Anaphora Resolution Involving Interactive Knowledge Acquisition

Rolf Schwitter

Centre for Language Technology, Macquarie University
Sydney 2109 NSW, Australia
Rolf.Schwitter@mq.edu.au

Abstract. Anaphora resolution in current computer-processable controlled natural languages relies mainly on syntactic information, accessibility constraints and the distance of the anaphoric expression to its antecedent. This design decision has the advantage that a text can be processed automatically without any additional ontological knowledge, but it has the disadvantage that the author is severely restricted in using anaphoric expressions while writing a text. I will argue that we can allow for a wider range of anaphoric expressions if we consider the anaphora resolution process as an interactive, machine-guided knowledge acquisition process in cases where no suitable antecedent can be found automatically. This knowledge acquisition process relies on the human author who provides additional terminological information for the anaphora resolution algorithm – if required – while a text is written.

1 Introduction

Computer-processable controlled natural languages are **engineered** subsets of natural languages designed to reduce ambiguity and vagueness that is inherent in full natural language [5, 10]. These controlled natural languages **look like** English but are in fact **formal** languages that can be translated unambiguously into the input language of an automated reasoner and be used for several reasoning tasks, among them for question answering. Similar to full natural language, these controlled natural languages allow for anaphoric expressions but their form and usage are considerably restricted. An anaphoric expression (= anaphor) is a word or a phrase that points back to an expression (= antecedent) that has been previously introduced in the text (see [8] for an overview). The two most important types of anaphoric expressions that are used in sentences and between sentences in controlled natural languages are pronominal anaphora and definite noun phrase anaphora. Definite noun phrase anaphora take the form of definite descriptions and proper names. Computer-processable controlled natural languages use relatively "safe" anaphora resolution algorithms that rely mainly on syntactic information, accessibility constraints and the distance of the anaphor to its antecedent. This makes it easy for the machine to resolve anaphoric expressions automatically but difficult for the human author to remember the approved forms of anaphoric expressions. In the following discussion, I will focus on definite descriptions, in particular on bridging definite descriptions [9], and discuss how these anaphoric expressions that require inference and inference-supporting knowledge can be handled in a controlled natural language context.

2 Definite Descriptions

In the simplest case, a definite description that is used anaphorically matches syntactically with its antecedent. For example, the anaphor in (2) matches with the noun phrase antecedent in (1):

1. An academic staff who teaches COMP448 in E6A owns a laptop computer.
2. *The academic staff* supervises Robert Black and leads the LT Centre.

However, the relation between an anaphor and its antecedent is often more complex than that of identity. The relation may be a synonymy relation as in (3), a hypernymy relation as in (4), a hyponymy relation as in (5), or a meronymy relation as in (6):

3. *The faculty member* supervises ...
4. *The staff* supervises ...
5. *The professor* supervises ...
6. *The hard disk* is broken and does not spin.

These bridging definite descriptions point back to a noun phrase antecedent that has already been introduced in (1) but they are characterised by a different head noun (3, 5, 6) or by a constituent – in our case a head noun (4) – that forms a part of the entire antecedent. The resolution of the anaphoric expressions in (3, 5, 6) requires additional ontological knowledge specified in a knowledge base and some reasoning while the anaphor in (4) can be resolved in principle on the syntactic level. Note that the definite descriptions in (2-5) refer to the same entity as the noun phrase antecedent in (1). But this is not the case in (6) where the referent of *the hard disk* is only "associated" with the laptop computer previously introduced in (1).

3 Knowledge Bases

On the one hand, WordNet [4] has been used as an approximation of a knowledge base for resolving bridging definite descriptions in unrestricted texts but this proved to be highly unreliable for the automatic identification of correct semantic relations [12]. The situation gets even worse if we work in a domain where a very specific vocabulary is required.

