Minimal Commitment in Determining the Most Cohesive Interpretation Based on Non-Monotonic Inheritance Networks

Dan Tufis and Octav Popescu

Natural Language Processing Laboratory
Research Institute for Informatics
8-10 Averescu Avenue,
71316 Bucharest
ROMANIA

Keywords: pragmatics, knowledge representation

Dr. **Dan Tufis** is a senior researcher and head of Natural Language Processing Laboratory at the Research Institute for Informatics in Bucharest. Born in 1954 in Bucharest,Romania, he graduated from the Polytechnical Institute of Bucharest in 1979. He received his Doctor Degree in Computer Science from the Department of Computer Science of the Polytechnical Institute of Bucharest. in 1992. His main interests are computational linguistics, artificial intelligence perspective of natural language processing, machine translation. He is a member of several scientific bodies (ECACL, GLDV, EAMT,DECUS,AOSR) and authored more than 50 papers, published in national and international journals and in Proceedings of national and international conferences as well. In 1989 he was awarded the "Traian Vuia" Prize of the Romanian Academy.

Octav Popescu received his B.Sc. degree from the Polytechnical Institute of Bucharest in 1986. His present position is senior research worker at Natural Language Processing , Machine Learning and Conceptual Modelling Laboratory, the Research Institute for Informatics in Bucharest. His topics of interest include natural language processing, knowledge representation, reasoning and software development. He is co-author of several papers published in Proceedings of top international conferences. He is a member of ECACL, EAMT and DECUS.

1. Introduction

Inheritance networks are becoming more and more popular among Natural Language Processing community. This is not only because of a long tradition in the network formations in NLP but also because of the insights which inheritance technology can take into language processing. However, most of the benefits of inheritance networks have been looked for at the level of lexicon organization and morphological processing, grammar definition and syntactic processing and partly at the level of semantic interpretation [CL, 1992]. Obviously, there is no reason why inheritance networks are to be kept apart from the pragmatic sphere. The paper describes an algorithm with direct relevance to the pragmatic interpretation of natural language based on world-knowledge represented by means of multiple inheritance semantic networks.

2. Motivation

In domain-based understanding of natural language, the semantic representation of the parsed input (which is supposed to be independent of the application) is usually coerced into a domain-specific interpretation. This coercion (which might be seen as the pragmatic phase of the input interpretation) ensures the pruning out of spurious interpretations as well as the recovery of some missing information in case of extragrammatical input (ellipses, unknown words, metonimies, etc.). For instance, an input question like "Did the Boston office call?" might require, in order to be properly answered, to be expanded into something like "Did anybody, working at the office located in Boston, call us?". Suppose that the domain knowledge base contains (among other things) information saying that a phone calling event has a person as agent (the caller).and that the persons are working in several places, including offices, that offices are located in different areas, including towns, and so on. Expanding the metonimy, as our example comes at revealing an explicit relationship between a town (Boston) and an office, between a phone-calling event and an office caller, and between a phone-calling event and the addressee.

In our IURES system (an environment for developing NL question/answering systems) [Tufis and Popescu, 1991], a very effective way of dealing with such issues was proposed. It was based on the notion of **cohesion of a relationship** between the concepts of a **multiple inheritance semantic network** modelling the universe of discourse. In case of incomplete information, the interpretation of a sentence is done so that the resulting meaning representation should be the most cohesive subnet of the semantic network modelling the universe of discourse.

Although the solution to the problem was thought within a specific context, the work reported here aims at offering a quite general solution to the question of finding out the most cohesive dependency between any two concepts of a non-monotonic multiple inheritance semantic network. In spite of its being a major problem of inheritance networks, as far as we know, no algorithm has been announced yet.

In the next section, we will define precisely the notions of **multiple inheritance semantic network, non-monotonic inheritance, maximal cohesion dependency** and **minimal commitment dependency**. The final section will present our proposed algorithm for finding with minimal commitment a maximal cohesion dependency between the two concepts of a MISN. Comments guiding the algorithm understanding will also be made.

