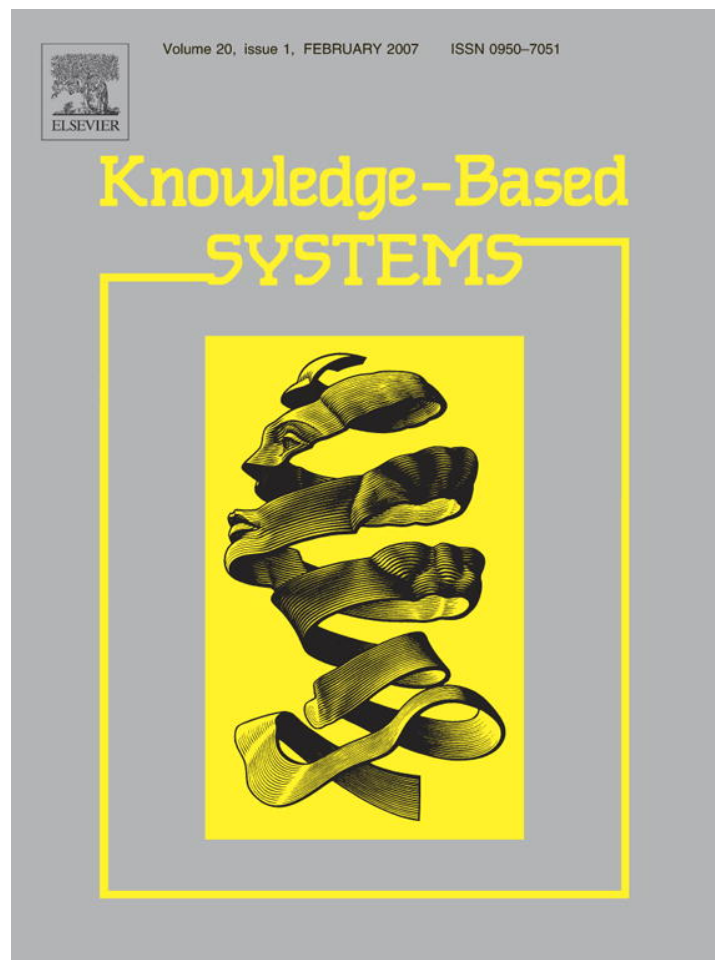


Provided for non-commercial research and educational use only.
Not for reproduction or distribution or commercial use.



This article was originally published in a journal published by Elsevier, and the attached copy is provided by Elsevier for the author's benefit and for the benefit of the author's institution, for non-commercial research and educational use including without limitation use in instruction at your institution, sending it to specific colleagues that you know, and providing a copy to your institution's administrator.

All other uses, reproduction and distribution, including without limitation commercial reprints, selling or licensing copies or access, or posting on open internet sites, your personal or institution's website or repository, are prohibited. For exceptions, permission may be sought for such use through Elsevier's permissions site at:

<http://www.elsevier.com/locate/permissionusematerial>

Semantic interpretation and knowledge extraction

Fernando Gomez ^{a,*}, Carlos Segami ^b

^a School of Computer Science, University of Central Florida, Orlando, FL 32816, United States

^b Department of Mathematics and Computer Science, Barry University, Miami Shores, FL 33161, United States

Received 21 June 2006; accepted 3 July 2006

Available online 28 July 2006

Abstract

A system that extracts knowledge from texts is presented. It is also indicated how the inferences necessary for the extraction of knowledge can be acquired by the system from sentences entered by users. The knowledge acquisition component is grounded on a semantic interpreter of English based on an enhanced WordNet. An evaluation of the system performance is included.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Semantic interpretation; Knowledge extraction; Inferences; WordNet

1. Introduction

The goal of this research is to build a knowledge base about a given topic by reading an encyclopedic article, or a Web site. Expert systems could use this database to tap in for pieces of knowledge, or a user could directly query the database for specific answers. Thus, two possible applications could be derived from our research: (a) the automatic construction of databases from texts for problem solvers and (b) querying text databases in natural language. There could be little doubt that a knowledge acquisition system (KAS) should be based on the output of a semantic interpreter. The more general the semantic interpreter the easier it should be to build different knowledge acquisition tasks for different domains. This paper describes a KAS that is fully based on the output of a semantic interpreter. We show that the inferences of the KAS are organized on the predicates used by the semantic interpreter to assign meaning to the grammatical relations of the sentence. Moreover, these inferences can be acquired by the system from natural language descriptions. The KAS uses the same ontology as that of the semantic interpreter. Because

the KAS is grounded on the semantic interpreter and shares the same ontology, the construction of different knowledge acquisition tasks reduces to telling the system some new inferences. The system builds the inference rules from the natural language descriptions of the users and stores them in the predicates used by the semantic interpreter. Incompatibilities between ontologies used by diverse components of the system do not exist because the KAS and the semantic interpreter share the same ontology. Furthermore, the user of the KAS does not have to be concerned with defining ontological categories, because these have been built for him/her in the semantic interpreter. This paper is organized as follows. Section 2 explains the semantic interpreter briefly. Section 3 explains the inferences based on the hierarchical organization of the predicates. Sections 4 and 5 describe the acquisition of inferences and the construction of the templates from the output of the semantic interpreter, respectively. Sections 6–8 provide the testing, related research and conclusions, respectively.

