CGExtract: Towards Extraction of Conceptual Graphs from Controlled English

Svetla Boytcheva[1], Pavlin Dobrev[2], and Galia Angelova[3]

[1] Department of Information Technology, FMI, Sofia University,
5 J. Bauchier Blvd., 1164 Sofia, Bulgaria,
svetla@fmi.uni-sofia.bg
[2] ProSyst Bulgaria,
p_dobrev@prosyst.bg
[3] Bulgarian Academy of Sciences, Linguistic Modeling Lab,
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria,
galia@lml.bas.bg

Abstract. Automatic extraction of formal knowledge specifications from Natural Language (NL) text is a challenging research and development area. Currently the task is considered feasible for restricted NL input only. A number of CG researchers approached the problem, applying Sowas algorithm for analysis of NL input by joins of relevant canonical graphs. This paper summarises the state of the art and describes the CGExtract prototype, which approaches the subject by integration of Parasite, an already existing component for NL analysis and Understanding (NLU). This powerful NLU machine produces logical form for each correct sentence and processes coreferences in extended discourse of several sentences. Given an initial type hierarchy and relevant lexicon information, CGExtract constructs new KB graphs corresponding to the input text, by (i) checking whether the input sentences represent one connected graph, (ii) proving the KB consistency, i.e. proving whether the new graph is in contradiction with the already existing KB graphs and (iii) proving whether the new graph yields loop definitions with already existing KB graphs. The CGWorld workbench [4] supports the user interface of CGExtract.

1 Introduction

Automatic extraction of formal knowledge specifications from NL text is a challenging research and development area. However, this enterprise still looks too hard, efforts consuming and very expensive. Automatic Knowledge Acquisition (KA) needs preliminary defined description of every (important) word expected in the input text, i.e. an input word can be recognised and correspondingly processed if and only if the system's lexicon contains *at least* some morphological information about that word. Initial (upper layer of the) type hierarchy and some semantic primitives are needed as well. In this way it turns out that automatic KA requires preliminary definitions of thousand words, efforts to provide relatively full text analysis and special attention to the consistency of knowledge,

which is acquired and stored in the knowledge base (KB). Due to all complications, currently the task of automatic KA is considered feasible for restricted NL input only.

A number of researchers focused on this subject in a number of projects dealing with extraction of Conceptual Graphs (CG) from restricted English. The input text of these prototype systems varies from single, isolated phrases and sentences to larger text collections dealing with discourse structures. The common approach is to analyse the restricted NL input using John Sowa's algorithms based on joins of predefined canonical graphs. We have to emphasize that this kind of semantic-driven NL analysis provides quick and efficient results for messages in somewhat telegraphic style and thus turned out to be feasible for e.g. knowledge extraction from medical reports about patients. As a contrast, this paper describes an attempt to exploit existing components for NL understanding in order to automatically construct a CG knowledge base from input text in restricted English.

A basic component in our CGExtract prototype is the system Parasite, developed by Allan Ramsay (University of Manchester, UK) [10]. Parasite performs the morphological, syntactic and semantic analysis of the input text of few English sentences and currently processes many kinds of coreferences in extended discourse. The system outputs detailed results of syntactic analysis combined with logical forms of sentences, which are determined to be semantically correct with respect to the given meaning postulates. Our elaboration CGExtract benefits from the linguistic diagnostic provided by Parasite (which concerns findings at all levels of NL analysis: morphology, syntax, semantics and discourse structure including recognition of discourse referents and their coreference). In this way, integrating Parasite enables us to define "controlled English" as "English accepted by Parasite". Such a powerful component allows for further development focusing on the proper KB problems, i.e. knowledge augmenting, structuring and consistency.

This paper is structured as follows. Section 2 discusses the notion of "controlled English" and overviews briefly some related research and prototype systems dealing with automatic acquisition of CG from English text. Section 3 presents problems for KA from free text, lists useful Parasite features and gives an example of Parasite output, which is the input for CGExtract. Section 4 considers CGExtract in more detail, focuses on the KB consistency problems, which we pretend to solve at present, and gives examples and evaluation. What we find innovative in our approach is the fact that building upon an existing, very elaborated NLU components, we process more complex sentences and even extended discourse and address problems related to KB consistency. Section 5 contains the conclusion and sketches future work.