3. Knowledge Representation

Let MISN = <C, R, L, '<', AKO> be a multiple inheritance semantic network where $C = \{c_1, c_2, \dots c_k\}$ is a set of concepts, $R = \{r_1, r_2, \dots, r_j\}$ a set of binary conceptual relations, $L = \{<c_i\ r\ c_j> \in CxRxC\}$, the set of links defining the graph underlying the network, '<' is a partial order relation defined on C, and $AKO = \{< c_i\ c_j > \in C \times C\}$ a set of *generalization/specialization* pairs of concepts. Both concepts and conceptual relations are assigned unique interpretations.

When $<c_1 c_2> \in$ AKO, c_1 is said to be a *specialization* of c_2 or, conversely, c_2 is said to be a *generalization* of c_1. For $<c_1 c_2> \in$ AKO, $c_1 < c_2$ is always true (the reciprocal does not hold).

If c_k is a specialization of ci and $<c_i\ r\ c_j> \in L$ we say that $<c_k\ r\ c_j>$ is a specialization of $<c_i\ r\ c_j>$ on its left side. Similarly, if c_m is a specialization of cj and $<c_i\ r\ c_j> \in L$, we say that $<c_i\ r\ c_m>$ is a specialization of $<c_i\ r\ c_j>$ on its right side. Obviously, a link could be specialized on both sides (see Figure 1).

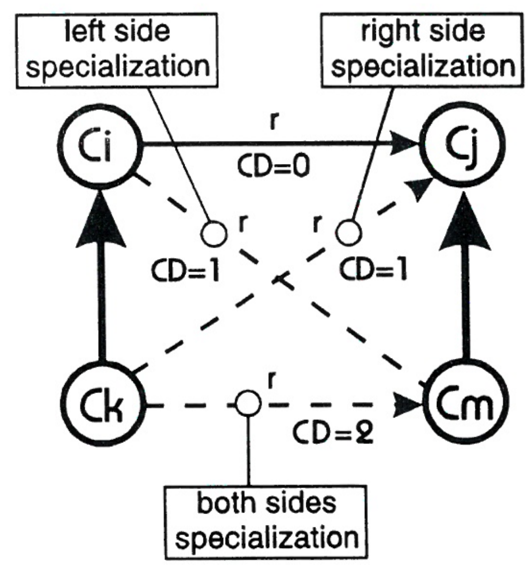

Figure 1. Links Specialization

The specialization/generalization of a link can also be viewed in terms of properties inheritance. In this respect, a link $<c_i\ r\ c_j>$ is used to represent the fact that c_i has the property r with the value c_j, or conversely, that c_j has the property r^{-1} with the value c_i. Asserting that c_k inherits from c_m is to assume that both the direct and inverse properties of c_i are valid, if not otherwise stated, for c_k too.

In case that a specialized link of the above type belongs to L, we talk about an *explicit specialization*. Otherwise it is an *implicit specialization*.

There is a great difference between the explicit and implicit links. The explicit links play a definitional role for the conceptual relations, enforcing restrictions on the pairs of concepts which might be related by them. The implicit links represent possible pairs observing the restrictions. While the former is the mandatory, the latter is the possibility.

In accordance with the semantic network consistency requirement, any explicit specialization of a given link blocks assuming some (otherwise assumable) links (see Figure 2).

This *shadowing* effect of specializing a link conforms with the principles of *default reasoning* [McCarthy, 1987].

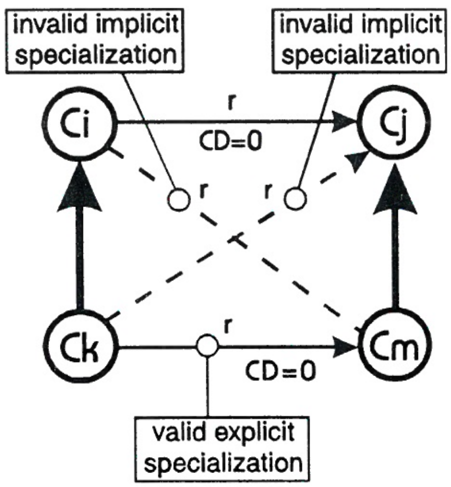

Figure 2. Valid and Invalid Specializations

A *dependency* in MISN between c_s and c_t is defined as a finite concatenation of binary conceptual relations $[r_1\ r_2 \ldots r_k]$ appearing in a sequence of links $< c_s r_1\ c_1 >\ < c_1\ r_2\ c_2 > \ldots < c_{k-1}\ r_k\ c_t >$, every link in the sequence being either in L or a valid specialization of another link in L. The number of binary relations appearing in a dependency is called the *length* of that dependency (LN).