On the other hand, one can bite the bullet and construct a (linguistically motivated) formal ontology for a particular application domain that contains – among other things – the required terminological knowledge for resolving definite descriptions, for example terminological statements such as:

6. `(define-concept professor (and leader teacher supervisor))`
7. `(define-primitive-concept teacher academic_staff)`
8. `(define-concept teaching_event`
   `(and event (all has_agent teacher) (some has_theme course)))`
9. `(equivalent academic_staff faculty_member)`
10. `(define-primitive-concept laptop_computer`
    `(some has_direct_part hard_disk))`

Note that this terminological knowledge (here expressed in an expressive description logic [1] that follows the KRSS[1] notation) can be specified directly in a controlled natural language and then be translated automatically into the above target representation:

11. Every professor is defined as a leader and a teacher and a supervisor.
12. Every teacher is an academic staff.
13. Every teaching event is defined as an event that has only teachers as an agent and that has a course as a theme.
14. Every academic staff is equivalent to a faculty member.
15. Every laptop computer has a hard disk as a direct part.

In this scenario, the knowledge base can be updated if new factual information becomes available and the semantic relations can be checked automatically by querying the description logic knowledge base. The information in the knowledge base can even be used to guide the writing process in a predictive way (using similar techniques as in [7]) since all background information has been carefully specified in advance.

However, there exists another scenario where a domain expert might want to assert new factual information but the terminological knowledge is not yet available. For example, the domain expert might want to assert first the sentence (1) and then the sentence (3) but the correct resolution of the anaphor in (3) would require the terminological information in (14). This suggests an approach where the domain expert supports the anaphora resolution algorithm and specifies additional terminological knowledge while a text is written.

4 Anaphora Resolution in PENG Light

PENG Light is a computer-processable controlled natural language that can be used for knowledge representation [11]. The language processor of the PENG Light system translates texts incrementally into TPTP[2] notation. Since the unification-based grammar of PENG Light is bidirectional, the language processor can take a syntactically annotated TPTP formula of a sentence as input and produce an output string in controlled natural language. The grammar of PENG Light is written in a DCG-style notation and is processed by a chart parser. The TPTP notation is built up during the parsing process together with a paraphrase that illustrates how a sentence has been interpreted and how anaphoric expressions have been resolved. The grammar maintains a list of accessible noun phrase antecedents during the parsing process whereas accessibility is defined in a similar way as in Discourse Representation Theory [6].

The proposed anaphora resolution algorithm for definite descriptions extends the existing algorithm of PENG Light in a systematic way. The existing algorithm resolves an anaphorically used definite description with the most recent accessible noun phrase antecedent that matches fully or partially with the anaphor and agrees in number and gender with that anaphor. The new algorithm relies

[1] `http://www.bell-labs.com/user/pfps/papers/krss-spec.ps`
[2] `http://www.cs.miami.edu/~tptp/`

on interactivity and allows the author to specify semantic relations between a noun phrase antecedent and a bridging definite description if this information is not already available in the knowledge base. This solution is compatible with a predictive authoring approach since the anaphora resolution process is machine-guided and the author selects among a number of options.

In order to process bridging definite descriptions the controlled natural language processor of PENG Light communicates with an automated reasoning engine. We have experimented with E-KRHyper [3], a model generator and theorem prover for first-order logic, with CEL [2], a polynominal-time classifier, and with RacerPro [13], a description logic reasoning system. We currently use RacerPro for processing terminological information since it is the most advanced and versatile description logic system but we search for more expressive alternatives.

The following simplified DCG rule for definite descriptions illustrates how feature structures are used to deal with syntactic, semantic and pragmatic information in PENG Light, and shows that the anaphora resolution algorithm is triggered whenever a definite description has been processed:

```
n3(..., fol:LF, sco:C, para:P1-P4, ant:A1-A3) -->
  det(..., def:yes, fol:LF, res:R1-R3, sco:C, para:P1-P2, ana:[]-D1),
  n2(..., fol:R1-R2, ..., para:P2-P3, ant:A1-A2, ana:D1-D2),
  { anaphora_resolution(n2, R2-R3, A2-A3, D2, P3-P4, ...) }.
```

The anaphora resolution algorithm takes a partial logical formula (`R2`), a list of accessible noun phrase antecedents (`A2`), a definite description (`D2`) and the current paraphrase (`P3`) as input and returns an updated logical formula (`R3`), an updated list of antecedents (`A3`) and an updated paraphrase (`P4`) as output. If the definite description is an anaphoric expression, then this expression is replaced by the noun phrase antecedent and this replacement is marked in the output list (`P4`) of the paraphrase.