2 Related research for controlled languages

Most of the research done for the acquisition of CG from text aims at the processing of somewhat realistic sentences and text specifications. However, full

text analysis is not easy to achieve and that is why all applications deal with some restricted NL, which is often called Controlled language (CL). A CL may be defined as a language designed for special purposes. Controlled languages range from artificial, programming languages with NL keywords (suitable for well-defined applications, but the lack of extensibility makes them difficult to adapt to new applications [15]) to subsets, which keep much of the NL richness. In general, however, (1) It is not clear how restricted a language should be to provide successful automatic KA, and moreover, there are no convincing practical applications defining the optimal restrictions, and (2) Ambitions, as well as existing unrestricted text sources, make us consider almost free NL as an ultimate goal.

What should be restricted in a CL, still preserving the essential part of its expressiveness? Some answers are given in the field of computational linguistics, which tried to define restricted languages for improving the quality of machine translation (see e.g. [5]). Other groups of scientists dealing with controlled languages defined restricted languages like Basic English, Esperanto, and recently the so-called Universal Language. In summary, the first important restriction concerns the vocabulary: much less words are allowed, with one meaning per word only, and few hundred predefined operators-verbs. Domain terms are to be additionally appended to the basic vocabulary. The rules are mainly advice on constructions that should be avoided, usually because they lead to ambiguity. The rules for controlled languages tend to be stylistic guidelines rather than hard and fast grammar specifications. .... The readability and clarity of a controlled language technical text often seems better that uncontrolled text, even for native English readers. ... It is not particularly difficult to train people to write controlled language texts, i.e. text which generally observes some set of fairly simple writing rules [5]. Examples of task-specific controlled languages used in companies like Xerox, British Aerospace, etc. show that dealing with some controlled NL is a promising choice in automatic KA from text [5].

The existing prototypes for acquisition of CG from text process de facto controlled English. It is worth noting that this kind of applications were very popular among the CG community in the 80-ies and 90-ies, after the publication of the idea how to perform NL-analysis using join ([16]) and two papers reporting the earlies implementation for English ([17, 18]). Today we can list dozens of prototypes developed for English, French, Italian and German languages in task-specific contexts (most references are omitted for brevity), and our perspective would be to overview the systems together with the restrictions they impose on the NL input.

One of the earliest rather developed prototype was DANTE, which analysed isolated complex Italian sentences [20]. This system is a serious effort to encode lexical semantics in a systematic way: it works with about 850 extended word-sense definitions. DANTE performs real morphological analysis . Its syntax analysis is provided by a grammar with app. 100 rules which covers about 80

Successful prototypes for analysis of medical texts were implemented in Hamburg University (METEXA, [14]) and Geneva University Hospital (RECIT, [13]).

Medical texts, esp. reports about patients, are characterized by rather telegraphic style, so the semantic structure is more important than the syntactic one when the systems analyse the NL input in order to translate it to CG. METEXA works with a corpus of 1500 radiological reports containing about 8000 different wordforms. It has a fullform lexicon, where the compound German terms are defined, and performs syntactic analysis of the input phrases. The semantic analysis works in parallel with the syntax analysis. METEXA translates the reports (consisting mostly of simple sentences and phrases) to CG and answers questions about the semantic representation of radiological reports. This system gives an important hint for treatment of semantic patterns in NL with cases. Another example of successful performance in the medical domain - RECIT - analyses sentences from medical texts in French, English and German and stores the sentence meanings into CGs. It works on free-text patient documents in digestive surgery and applies the so called proximity processing, which aims at the decomposition of the sentence into meaningful fragments, given a partial interpretation of the sentence. Thus RECITs analyser is a modular system, composed of two parts which are necessary to separate the language-independent module from the language-specific module. RECIT is not based on a formal grammar but on a set of sequential semantically-driven procedures which incrementally build a meaningful structure of the NL input. Typical medical expressions are recognised by frequent-association rules. Some syntactic ambiguities are resolved by local syntactic rules proper to each natural language. Minor changes are necessary for adding of a new natural language to the system [13].

As we see, the NL input restrictions of the abovementioned prototypes can be summarised as (i) limited vocabulary, (ii) processing of phrases and simple sentences, (iii) often missing syntactic analysis as a separate module. More recent prototypes have similar limitations. For instance, [8] treats limited vocabulary (which is naturally restricted by the domain) as well as simple sentences and imperative phrases. Knowledge Extractor [2] acquires graphs from NL fragments selected by the knowledge engineer who operates with the workbench in order to assure successful acquisition of the desired CG specifications from on-line available technical texts. (The paper [3] presents in more detail the NL analysis). CG Mars Lender [6, 7] skips unknown words in the input sentences exemplified in [6], which is a flexible strategy to define a controlled language. The designers of the workbench WebKB developed the so called Formalized English, providing acquisition of different ontological structures from text statements [9]. Although potentially different (due to the rather elaborated NLU kernel), our workbench CGExtract is restricted in a similar manner since it deals with restricted vocabulary, restricted grammar rules and restricted set of semantic primitives.