2. The semantic interpreter

One of the problems with semantic interpretation is the difficulty of achieving it on a large scale because of the need to craft complex inference and semantic interpretation rules. Recently, Mooney [14] has observed that much cur-

* Corresponding author.

E-mail addresses: gomez@cs.ucf.edu (F. Gomez), csegami@mail.barry.edu (C. Segami).

rent NLP research is more directly related to information retrieval than to language understanding, and that the overall trend in NLP can be concisely expressed by the phrase “scaling up by dumbing down.” For our KAS, we have used the semantic interpreter described in [4,5]. This semantic interpreter, which has been under construction for several years, is intended to overcome the problems discussed by Mooney by (a) anchoring the semantic interpretation process on a general ontology of English, namely WordNet 1.6 (henceforth, WN) [13], and by (b) defining predicates for WN verb classes [3] and linking them to grammatical relations and to the ontology for nouns. The WN verb classes and, to a lesser extent, the noun ontology have undergone considerable reorganization and redefinition following the criteria imposed by the semantic interpretation algorithm [6]. The interpretation algorithm, which is driven by the definition of these predicates, offers a solution to the following semantic interpretation problems: determination of the meaning of the verb, identification of semantic roles and adjuncts, and attachments of prepositional phrases (PPs). An interesting aspect of the algorithm is that the solution of all these problems is interdependent. As of this writing, we have defined 3017 predicates. This has resulted into a massive semantic knowledge-base linked to grammatical relations. The predicates in the semantic interpreter form a hierarchy in which semantic roles and inferences are inherited by subpredicates from their superpredicates. For instance, the predicate *graduate from* has the following superpredicates.

```
GRADUATE-FROM
  RECEIVE-AN-ACADEMIC-DEGREE
  GET-AWARD
  GET
  TRANSFER-OF-POSSESSION
  ACTION
```

The lexical definition of the predicates is a frame-like representation containing the selectional restrictions followed by the grammatical relations for that role given those selectional restrictions. The syntax for a semantic role is

```
(role (<slr> (<grs>
  (<slr> (<grs>
  .....
  (<slr> (<grs>)))
```

where <slr> stands for any number of selectional restrictions, and “<grs>” for any number of grammatical relations. The order in which the grammatical relations are listed is irrelevant. However, the order of the selectional restrictions is relevant in the sense that the first selectional restriction that matches is selected and the others in the list are not tried. Hence, the list of selectional restrictions is a preference list [18]. This preference list is critical in selecting the correct sense for many head nouns in the noun phrases of the verb arguments. A selectional restriction preceded by the

sign “-” (-slr) means that the semantic role is not realized by that ontological category. The entry for the predicate *graduate-from is*

```
[GRADUATE-FROM
 (IS-A (RECEIVE-AN-ACADEMIC-DEGREE))
 (WN-MAP (GRADUATE1))
 (AGENT (HUMAN) (SUBJ))
 (THEME (ACADEMIC-DEGREE) ((PREP WITH)))
 (FROM-POSS (EDUCATIONAL-INSTITUTION
 ORGANIZATION) ((PREP FROM))) ]
```

The entry *wn-map* means that all the synsets of *graduatel* and all the verbs that fall under the class of *graduatel* are mapped into the predicate *graduate-from*. The entry for the *theme is* intended to interpret sentences such as “X graduated with a degree in Physics from Y,” in which the *theme is* realized by [with NP] if the head noun of the NP is an *academic-degree*. The PP [from NP] matches the *from-poss* if the head noun of the NP is an *educational-institution*. This is the category preferred. However, if the head noun of the NP is not an *educational-institution*, but it is an *organization*, the *from-poss* role will also match. The default category, *organization*, is needed because some educational institutions are not part of the WN database, and this allows to automatically obtain them.

The semantic interpretation algorithm [4] is activated by the parser after parsing a clause. The parser does not resolve structural ambiguity, which is delayed until semantic interpretation. The goals of the algorithm are to select one predicate from the list of predicates for a verb form, attach PPs and identify semantic roles and adjuncts. For each grammatical relation (SR) in the clause and for every predicate in the list of predicates, the algorithm verifies if the predicate explains the SR. A predicate *explains* an SR if there is a semantic role in the predicate realized by the SR and the selectional restrictions of the semantic role subsume the ontological category of the head noun of the grammatical relation. This process is repeated for each SR in the clause and each predicate in the list of predicates. Then, the predicate that explains the most SRs is selected as the meaning of the verb. The semantic roles of the predicate have been identified as a result of this process. In case of ties, the predicate that has the greatest number of semantic roles realized is preferred. Every grammatical relation that has not been mapped into a semantic role must be an adjunct or an NP modifier. The entries for adjuncts are stored in the root node *action* and are inherited by all predicates. Adjuncts are recognized *after the meaning of the verb has been determined* because they are not part of the argument structure of the predicate.