Finding a dependency between two concepts, which comes to identifying a relationship between them, could be regarded as a fundamental type of non-monotonic reasoning. The non-monotonic character is given by the notion of *shadowing inheritance*. A similar position is convincingly defended in [Horty et al 1990].

If two given concepts c_s and c_t, present more than one dependency, that dependency evincing a minimal number of binary relations is said to be *the most cohesive dependency* between c_s and c_t.

Let us now formally define our notions of multiple inheritance semantic network and *valid dependencies* in such a network. We will do it by using a first-order predicate calculus notation, and by postulating a set of axioms to be interpreted within a natural deduction framework.

Let **ako** be the predicate denoting a specialization/generalization relation between concepts as defined by the AKO set;

e_link be the predicate denoting an explicit link in a semantic network as defined by L;

$<$ be the predicate denoting the partial order of a MISN;

link be the predicate denoting an arbitrary link (either implicit or explicit) in a semantic network;

dep be the predicate denoting a dependency between two concepts in a semantic network.

The properties of the predicates above are granted by the following prenex form axioms:

ako axiom

(ak1) $\forall c_i, c_j \left[\mathbf{ako}\left(c_i, c_j\right) \rightarrow \left(c_i < c_j\right)\right]$

axioms of $<$

(po1) $\forall c_i\ [\ (c_i < c_j)$

(po2) $\forall c_i, c_j\ [\ (c_i < c_j) \rightarrow \neg\ (c_j < c_i)]$

(po3) $\forall c_i, c_j, c_k\ [(c_i < c_j) \wedge (c_j < c_k) \rightarrow (c_i < c_k)]$

network consistency restrictions

(cr1) $\forall c_i, c_j, c_m, r[\mathbf{e\_link}\,(c_i,r,c_j) \rightarrow \neg \mathbf{e\_link}\,(c_i,r,c_m])$

(cr2) $\forall c_i, c_j, c_k, r[\mathbf{e\_link}\,(c_i,r,c_j) \rightarrow \neg \mathbf{e\_link}\,(c_k,r,c_j)]$

(cr3) $\forall c_i, c_j, c_k, c_m, r$

$[\mathbf{e\_link}\ (c_i,r,c_j \wedge \mathbf{e\_link}\ (c_k,r,c_m) \rightarrow [(c_k<c_i) \leftrightarrow (c_m<c_j)] \wedge [(c_i<c_k) \leftrightarrow (c_j<c_m)]]$

(cr4) $\forall c_i,c_j,c_k,c_m,r, \exists c_p,c_q\ [\mathbf{e\_link}\ (c_i,r,c_j) \wedge \mathbf{e\_link}\ (c_k,r,c_m) \wedge \neg\ [(c_k<c_i) \vee (c_m<c_j)] \wedge \neg\ [(c_i<c_k) \vee (c_j<c_m)] \rightarrow [\mathbf{e\_link}(c_p,r,c_q) \wedge (c_i<c_p) \wedge (c_j<c_q) \wedge (c_k<c_p) \wedge (c_m<c_q)]]$

specialization axioms

(sp1) $\forall c_i,c_j,r\ [\mathbf{e\_link}\,(c_i,r,c_j) \rightarrow \mathbf{link}\,(c_i,r,c_j)]$

(sp2) $\forall c_i,c_j,c_k,c_m,r\ [\mathbf{link}\,(c_i,r,c_j) \wedge (c_k<c_i) \wedge \neg \mathbf{e\_link}\,(c_k,r,c_m) \rightarrow \mathbf{link}\,(c_k,r,c_j)]$