As a conclusion, we summarize typical features of the KA systems listed above: (i) extracting CG from text, they juxtapose labels to the acquired types according to the input words. For instance, from the sentence *Rupert Murdock owns Fox* CG Mars Lander acquires the concepts [Rupert Murdock], [own] and [Fox] (see [6]). (ii) Conceptual relations are chosen either according to the thematic roles assigned to verbs-events in the canonical graphs, or correspond to

a list of preliminary determined keywords yielding conceptual relations (e.g. prepositions are often considered as markers of conceptual relations). (iii) These systems have restricted capacity to acquire contexts and coreference links among instances. Our prototype CGExtract shares the same features and limitations: input words are mapped to type labels; conceptual relations appear according to strictly defined rules; and there is no acquisition of contexts. However, there are some essential improvements allowed by Parasite: (i) there is an elaborated module for syntactic analysis, which provides intermediate structures for recognition of discourse references among sentences and thus coreferences between instances are possible when Parasite recognizes coreference between corresponding referents in the input discourse; and (ii) there is a prover which check the semantic correctness of the input sentences against the available meaning postulates. It allows to treat the variability of the input text in a more consistent manner and provides innovative aspects of the analysis of the NL input paraphrases.

3 Processing linguistic phenomena in automatic KA

All systems for automatic KA from text have to perform at first lexical (morphological) and syntactic analysis of the input. For simplicity we assume here that there are no principal difficulties to perform the morphological and syntactic analysis and the so called tokenization; obviously there are practical difficulties for every particular application, but at least theoretically the basic paradigm of word and sentence analysis is already clearly established and well-known. Here we focus on the semantic analysis of the input text.

Paraphrases and variations are one of the basic problems in understanding the sentence and text semantics. Let us consider for instance the simple sentences 1.1-1.5, which encode the same statement. Sentences 1.1-1.4 contain the same meaningful words, have similar syntactic structure and therefore, in principle, have to cause automatic KA of the same conceptual graph:

(1.1) Newly issued securities are traded on primary market.

(1.2) All newly issued securities are traded on primary markets.

(1.3) Every newly issued security is traded on primary markets.

(1.4) A newly issued security is to be traded on primary market.

(1.5) Primary market operates with newly issued securities.

However, it would be rather difficult to prove that graphs acquired from (1.3) and the universal reading of (1.1) are equivalent. An even more complicated case is the graph corresponding to (1.5), which would contain a type [OPERATE], so proving its equivalence to e.g. (1.4) would not be trivial and would require relevant semantic definitions of OPERATE and TRADE. In this way, the "little" variations of the article, number, quantification (a, the, some, every, all, most, usually, often), the negation and other logical operators create a practically unrestricted set of "almost" equivalent sentences, which - according to the accepted KA practice - have to be roughly encoded as one statement.

Further examples from terminological dictionaries, written for human readers, convince us that automatic recognition of paraphrases in free text is almost

hopeless; understanding that definitions 2.1-2.3 are similar might be problematic for many human beings too.

(2.1) secondary market - financial market, trading with securities, already existing; usually a stock exchange

(2.2) secondary market - exchanges and over-the-counter markets where securities are bought and sold subsequent to original issuance, which took place in the primary market.

(2.3) secondary market - When stocks or bonds are traded or resold, they are said to be sold on the secondary market. The majority of all securities transactions takes place on the secondary market.

In this way, if we want to approach systematically the task of automatic KA, our first step is to define acceptable restrictions of the linguistic phenomena in the input text; this controlled English language should be on the one hand free enough to allow for expression of the facts the knowledge engineer wants to insert in the KB and, on the other hand, limited enough to avoid ambiguities and misunderstandings. Defining a controlled NL might be difficult since we have to explain the restrictions to the end users; but our solution is either to rely on the linguistic machine Parasite and its capacity for analysis and diagnostics, or to restrict the user to write and iteratively correct input text in especially designed text fields for acquisition of type hierarchy, type definitions and graphs where we impose constraints on the sentences appearing in each field (see section 4).

Parasite works using a lexicon, syntax grammar rules and a knowledge base of type (word) hierarchy and meaning postulates. The lexicon contains the morphological description of the words recognised in the input text. The grammar currently covers most of the English syntax, including complex embedded sentences. The hierarchy is a tree, but each type might participate more than once. The meaning postulates define in logical format the word semantics. It is not obligatory to define in advance the semantics of each word to be processed; the designer only has to keep in mind that the prover of the semantic correctness works with the available postulates. Parasite is an open system and allows for the insertion of new words, grammar rules and meaning postulates. When started Parasite checks the KB consistency (contradictions, loop definitions).