The anaphora resolution algorithms first checks if the definite description matches fully with the first of the accessible noun phrase antecedents in the input list `A2`. The antecedents are ordered and represented as terms that contain syntactic and semantic information, for example:

16. `object(X,academic_staff)#[third,sg,masc_fem]#[academic,staff]`

The matching is done over the syntactic information, and if no solution can be found, then the algorithm checks for a partial match. If the partial match succeeds, then the algorithm queries the terminological part of the knowledge base for a hypernymy relation using the description logic reasoner RacerPro. RacerPro's allows us to query this semantic relation in a direct way, for example:

17. `(concept-subsumes? staff academic_staff)`

If there is no partial match, then the algorithm checks sequentially for synonymy, hyponymy and meronymy relations (note that the optimal sequence is genre-specific but we don't know in advance what the author is going to specify):

18. `(concept-equivalent? academic_staff faculty_member)`
19. `(concept-subsumes? academic_staff professor)`
20. `(concept-subsumes? (some has_direct_part hard_disk) laptop_computer)`

The actual resolution is reflected in a paraphrase and the domain expert can accept or reject the solution. If no solution can be found, then the domain expert has to decide if the noun phrase is a discourse new definite description or semantically related to one of the accessible antecedents in `A1`. If the former is the case, then the domain expert simply accepts the new expression and a new discourse referent is introduced. If the latter is the case, then the domain expert has to specify the semantic relation between the antecedent and the anaphoric expression on the **interface** level by selecting the relevant antecedent and the corresponding semantic relation from a menu. Once this has been done, the knowledge base is updated and the new bridging definite description is licensed in the text.

5 Conclusions

I argued that anaphora resolution for definite descriptions in computer-processable controlled natural languages can be interpreted as an interactive knowledge acquisition process in those cases where no suitable noun phrase antecedent can be found by the machine. The presented approach relies on a domain expert who works in **collaboration** with the machine and who provides the required ontological knowledge while a text is written – this approach brings the human back into the loop.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook, Cambridge University Press (2003)
2. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL – A Polynomial-time Reasoner for Life Science Ontologies. In: Proc. of IJCAR'06, LNAI 4130, pp. 287–291 (2006)
3. Baumgartner, P., Furbach, U., Pelzer, B.: Hyper Tableaux with Equality. In: Proceedings of CADE-21, LNAI 4603, pp. 492–507 (2007)
4. Fellbaum, C.: WordNet, An Electronic Lexical Database, MIT Press (1998)
5. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Reasoning Web, LNCS 5224, pp. 104–124 (2008)
6. Kamp, H., Reyle, U.: From Discourse to Logic, Kluwer (1993)
7. Kuhn, T., Schwitter, R.: Writing Support for Controlled Natural Languages. In: Proceedings of ALTA 2008, Tasmania, Australia, pp. 46–54 (2008)
8. Mitkov, R.: Anaphora Resolution. In: Mitkov, R. (ed.), Oxford Handbook of Computational Linguistics, pp. 266–283, Oxford University Press (2003)
9. Poesio, M., Vieira, R., Teufel, S.: Resolving Bridging References in Unrestricted Text. In: Proceedings of ACL-97 Workshop on Operational Factors in Practical, Robust, Anaphora Resolution for Unrestricted Texts, 7-11 July, pp. 1–6 (1997)
10. Schwitter, R., Tilbrook, M.: Meaningful Web Annotations for Humans and Machines using CNL. In: Expert Systems, 25(3), pp. 253–267 (2008)
11. Schwitter, R.: Working for Two: a Bidirectional Grammar for a Controlled Natural Language. In: LNAI 5360, pp. 168–179 (2008)
12. Vieira, R., Poesio, M.: An Empirically-Based System for Processing Definite Descriptions. In: Computational Linguistics, 26(4), pp. 539–593, MIT Press (2000)
13. Wessel, M., Möller, R.: A Flexible DL-based Architecture for Deductive Information Systems. In: Proceedings of the FLoC'06 Workshop on Empirically Successful Computerized Reasoning, pp. 92–111 (2006)