3. Inferences based on the hierarchical organization of predicates

Assume that somebody wants to use our system to acquire knowledge about the schools attended by people

as students. The main relation to recognize is that of *attend school as a student*, which can be expressed in many ways. For instance, the text may say that person X entered school Y, that X was transferred to Y, that X graduated from Y, that X was educated at Y, that X received/got/obtained a degree from Y, that X studied at Y, that X was/became a student at Y, that X was an alumnus of Y, that X's parents sent X to Y, that X withdrew from Y, that Y accepted/admitted X, etc. Besides recognizing that all these verbs may imply *attend-a-school*, the algorithm must identify all semantic roles and adjuncts of the sentence and map them into the relation being acquired. For instance, if the sentence says that "X graduated from Y in 1943," the algorithm must recognize that "from Y" is the school attended by X and that "in 1943" is a temporal adjunct expressing the end-time in which X attended Y. This mapping should not be from grammatical relations for the verb "graduate" to semantic roles of *attend-a-school*, but from semantic roles for the predicate *graduate-from* to semantic roles for the predicate *attend-a-school*. There are several reasons for it, the most important being that other verbs besides "graduate" may express the relation "graduate-from," such as "X received/obtained/got a degree from Y." Another related reason is that the same semantic role may be realized by different grammatical relations.

The hierarchical organization of the predicates provided by the semantic interpreter does permit already to establish the inference *attend-a-school* for many verbs, because these are mapped into subpredicates of *attend-a-school*. For instance, the verb "enter" followed by a post-verbal NP whose head noun is an *educational-institution* is recognized by the interpreter as the predicate *enter-a-school* whose superconcepts are given by

```
ENTER-A-SCHOOL
ATTEND-A-SCHOOL
ATTEND-AN-EVENT-ORGANIZATION
INTERACT
ACTION
```

The verb "transfer" followed by [to NP], where the head noun of the NP is an *educational-institution* is recognized as *transfer-to-school* which is also a subpredicate of *attend-a-school*. In these cases, the builder of the KAS application has to do nothing since the predicate *attend-a-school* already exists. The hierarchies of predicates in the interpreter have been designed with the idea of maximizing the common sense inferences that can be established by inheritance, and of centering the inferences into a generic predicate rather than on individual senses of verb forms, which would lead to a proliferation of inference rules. However, inference rules connecting predicates will be needed as explained in the next section. These observations apply to every class of predicates constructed by the interpreter. For instance, if one wants to acquire knowledge about the things somebody values/respects/appreciates,

etc, the semantic interpreter already has the predicate *value-something* whose hierarchy is

```
VALUE-SOMETHING
RESPECT-VALUE-SOMETHING
JUDGE
ACTION
```

This predicate does not only include one of the senses of "value," but all WN verbs under the class *respect* (see below), plus *treasure*, *appreciate*, and one of the senses of "recognize."

```
respect, esteem, value, prize, prise
=> think the world of
=> reverence, fear, revere, venerate
=> enshrine, saint
=> worship
=> admire, look up to
```

If one wants to acquire a relation for each of the jobs somebody had, their location, time, and duration, the interpreter already provides a hierarchy of subpredicates of *work-be-employed*. For instance, the predicate *do-service*, encompassing such usages as "serve as ambassador/teacher/etc." has the hierarchy

```
DO-SERVICE
WORK-BE-EMPLOYEED
TRANSFER-OF-POSSESSION
ACTION
```

The examples are innumerable because the predicate classes cover most English verbs. The only thing needed in order to use the system for some acquisition task is the addition of some inference rules for some of the predicates relevant to the task, which we explain next.

4. Acquiring lateral common sense inferences

However, not all inferences can be established from the hierarchical organization of the predicates. The user needs to tell the system some inferences that link predicates to predicates "laterally," that is, not through the *is-a* relation. This is done by defining inference rules. These rules infer predicates and map semantic roles from the inferring predicate into roles of the inferred predicate. For instance, the predicates *graduate-from*, *study-a-subject*, and others are not classified as subpredicates of *attend-a-school*. However, that relation needs to be inferred if the sentence is "X graduated from Y," or "X studied at Y," where Y is an *educational-institution*. These inferences express common sense knowledge which are part of the meaning of these predicates. This is the knowledge that is between the lines when one reads, and, as a consequence, can not be extracted from texts. Rather, it is a necessary condition for under-

standing the texts [12]. The hierarchy of predicates for *receive-an-academic-degree* is

```
RECEIVE-AN-ACADEMIC-DEGREE
  GET-AWARD
    GET
      TRANSFER-OF-POSSESSION
        ACTION
```

The inferences can be acquired automatically in so far as the semantic interpreter builds a semantic interpretation for the sentence. The user does not need to have knowledge of the workings of the parser or the semantic interpreter. For instance, suppose that a user wants to acquire knowledge about the schools attended by somebody as a student. There are different ways in which a user may proceed. This is a systematic way.

- (a) First, the user needs to find out the predicate for “attend a school.” This can be determined by typing a sentence using those words, e.g., “Jennifer attended school.” Many other sentences are possible, e.g., “Carol attended Harvard University.” In fact, any sentence in which the head noun of the post-verbal NP is a subconcept of “educational institution” in the WN ontology would be fine. The system parses and interprets this sentence. The user can see the predicate constructed by the system for “attend school.” Let that predicate be *attend-a-school*. Then, the user finds all the predicates that infer *attend-a-school* by performing steps (b) and (c).
- (b) The subpredicates for *attend-a-school* can be found by typing “(find subpredicates attend-a-school).”
- (c) The lateral inferences of *attend-a-school* can be found by typing “(find-lateral-inferences attend-a-school).” This will provide the user with all predicates that infer the predicate *attend-a-school* laterally. The union of (b) and (c) provides all the predicates that infer *attend-a-school*. One can find out all the verb senses that are mapped to these predicates by typing “(wn predicate verb).”