(sp3) $\forall c_i,c_j,c_k,c_m,r\ [\mathbf{link}\ (c_i,r,c_j) \wedge (c_m<c_j) \wedge \neg \mathbf{e\_link}\,(c_k,r,c_m) \rightarrow \mathbf{link}\,(c_i,r,c_m)]$

dependency axioms

(dp1) $\forall c_i,c_j,r\ [\mathbf{link}\,(c_i,r,c_j) \rightarrow \mathbf{dep}(c_i,\text{append}\,(\text{NIL},r), c_j)]$

(dp2) $\forall\ c_i,c_j,c_k,p,r$ [**dep** (c_i,p,c_j) Λ **link** (c_j,r,c_k) $\rightarrow$**dep** $(c_i$,append (p,r), $c_k)$]

A dependency between c_i and c_j is valid iff **dep**$(c_i$, p, $c_j)$ is a theorem.

We have seen that an implicit link $<c_k,r,c_m>$ of some specific concepts c_k and c_m can be inferred on the basis of an explicit link $<c_i,r,c_j>$ existing between an ancestor of c_k and an ancestor of c_m. This is possible by a proof consisting in repeatedly applying the axioms (sp2) and (sp3).

Let us call the number of steps in such a proof the *commitment degree* (CD) of the inferred implicit link. This figure expresses the number of presuppositions made on the existence of a link between two given concepts. Obviously, for an explicit link, its commitment degree is 0 (see Figures 1 & 2).

We extend the notion of commitment degree by applying it to a dependency $<c_i\ [r_1\ r_2 \dots r_n]\ c_j>$ by defining it to be the sum of CDs of the links determining the dependency.

We have proved elsewhere [Tufis & Popescu, 1992] that, due to non-monotonicity of inheritance reasoning, new information acquisition might invalidate some previously valid dependencies. Naturally, this does not apply to a dependency having a zero CD. Although not at all obvious, it is quite intuitive that the chance of a dependency for being withdrawn as an effect of knowledge acquisition is better as its CD is higher.

Now we are ready to provide a pragmatic motivation for the algorithm discussed in the sequel. When the system is faced with two concepts but not explicitly told which their relationship is, it has to make a rational guess. If the world knowledge allows that multiple dependencies between them to exist, it seems to be psychologically motivated to choose the dependencies with fewer link traversals (**cohesion principle**). On the other hand, among several most cohesive dependencies, it seems natural to prefer the most robust one (**minimal commitment principle**).

In order that such an approach should be effective, the knowledge base is to be organized according to the same principles.

4. The Algorithm

The task to be solved can be formally stated as follows: being given two concepts, a source c_s and a target c_t, do find that dependency between them? $d = [r_1\ r_2 \dots r_k]$, so that its length (LN) and commitment degree (CD) are both minimal.

In studying the algorithm below, consider the following notational conventions:

- Let $D_{i,j}$ denote a cluster of dependencies between cs and any arbitrary $?c_k$ of the MISN, each of the dependencies having the length i and the commitment degree j:
  $D_{i,j} = \{<c_s\ ?d\ ?c_k> \mid LN(d)=i \& CD(d)=j\}$;
- By clustering the sets $D_{i,j}$ on the length of their dependencies, one gets the sets:
  $D_k = \{D_{i,j} \mid i=k \& j \geq 0\}$;
  For the sake of the algorithm, the elements $D_{i,j}$ of any D_k should be considered as sorted according to the increasing values of j (recall that j represents the CD of any dependency in a $D_{i,j}$ set).
- We will also use the following notations:
  $D_{[k \dots m]} = \{D_i \mid k \leq i \leq m\}$;
  $D_{[k \dots]} = \{D_i \mid i \geq k\}$
  As in the case of D_k, the above sets are, this time, to be considered as sorted, according to the increasing values of i (the LN of the dependencies).

