with a 'gap' ('he sees *gap'*, with 'that' referring back to 'hamburger', and being used to plug the gap). The Sentence parse itself calls for parsing an NP in the subject position and one in the direct object position as the lexicon indicates 'eats' can be transitive or intransitive.

In all cases CELT takes the first successful parse as the CELT grammar requires that there be only one parse per sentence.

2.3 Frame-Slot Parse Structures

As each nonterminal is parsed a frame-slot structure is returned as the set of features for the nonterminal parse. For example, the features for 'a hamburger' are:

```
sem<->
     noun<->hamburger
     head<->?hamburger60
     type<->Food
     quan<->existential
     id<->105738264
syn<->
     reply<->[a, hamburger]
     det<->a
     ncat<->object
```

The actual features for this example also include a 'sub' feature for the relative sentence. The value for that feature is always a sentence parse.

A sentence parse has features for the subject, direct object, indirect object, and predicate. The first three are parsed as NPs, however either or both of the sentence objects may be empty. CELT's parse of 'John eats a hamburger' is shown below:

```
sem<->
     head<->?event68
     id<->200794578
     subj<->
           noun<->John
           head<->John-1
           type<->[Human, Male, FullyFormed]
```

```
            quan<->definite
        pred<->Eating
        dobj<->
            noun<->hamburger
            head<->?hamburger72
            type<->Food
            quan<->existential
            id<->105738264
        adjs<->[]
    syn<->
        rel<->no
        act<->eat
        vcat<->[_G2534, transitive, _G2540]
        max<->3
        role<->assertion
```

In the DRS implication representation the NP is moved to the antecedent and replaced by a pointer to that NP.

2.4 KIF Code Generation

The final code generated combines the code for *John eats every hamburger that he sees.* with the code generated for *He observes a hamburger at Wendys.* The use of DRSes and anaphoric resolution allows 'He' to refer to 'John'.

```
(and
  (instance John-1 CognitiveAgent)
  (attribute John-1 Male)
  (attribute John-1 FullyFormed)
  (instance John-1 Human)
  (exists
    (?event4 )
    (forall
      (?hamburger ?event13)
      (=>
        (and
          (instance ?hamburger Food)
          (patient ?event13 ?hamburger)
          (instance ?hamburger Physical)
          (experiencer ?event13 John-1)
          (instance ?event13 Seeing))
        (and
          (instance ?event4 Eating)
          (agent ?event4 John-1)
          (patient ?event4 ?hamburger)))))
 (exists (?event 17 ?hamburger21)
   (instance ?event17 Seeing)
   (experiencer ?event17 John-1)
   (instance ?hamburger21 Food)
   (patient ?event17 ?hamburger21)
   (instance Wendys_Burger_Store Building)
   (located ?event17 Wendys_Burger_Store)))
```

The question *Who eats what?* would be translated by CELT to a query for the theorem prover:

```
(and
  (instance ?event Eating)
  (instance ?who Human)
  (agent ?event ?who)
  (instance ?what Physical)
  (patient ?event ?what))
```

where ?who and ?what are free variables to bind. The results are that ?who binds to John-1 and ?what to '?hamburger21' and CELT answers 'John eats a hamburger.' using an answer reply template that was constructed at the same time the question is parsed.

3 EXTENDING CELT TO NEW DOMAINS

Although many of the function words of CELT are limited, such as the quantifiers 'every' and 'some', or the determiners allowed, all of the content words can be extended: nouns, verbs, adjectives, and adverbs. In addition possessive phrases 'X of Y', two-place adjectives 'X is ___ Y', superlatives ('X is the ___est' or 'X is the most ___'), and prepositions can be extended ('X *verb* Y ____ Z').

Code generation templates for non one-to-one mappings can also be specified. In this way both the front-end (lexical input) and back-end (KIF code generation) translation of CELT can be extended.

3.1 Lexical Extensions

CELT lexical extensions are defined as facts describing the grammatical properties of that part of speech. For example, the template for a noun specifies the WordNet sense ID, the SUMO concept that it maps to[1], the noun's gender, and whether it is a mass noun or a count noun.

3.2 Ontological Extensions

The terms in the lexical templates should all be part of SUMO or an ontology built over it. For example, 'lieutenant colonel' is a term in the Army Reference Ontology, a mid-level ontology for use in army applications that specifies Army chain-of-command, weapons, etc.

3.3 KIF Code Generation Extensions

Lambda expressions can be used when mappings from terms are not one-to-one. Prolog does not have a built-in lambda operator but one can be simulated with a 'lambda' predicate and code for applying lambda expressions to arguments, as described in [3].

Domain-specific possessive constructs are handled in this way, with lambda expressions. For example, 'X of Y' translates to the KIF code (hasBodyPart X Y) when X is of SUMO type BodyPart and Y is of SUMO type Human.

The specific kinds of arguments for each lambda expression depend on the part of speech. For example,

[1] If there is not a one-to-one mapping then code generation templates are required, but this is the exception.

for adverbs, all lambda expressions are one-place and the argument that CELT provides is the event being modified.

4 RELATED WORK

4.1 Knowledge Representation and Ontologies

In CELT, different domains could be given different default preferences: an aerospace domain could prefer a 'fly by vehicle' word sense for 'fly' and an appropriate concept translation to the ontology, whereas a zoology domain could prefer the 'fly by flapping wings' word sense and concept translation. In either case CELT will generate a warning indicating the word sense chosen:

```
   Warning: interpreted the meaning of the verb
'travel' as  WordNet word sense #1 of 5 senses,
ID 201253107, meaning change location; move,
travel, or proceed; "How fast does your new car
go?" "We traveled from Rome to Naples by bus";
"The policemen went from door to door looking
for the suspect"; "The soldiers moved towards
the city in an attempt to take it before night
fell".  Maps to SUMO 'Motion'.
```

Such a warning can be used to indicate the need for post-editing changes. We intend these changes to again be targeted to WordNet meanings and their glosses so the user need not know the underlying ontological concepts or their names.