Suppose that the predicate for the verb “enroll” in the sentence “Peirce enrolled at Harvard University” does not infer the predicate *attend-a-school*. The output of the semantic interpreter is

```
(CLAUSE CL25 (P ‘Peirce enrolled at Harvard University’ )
(SUBJ ((? PEIRCE)) ((PHILOSOPHER PEIRCE1)) (AGENT))
(VERB ENROLLED ((MAIN-VERB ENROLL ENROLLED)) ENROLL-AT-ORGANIZATION
  (ENROLL1) SUPPORTED BY 2 SRS)
(PREP AT
  (PREP-NP ((PN HARVARD UNIVERSITY))
    ((UNIVERSITY HARVARD_UNIVERSITY1)) (TO-ORGANIZATION))
  (ATTACH VERB CONFIDENCE STRONG))
```

The numbers correspond to WN sense numbers. The semantic role is listed at the end of the sublist for the grammatical relation, and the verb predicate follows immediately the verb. Now if one types “(inferences),” the system lists all inferences (lateral and hierarchical) for the predicate *enroll-at-organization*. Suppose that a user wants to add the lateral inference that if one enrolls at an educational institution, one attends that institution. The user only needs to type: *if a human enrolls at an educational institution, then that human attends that educational institution.*

The system responds by: *If human X1 enrolls at educational institution X2, then X1 attends X2.*

After this feedback from the system is received several times, it is easy to give names to the entities in the sentence, making them shorter and easier to write and understand. In the remainder of the paper, we will use names for the entities to ease understanding. The system parses the sentence and automatically builds the inference rule. The inference rule is stored in the predicate *enroll-at-organization*, which infers *attend-a-school* and maps semantic roles from *enroll-at-organization* to *attend-a-school* as follows:

```
((if% x-is-a $at-organization educational-
al-institution)
  (add-inference
    ((pr (attend-a-school))
      (agent ($agent))
      (to-loc ($at-organization))))))
```

The rule says that if the *at-organization* role of the predicate *enroll-at-organization* is a subconcept of *educational-institution*, then infer the predicate *attend-a-school* with *agent* the *agent* of the predicate *enroll-at-organization*, and *to-loc* the *at-organization* role of *enroll-at-organization*. The system does not specify in the if-part of the rule that the agent must also be a subconcept of *human*, because the predicate *enroll-at-organization* requires its agent to be a human. In general, the syntax for role mapping in the inference rules is

```
(<role> (<$role>))
```

where *<role>* is the role in the predicate being inferred, and *<\$role>* is the role of the predicate in which the inference rule is anchored. The user can now add the other inference rules needed in the same way, that is, by typing them, e.g., “If human X1 is transferred to educational institution X2, then X1 attends X2,” “If human X1 receives a degree from an educational institution X2, then X1 attends X2,” etc.

For a second example to indicate how a user could proceed, suppose that she wants to acquire from the encyclopedia the medicines prescribed for a disease, say pneumonia. Besides using the verb “prescribe,” this can be expressed by saying “Pneumonia is treated/cured with ...” and in other ways. WN can help in finding out the relevant verbs. One just needs to type the class of verbs under the synset *treat, care for*. WN lists many verbs in this class that are not relevant to the acquisition task at hand. But, it does identify most of the pertinent ones. Some verbs are not listed because their meaning is not directly related to *treat, care for*. Three verbs not listed are “prescribe,” “recommend,” and “receive.” Of these, “prescribe” is the most relevant to the task, and “receive” the least. But, one may find sentences such as “Pneumonia patients receive antibiotics.” “Receive” is the reciprocal of “give,” which is listed by WN, e.g., “Physicians give antibiotics to pneumonia patients.” Now, the user may start by typing some sentences with the verbs “cure” and “treat,” and see the output of the interpreter. If she types the sentences “Physicians treat/cure pneumonia with antibiotics,” she can see that the predicates for the sentences with the verb “treat” and “cure” are *to-treat-a-disease* and *cure-a-disease*, respectively. Both of these predicates have *medicate* as superpredicate. Now, she can find out all the relevant verbs that have any of these predicates as superpredicates by typing (wn predicate verb). Three of the verbs for which none of their senses are mapped to any predicate that has *medicate* as a superpredicate are “prescribe,” “recommend,” and “receive.” Then, she/he can provide the inferences for those three verbs as has been explained.

4.1. Detecting over-specification

People tend to over-specify the ontological categories in the rules. For instance, a user may type the following:

R1. *If human X1 enrolls at university X2, then X1 attends X2.*

The *problem* with this rule is that “university” is too specific. The system will miss inferring that “Kennedy attended Riverdale High School” from “Kennedy enrolled at Riverdale High School,” because “High School” is not a subconcept of “university.” The acquisition system can detect these cases of overspecification by comparing the ontological category of the semantic roles of the predicate in the if-part of the rule against the ontological categories of the definition of the predicate in the semantic interpreter. The role corresponding to the PP (at university) has *educational-institution* as its selectional restriction, which is a superconcept of “university,” “high school,” “academy,” “conservatory,” etc. As a consequence, the system may ask the user if she wants to draw the inference when the human attends any of those institutions, and, as a result, if she would like to change rule R1 to rule R2.