The main procedure describes a Dijkstra (1959) type algorithm for finding out the shortest path between two nodes in the graph underlying the MISN.

```
procedure find_dependency (C_s; C_t) is
D_{0,0} ← {<c_s[]c_s{}visited>}
for each D_{0,j} ≠ {} in D_0 do
   D_{1,j} ← extend (D_{0,j})
   D_{0,-1} ← generalize (D_{0,j})
end for;
for each D_i ≠ {} in D_[1...] do
    for each D_ij ≠ {} in D_i do
      if exist <c_s ?d c_t ?SH visited > in D_ij then
         return d;
      end if;
   D_{i,j-1} ← D_{i,j+1} ∪ specialize (D_{i,j});
   D_{i,j-1} ← extend (D_{i,j});
   D_{i,j-1} ← D_{i,j+1} ∪ generalize (D_{i,j});
  end for;
 end for;
end procedure;
```

The next procedure extends each dependency in the given $D_{i,j}$ by all possible conceptual relations not shadowed by explicit specializations (axioms (sp1) and dp2)), starting from its final concept.

```
procedure extend (D_i,j) is
E ← {};
  for each <?c_s ?d ?c_k ?SH ?mark>
  in D_i,j do
    for each <c_k ?r ?c_m> in L do
      if not r in SH and
      status (c_m; D[0...i]) ≠ visited then
        E ← E ∪ {<c_s [d |r] c_k {} visited>};
      end if;
    end for;
  end for;
  return E;
end procedure;
```

The two following procedures replace the last concept of each dependency in $D_{i,j}$ by all its generalizations/specializations respectively, thus implementing the process of left/right specialization of the conceptual relations involved (axioms (sp2) and (sp3)).

```
procedure generalize (D_i,j) is
G ← {};
for each <?c_s ?d ?c_k ?SH ?mark> in D_i,j do
R ← {};
for each <c_k ?r ?c_m> in L do
R ← R ∪ {r};
end for;
for each <c_k ?c_i> in AKO do
if status (c_i; D_[o...i-1] ∪ {D_i,[o...j]}) ≠visited then
G ← G ∪ {<c_s d c_i R generalized>};
end if;
end for;
end for;
return G;
end procedure;
```

```
procedure specialize (D_i,j)is
S ← {};
for each <?c_s [?d |?r ] ?c_j ?SH ?mark>
in D_i,j do
 if mark ≠ generalized then
 for each <?c_m c_j> in AKO do
  if not exist <?c_k r c_m> in L and
  status (c_m; D_[0...i-1] ∪ {D_i,[o...j]}) ≠ visited then
    S ← S ∪ {<c_s [d | r] c_m {} visited> };
   end if;
  end for;
 end if;
 end for;
 return S;
end procedure;
```

The procedure 'status' checks out whether the given concept has been previously reached on any of the dependencies built up so far. It helps avoid looking for dependencies which have sequences of relations in common with shorter ones.

```
procedure status (c_i; H) is
status ← notvisited ;
for each D_i in H do
 for each  D_i,j in D_i do
  if exist <?c_s ?d ?c_i ?SH ?mark> in D_i,j then
     if mark = visited then
       return visited;
      else
         status ← mark;
      end if;
    end if;
   end for;
  end for;
 return status;
end procedure;
```

REFERENCES

CL . **Computational Linguistics. Special Issues on Inheritance**. Vol. 18, Nos.2-3, 1992.

DIJKSTRA, E.W., **A Note on Two Problems in Connection with Graphs**, NUMERICAL MATHEMATICS, No. 1, 1959, pp. 269-271.

HORTY, J.F., THOMASON, R.H.and TOURETZKY, D.S., **A Skeptical Theory of Inheritance in Non-monotonic Semantic**

Networks, ARTIFICIAL INTELLIGENCE, Vol. 42, Nos. 2-3, 1990, pp. 311-348.

MCCARTHY, J., **A Form of Non-monotonic Reasoning**, in M. Ginsberg (Ed.) Readings in Non-monotonic Reasoning, MORGAN KAUFMANN, 1987, pp. 145-152.

TUFIS, D.and POPESCU, O., **IURES-2-Natural Language Environment and the Computer Speak Paradigm**, Proceedings of ICYCS'91, 1991, pp. 409-413.

TUFIS, D. and POPESCU, O., **A Fast Algorithm for the Shortest Path Problem in Multiple Inheritance Semantic Networks**, STUDIES IN INFORMATICS AND CONTROL, Vol.1, No.2,1992, pp.99-112.