4.2 Natural Language

CELT is not intended purely as a Q/A system, such as FALCON [8], although that is one application to which CELT could be put. When used in this way, CELT answers questions from a knowledge base using a theorem prover. In contrast, FALCON accesses text from a database of documents. FALCON, and similar Q/A systems tested in TREC evaluations such as [17] are concerned with efficient question answering, information extraction, and document retrieval. CELT, on the other hand, is intended as a tool to assist knowledge engineers and subject matter experts.

CELT is also not a semantic grammar, like LADDER [9]. Instead it is a domain-independent tool with authorable domain-specific extensions. The *syntactic* grammar is domain-independent. It is not extensible by the user nor does the user have to compose grammar rules.

4.3 Controlled English

AECMA English [1] is an Aerospace industry standard for simplified English, intended to avoid ambiguity in aircraft maintenance manuals. Boeing Simplified English [2] is Boeing's implementation of simplified English, and has a checker that parses sentences to flag deviations from the standard, not to translate technical manuals into a formal knowledge representation. CELT also uses a simplified syntax but does not limit the parts of speech or the number of word senses a word can have. Nor is the number of words limited. Instead, a new vocabulary overlaid over the generic WordNet vocabulary can be provided for each new domain. More importantly, CELT is not a domain specific system. It is a completely general language which can be specialized and extended for particular domains.

Sowa advocates formal languages as a means of facilitating knowledge representation in [14]. For further discussion of ACE and other controlled English grammars and applications, see also [15].

5 CONCLUDING REMARKS

All grammars leak. No grammar can cover all of English, and CELT is no exception. Instead, it is intended to be simple enough that a user can predict whether a sentence or query is syntactically correct *before* submitting it to the interface. Furthermore, it is intended to simplify the correction of mistakes, whether these are from missing words in the lexicon, or from a heuristic wrong choice of a word sense. Overall, it is a tool to make knowledge entry more rapid, and to decrease the ontological expertise required by those who enter knowledge in very large knowledge bases.

Evaluation of CELT is important, and we are beginning to conduct experiments with CELT on the OpenMind [13] and Brown [5] corpora. While only a small percentage of the sentences in those corpora conform to the CELT grammar, they provide guidance on which grammatical features are common enough to warrant extension of the controlled grammar. In addition, for those sentences which do conform, they provide an independent test of the semantic interpretation of parsed sentences. CELT developers can check the logical output of the parsed sentences and see whether the logical form is a legitimate interpretation of the semantic content of the sentence.

REFERENCES

[1] AECMA Simplified English. See http://www.aecma.org/Publications.htm

[2] Boeing's Simplified English Checker. See http://www.lim.nl/monitor/boeing.html. From the Jan/Feb 1993 issue of Language Industry Monitor.

[3]. Michael A. Covington. (1993) Natural Language Processing for Prolog Programmers. Prentice Hall.

[4] Christiane Fellbaum (Editor). (1998). WordNet: An Electronic Lexical Database (Language, Speech, and Communication). MIT Press.

[5] Francis, W., and Kucera, H., (1964). Brown Corpus Manual. Revised 1979. Available at http://www.hit.uib.no/icame/brown/bcm.html

[6] N. E. Fuchs, U. Schwertel, R. Schwitter. 1999. Attempto Controlled English (ACE) Language Manual, Version 3.0, Technical Report 99.03, Department of Computer Science, University of Zurich, August 1999.

[7] Genesereth, M., (1991). ``Knowledge Interchange Format", In Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning, Allen, J., Fikes, R., Sandewall, E. (eds), Morgan Kaufman Publishers, pp 238-249.

[8] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. G. Erju, V. Rus, and P.Morarescu. FALCON: Boosting Knowledge for Answer Engines. NIST Special Publication 500-249: The Ninth Text Retrieval Conference (TREC 9), 479-488.

[9] Gary Hendrix. (1980) Mediating the views of databases and database users. Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling. June 1980. Available through ACM Digital Library, see http://portal.acm.org/.

[10] Kalman, J. K.(2001) Automated Reasoning with OTTER. Rinton Press.

[11] Hans Kamp, Uwe Reyle. (1993). From Discourse to Logic. Kluwer Academic Publishers.

[12] Niles, I., & Pease, A. (2001). Toward a Standard Upper Ontology, in Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). See also http://ontology.teknowledge.com.

[13] Singh, Push. (2002). The Open Mind Common Sense Project. Available: http://www.kurzweilai.net/meme/frame.html?main=/articles/art0371.html

[14] John F. Sowa. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Company.

[15] John F. Sowa. (2000). Controlled English.
See http://users.bestweb.net/~sowa/misc/ace.htm.

[16] Stickel, M., R. Waldinger, M. Lowry, T. Pressburger, and I. Underwood. Deductive composition of astronomical software from subroutine libraries. Proceedings of the Twelfth International Conference on Automated Deduction (CADE-12), Nancy, France, June 1994, 341-355. See also http://www.ai.sri.com/~stickel/snark.html.

[17] Voorhees, E. (2000). Overview of the TREC-9 Question Answering Track. NIST Special Publication 500-249: The Ninth Text Retrieval Conference (TREC 9), 71-80.