R2. *If human X1 enrolls at educational institution X2, then X1 attends X2.*

The cases of over-specification may occur in more than one semantic role of the rule, e.g., “If human X1 majors in

mathematics at university X2, X1 attends X2,” in which the discipline in which X1 majors, and the educational institution are both overspecified.

Another form of over-specification occurs if a rule is defined in subpredicate, say *p1*, when it should be defined in one of the superpredicates of *p1*. For instance, if a user defines the inference “If human X1 graduates from educational institution X2, X1 attends X2.” The inference is stored in the predicate *graduate-from*, which is a subpredicate of *receive-an-academic-degree*. This definition does not cause any problems in establishing an inference, but it makes it necessary to define the same rule in the predicate *receive-an-academic-degree*, when one rule under this predicate is the only one needed. Here again, the hierarchies of predicates minimize the need of inference rules because these are inherited by subpredicates from superpredicates. The lateral inference rules can be viewed as a semantic network of relations (verb predicates) connected by conditional links. Besides connecting the predicates, the network maps semantic roles from predicate *p1* into roles of predicate *p2*. Predicate *p2* may connect to other predicates, or infer other predicates as you prefer to express it, by means of these conditional links, resulting into a chain of relations. Predicate *p1* may infer *p2*, and vice versa. That is to say the link connecting *p1* to *p2* may be bidirectional. For instance, one may want to infer that “X attended Y” from “X studied at Y,” and that “X studied at Y” from “X attended Y.” In fact, one can define an inference rule on *attend-a-school* that infers *study-a-subject* and vice versa.

4.2. The algorithm for firing the rules

The algorithm that fires the rules is not caught in a circularity because it keeps track of all predicates that have been inferred. The algorithm is

Let A be the interpretation structure built by the interpreter. Initialize the list Exclude to nil. Initialize the list Inferences to nil. After applying this algorithm, this list will contain the inferences obtained from the predicate in A.

- (1) Let *pr* be the predicate in A. Add *pr* to the list Exclude.
- (2) Let I be the list of structures obtained from firing the inference rules associated with *pr*.
- (3) For each structure *a1* in I If the predicate of *a1* is not in the list Exclude,
 - (a) Add *a1* to the list Inferences.
 - (b) Repeat 1, 2, and 3 with A replaced by *a1*.

4.3. Mapping temporal adjuncts

Temporal adjuncts are not relevant if one is acquiring knowledge about drugs prescribed for certain diseases, causes of those diseases, imports and exports by countries, etc. But, if the acquisition task consists of events, the map-

ping of temporal adjuncts may be relevant for the acquisition task. For instance, suppose that somebody wants to acquire the *beginning-time* and *end-time* in the relation *attend-a-school*, meaning the time in which somebody starts attending a school, and the time in which one ends attending the school. The *beginning-time* can be expressed by sentences such as “X1 enrolled at/entered school X2 in 1889”. The *end-time* can be expressed by “He received a degree/graduated from school X2 in 1889.” Both temporal relations can be expressed by aspectual verbs such as “finish,” “begin,” “terminate,” etc. Thus, from the sentence “He graduated in 1890” one wants to infer that the *end-time* of *attend-a-school* is 1890, and from the sentence “He entered Harvard in 1886” the *beginning-time* of *attend-a-school* is 1886. These temporal relations can be obtained by expressing them in English as the following

R3. *If human X1 receives an academic degree from educational institution X2 in 1890, X1 attends X2 up to 1890.*

R4. *If human X1 enters educational institution X2 in 1893, X1 attends X2 from 1893.*

All temporal adjuncts are generalized by the acquisition system. One does not need to express rule R3 by saying: “If human X1 receives an academic degree from educational institution X2 in time X3, X1 attends X2 up to time X3.” The inference constructed for rule R3 is

```
((if x-is-a from-poss educational-
institution)
  (add-inference
    ((pr (attend-a-school))
      (agent ($agent))
      (to-loc ($from-poss))
      (end-time ($at-time))))))
```

The inference for rule R4 contains the mapping (*beginning-time*(\$at-time)).

5. From the interpreter output to the templates

We have tested the system to acquire knowledge about the educational institutions attended by people. The final output of the knowledge extraction system is a list of templates. In the application we are describing, the system pro-

duces one template for each educational institution attended. Each template consists of a number of slots, which are pre-determined by the user, and which vary from application to application. In the *attend-a-school* application, the template is defined as follows:

```
[ (ATTENDED ())
  (VERBAL-CONCEPT ())
  (LOCATION ())
  (TIME ())
  (FROM-TIME ())
  (END-TIME ())
  (ENTER-AT-AGE ())
  (GRADUATE-AT-AGE ())
  (SUBJECT ()) ]
```

The *attended* slot indicates the institution attended, *verbal-concept* is the verbal concept identified by the interpreter for the sentence, *location* is the location of the institution, *time* indicates when the institution was attended (e.g., “he attended Harvard in 1950.”), *from-time* and *end-time* are the starting time and end time of attendance, *enter-at-age* and *graduate-at-age* indicate attendance in terms of the age of the individual (“he entered/graduated from Harvard at the age of 20”), and *subject* refers to the field of study.

Each new sentence read by the knowledge extraction system is parsed and interpreted, and the interpreter output is then transformed into a set of knowledge representation structures. It is this set of knowledge representation structures that the knowledge extraction system uses to build the templates. As an illustration, if the sentence “John Kennedy attended elementary schools in Brookline and Riverdale” is read, the parser produces the output:

```
(SUBJ ((PN JOHN KENNEDY))
  VERB ((MAIN-VERB ATTEND ATTENDED) (TENSE
  SP))
  OBJ ((ADJ ELEMENTARY) (NOUN SCHOOLS))
  PREP (AND ((IN ((PN BROOKLINE)))) ((IN ((PN
  RIVERDALE))))))
```

The parser output becomes the input to the interpreter, which produces the following interpretation:

```
(CLAUSE CL1
(SUBJ ((PN JOHN KENNEDY))
  ((PRESIDENT_OF_THE UNITED STATES1 JOHN_KENNEDY)) (AGENT))
(VERB ATTENDED ((MAIN-VERB ATTEND ATTENDED)) ATTEND-A-SCHOOL
  (ATTEND1) SUPPORTED BY 2 SRS)
(OBJ ((ADJ ELEMENTARY) (NOUN SCHOOLS))
  ((GRADE_SCHOOL1 ELEMENTARY_SCHOOL1)) (TO-LOC))
(PREP AND
  (IN (PREP-NP ((PN BROOKLINE)) ((LOCATION BROOKLINE)) (AT-LOC))
    (ATTACH VERB CONFIDENCE WEAK))
  (IN (PREP-NP ((PN RIVERDALE)) ((LOCATION RIVERDALE)) (AT-LOC))
    (ATTACH VERB CONFIDENCE WEAK)))
)
```

The verbal concept is identified as *attend-a-school*. The *agent* of the action is *Kennedy*, a subconcept of *person*. The *to-loc* role is *elementary school*, a subconcept of *grade-school*. Brookline and Riverdale are subconcepts of *location* and fill the *at-loc* semantic role. The interpreter output is then transformed into the following set of knowledge representation structures:

```

RIVERDALE
  (instance-of (location))
  (related-to (@a9) (@a10) (@all))
  (cname ('riverdale'))
BROOKLINE
  (instance-of (location))
  (related-to (@a9) (@a10) (@all))
  (cname ('brookline'))
ELEMENTARY_SCHOOL1
  (is-a (grade_school1))
  (related-to (@a9) (@a10) (@all))
  (cname ('elementary_schools'))
JOHN_FITZGERALD_KENNEDY
  (is-a (person))
  (attend-a-school ($null ($more (@a9) (@all))))
  (study-a-subject ($null ($more (@a10))))
  (cname ('john_kennedy'))
@A9
  (args (john_fitzgerald_kennedy) (elementary_school1) (brookline)
    (riverdale))
  (pr (attend-a-school))
  (agent (john_fitzgerald_kennedy (q (constant))))
  (to-loc (elementary_school1 (q (constant))))
  (at-loc (brookline (q (constant))) (riverdale (q (constant))))
  (instance-of (action))
  (time (past))
ATTEND-A-SCHOOL
  (related-to (@a9) (@all))
@A10
  (instance-of (inference (@a9)))
  (pr (study-a-subject))
  (agent (john_fitzgerald_kennedy (q (constant))))
  (args (john_fitzgerald_kennedy) (elementary_school1) (brookline)
    (riverdale))
  (at-educational-institution (elementary_school1 (q (constant))))
  (at-loc (brookline (q (constant))) (riverdale (q (constant))))
  (time (past))
STUDY-A-SUBJECT
  (related-to (@a10))

```

sentence, as is the case of @A10, above. The verbal concept of each a-structure is examined (*pr* slot). If the verbal concept is *attend-a-school* or a subconcept of *attend-a-school*, then the slots of the a-structure are mapped into the slots of the template. The way to carry out this mapping must be defined by the application developer. In the *attend-a-school* application, the mapping is defined as follows:

Of these structures, the only ones of interest are the *a-structures*, those identified by the symbols @Axx. A-structures represent either the action described by the sentence, such as @A9 above, or an inference obtained from the

TEMPLATE SLOT
attended
verbal-concept
location

A-STRUCTURE SLOT
to-loc
pr
at-loc

time	at-time
graduate-at-age	graduate-at-age
enter-at-age	enter-at-age
from-time	from-time
end-time	end-time
subject	subject-of-study

Thus, in our example, the template produced is

```
JOHN_FITZGERALD_KENNEDY
  (ATTENDED (ELEMENTARY_SCHOOL1))
  (VERBAL-CONCEPT (ATTEND-A-SCHOOL))
  (LOCATION (BROOKLINE) (RIVERDALE))
```

Once the template is obtained, it must be integrated with the other templates already obtained from previous sentences. Several basic criteria are followed. First we determine if the slots of the new template are part of an existing template. In this case, nothing new is being added, so the template is discarded. Next, if the template does not contain the *attended* slot, then we append the slots of the new template to the previous template created. This handles cases like, “John attended Harvard. He majored in Physics.” Next, if the *attended* slot in the new template is the same as the *attended* slot of an existing template, then we append the slots of the new template to the existing template. Otherwise, we add the new template to the list of templates. After reading all the relevant sentences, the output produced for the article on Kennedy is

```
JOHN_FITZGERALD_KENNEDY
  (ATTEND-A-SCHOOL (ELEMENTARY_SCHOOL1))
    (LOCATION (BROOKLINE) (RIVERDALE))
  (ATTEND-A-SCHOOL (CANTERBURY_SCHOOL))
    (LOCATION (NEW_MILFORD))
    (TIME (1930))
  (TRANSFER-TO-SCHOOL (CHOATE_ACADEMY))
    (LOCATION (WALLINGFORD))
    (TIME (NEXT_YEAR))
    (FROM-TIME (NEXT_YEAR))
    (GRADUATE-AT-AGE (18))
    (END-TIME (1935))
  (ATTEND-A-SCHOOL (PRINCETON_UNIVERSITY1))
    (FROM-TIME (THAT_FALL))
  (ENTER-A-SCHOOL (HARVARD_UNIVERSITY1))
    (TIME (1936))
    (FROM-TIME (1936))
    (SUBJECT (GOVERNMENT1) (INTERNATIONAL_RELATION))
  (ATTEND-A-SCHOOL (STANFORD_UNIVERSITY_GRADUATE_BUSINESS_SCHOOL))
    (FROM-TIME (THEN1))
```

Some time references still need to be solved, such as *next-year*, *that-fall*, and *then*. This can be done by accessing the other entries in the frame. However, these

temporal references have not been implemented. These results show the system’s high degree of precision. Recall is also high, having missed Kennedy’s age when entering Canterbury School, the end time for Princeton, the graduation date for Harvard, and the end time for Stanford.

6. Testing

In order to test the system, we selected 50 articles at random from over 5000 biographical articles in The World Book Encyclopedia (World Book, Inc., Chicago. 1987) For each article selected, the template built by a human was compared with the template built by the system. We counted the number of slots in the template that were filled correctly by the system, the number that were filled incorrectly, and the number that were missed. We let C be the total number of correct slots for all articles, I the total number of incorrect slots, and M the total number of missed slots. Then, the measure of recall is given by $C/(C + I + M)$, and the measure of precision is $C/(C + I)$. The results obtained were 87% recall and 97% precision.

Many of the articles selected contained only two or three sentences relevant to the task. This is just the nature of the biographical articles in the World Book Encyclopedia (World Book, Inc., Chicago. 1987). A lower number of articles contained between 4 and 10 relevant sentences, and a few others more than 10. The system gets very few incorrect slots, and therefore very high precision, because of the accuracy of the semantic interpreter. The system fails

to interpret some adverbial clauses with an elliptical verb, e.g., “After a few months at Oxford University, Brummell was left ...” This is a problem that has recurred several

times, and which we plan to solve in a general way. In the Carter and Eisenhower articles, the system fails to infer that “to receive an appointment to the US Naval Academy” means to be admitted to the US Naval Academy as student, e.g., “In 1942, . . . Carter received an appointment to the US Naval Academy.” Other failures are due to some discourse problems, which in general are not acute in the Encyclopedia. We use a centering model [7] with specific knowledge based on the rhetorical structure of the encyclopedic articles. In the sentence, “When Dutch was 9 years old, he and . . . settled in Dixon, Ill, where the boy finished high school,” the system does not resolve the definite reference “the boy.”

7. Related research

This work is related to that described in [8] in which the acquisition of knowledge is closely connected to the semantic interpretation process. A paper that deals with the issue of inferences using WN is [9]. The authors implement a marker propagation algorithm that uses the verb entailment, the glosses and the concept hierarchy in WN. As the authors observe, the lack of semantic relations for the verbs and the few number of entailments that WN provides are some of the serious limitations with their approach.

There have been several systems in relation to the MUC project [15] that extract patterns from texts. These systems rely on the user to identify the relevant patterns, or on annotated corpora. None of these systems approach the semantic interpretation of complete sentences. For instance, in [1] a system is described that uses WN for knowledge extraction. The user identifies the patterns of interest and the system uses WN for the generalization process. Riloff [16] generates extraction patterns from annotated texts. Other systems require pre-constructed templates [2]. However, a semi-automated system that does not require annotated texts is [17] that constructs a domain lexicon by using a bootstrapping algorithm that starts with a set of *seed* words, and adds new words belonging to a semantic category. The enhanced list of seed words is then reviewed by a human who selects the words that should be added to the domain lexicon from those proposed by the algorithm. This system may be very useful for building lexicons for specialized domains, but not for acquiring knowledge from encyclopedic texts dealing with general domain knowledge. Moreover, because the system does not address the issues of semantic interpretation in a general context, its scope of applications will be limited to the extraction of some well-defined patterns. Similar remarks apply to the work on acquiring hyponyms from patterns that originated in [10]. This work does not assign meaning to the constituents of the sentence.

This work also differs from work reported in [11] in that the knowledge acquisition designer does not have to be concerned with defining ontological categories, or semantic interpretation rules because they are already part of the semantic interpreter. Moreover, the ontological categories,

namely those of WordNet, are of a general nature and have received a wide acceptance in the natural language processing community.

As Mooney [14] has indicated, most machine learning methods on information extraction acquire low-level syntactic patterns, and they do not deal with the problem of semantic interpretation of complete sentences. An exception is CHILL [19] that uses machine learning methods to build interfaces to databases. The system is trained on a corpus consisting of several hundred queries, and the semantic knowledge learned by CHILL is domain dependent.

A critique that can be leveled against our approach could be that it needs the hand-crafted construction of verb predicates, which is a rather difficult and time-consuming job. The reply to this is that once the verb predicates are defined, they are defined for every natural language application. This is so because their definitions are not tied to any given application, and their selectional restrictions are based on a general ontology of English. Moreover, as we have indicated in this paper the semantic interpreter can be used for automatically acquiring inference rules to be used by the knowledge acquisition system. In our forthcoming work, we are using the semantic interpreter to acquire concept definitions from natural language sentences. These definitions will be applied to noun sense disambiguation and other semantic tasks that cannot be solved by the definition of the predicates. Thus, the semantic interpreter will be used as a bootstrapping mechanism for improving itself.

8. Conclusions

We have described an approach to knowledge acquisition that is based on a semantic interpreter of English designed to achieve semantic interpretation on a large scale. The system also can acquire some inferences needed for understanding and acquisition from English sentences entered by users. The system acquires some inference rules because it already knows a lot in the form of a general noun ontology that has been reorganized for semantic interpretation and a rich set of predicates linked to the ontology of nouns and to grammatical relations. As of this writing, we have defined 3017 predicates and mapped 95% of WordNet 1.6 verb classes into predicates. Because the knowledge acquisition system and the semantic interpreter share the same ontology, the definition of new acquisition tasks is a natural extension of the semantic interpreter. The system has been tested on biographical articles taken from the The World Book Encyclopedia (World Book, Inc., Chicago. 1987) producing very solid results.

References

- [1] J. Yue Chain, A. Bagga, A.W. Biermann, The role of WordNet in the creation of a trainable message understanding system, in: Proceedings of AAAI-97, Menlo Park, CA, 1997, pp. 79–85.

- [2] M.E. Calif, R.J. Mooney, Relational learning of pattern-match rules for information extraction, in: *Proceedings of the ACL Workshop on Natural Language Learning*, Stanford, California, 1997, pp. 9–15.
- [3] C. Fellbaum, A semantic network of English verbs, in: C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database and Some of its Applications*, MIT Press, Cambridge, Mass, 1998, pp. 69–104.
- [4] F. Gomez, An algorithm for aspects of semantic interpretation using an enhanced WordNet, in: *Proceedings of the 2nd North American Meeting of the North American Association for Computational Linguistics, NAACL-2001*, 2001, pp. 87–94.
- [5] F. Gomez, Building verb predicates: a computational view, in: *Proceedings of the 22nd Meeting of the Association for Computational Linguistics, ACL-04*, Barcelona, Spain, 2004, pp. 351–358.
- [6] F. Gomez, Grounding the ontology on the semantic interpretation algorithm, In: *Proceedings of the Second International WordNet Conference*, Masaryk University, Brno, 2004, pp. 124–129.
- [7] B.J. Grosz, A.K. Joshi, S. Weinstein, Centering: a framework for modelling the local coherence of discourse, *Computational Linguistics* 21 (2) (1995) 201–225.
- [8] U. Hahn, K. Schnattinger, A text understander that learns, in: *COLING-ACL*, Montreal, Quebec, 1998, pp. 476–482.
- [9] S. Harabagiu, D. Moldovan, Knowledge processing on extended WordNet, in: C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database and Some of its Applications*, MIT Press, Cambridge, Mass, 1998, pp. 379–405.
- [10] M.A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: *Proceedings of COLING-92*, Nantes, France, 1992, pp. 530–545.
- [11] R.D. Hull, F. Gomez, Automatic acquisition of biographic knowledge from encyclopedic texts, *Expert Systems with Applications* 16 (1999) 261–270.
- [12] D.B. Lenat, R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*, Addison-Wesley Publishing Company, Reading, Mass, 1989.
- [13] George Miller, Nouns in WordNet, in: C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database and Some of its Applications*, MIT Press, Cambridge, Mass, 1998, pp. 23–46.
- [14] R.H. Mooney, Learning for semantic interpretation: scaling up without dumbing down, in: J. Cussens, S. Dzeroski (Eds.), *Learning Language in Logic*, Springer Verlag, Berlin, 2000, pp. 57–66.
- [15] MUC-4. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, Morgan Kaufmann, San Mateo, California, 1992.
- [16] E. Riloff, An empirical study of automated dictionary construction for information extraction in three domains, *Artificial Intelligence* 85 (1996) 101–134.
- [17] E. Riloff, M. Schmelzenbach, An empirical approach to conceptual case frame acquisition, in: *Proceedings of the Sixth Workshop on Very Large Corpora*, 1998, pp. 49–56.
- [18] Y.A. Wilks, Preference semantics, in: E.L. Keenan (Ed.), *Formal Semantics of Natural Language*, Cambridge University Press, Cambridge, UK, 1975.
- [19] J.M. Zelle, R.J. Mooney, Learning semantic grammars with constructive inductive logic programming, in: *Proceedings of AAAI-93*, 1993, pp. 817–822.