As a typical NLU artefact (in contrast to some prototypes for automatic KA), Parasite analyses every input string. It processes separate sentences as well as extended discourse of several sentences. Given a text paragraph, the user might choose analysis type: either independent analysis sentence by sentence, or analysis of all sentences as coherent discourse.

The analysis is performed step by step, starting by morphological and syntactic analysis. Diagnostics is available in cases of unknown or non-correctly derived words, as well as for wrong or ambiguous sentence structure. Soft parsing techniques provide correct analysis of sentences with "small" syntax errors (e.g. wrong subject-verb agreement). Some ambiguity types are resolved by heuristically predefined preference scores; currently the PP-attachment problems are tackled. Syntax analysis fails in case of unknown input words and unresolvable ambiguities. Then the user of CGExtract is offered the possibility to correct or

paraphrase the input and to continue working with the same text (more discussion in Section 4).

After correct syntax analysis Parasite performs semantic analysis (see [1, 11, 12]). Meaning postulates are encoded in a language which is a "dynamic, constructive version of Ray Turner's 'property theory': everyone else should just note that if you want computers to do anything about meaning you have to express it in a language which is (a) expressive enough to capture the richness of natural language and (b) formal enough for a computer to do something with it". There are not too many contenders" [11].

Example of Meaning postulate for "bond":

```
lexicalMP(
forall(X :: {bond(X)}, debt(X) &
               exists(Z::{capital(Z)}, theta(Z,$of,X) &
                  exists(Y::{devide(Y)}, theta(Y,$object,Z))))


Equivalent CG for "bond":
typedef BOND(X) is
[debt:x]->(of)->[capital]<-(obj)-<[devide]
```

4 CGExtract: a case study of KA

CGExtract assumes that its user is a knowledge engineer, since it provides options to choice among sophisticated KB structures. In this section we describe in more details the construction process and the KB consistency checks performed at present.

```
Inserted sentence:
```

A local government authority issues a municipal bond to pay for a community infrastructure project.

```
Dependency tree:

{issue,s}
-------------------------------------------------------------------------------------------
{authority,}                        to                 for                                {bo
----------------------------    ------    ------------------------------    ---------------
a {local,} {government,}  {pay,}              {project,}             a   {municipal,}
                                          ---------------------------------
                                       a {community,} {infrastructure,}

Model:
```

```
issue(#354).
  theta(#354, agent, #356).
  theta(#354, for, #357).
  theta(#354, object, #358).

government(#356, lambda(A, local(A, lambda(B, authority(B))))).

infrastructure(#357, lambda(A, community(A, lambda(B, project(B))))).

municipal(#358,
          lambda(A,
                 bond(A)
                 & purpose(A,
                           lambda(B,
                                  theta(#359, agent, B)
                                  & C . #359
                                    & pay(lambda(D, theta(#359, identity, D)))
                                      & lambda(D, theta(#359, identity, D))
                                        is event)))).
```

Figure 1. Input sentence, its dependency tree and model produced by Parasite.

Figure 1 displays Parasite output for one English sentence *A local government authority issues a municipal bond to pay for a community infrastructure project.* The logical form and the model are input for CGExtract, from where we start the construction of a CG KB. Note the thematic roles AGENT, OBJ, FOR, encoded by Parasite as theta-terms in the model as well as the PURPOSE-relation which links the embedded sentence 'to pay for a community infrastructure project' to the main predicate ISSUE. These roles yield conceptual relations for the corresponding conceptual graphs.

As we've already said, CGExtract does not deal with the proper NL analysis. Given (i) already defined initial type hierarchy and means for recognition of complex type labels in the text (i.e. INCOME_TAX is a concept label recognized from the consequent tokens income tax);

(ii) corresponding words in Parasite's lexicon and relevant amount and content of meaning postulates, and

(iii) input text of few English sentences,

CGExtract (i) constructs new types, type definitions and graphs (one graph per given input fragment) and (ii) proves KB consistency.

4.1 Acquisition of types, type definitions and graphs from English text

To distinguish between universal and existential readings of sentences, CGExtract supports separately the three cases of KB assertion, as shown at Fig. 2.

View and Update Type Labels allows for browsing of already asserted complex type label like FINANCIAL_INSTRUMENT, FINANCIAL_MARKET etc. as well as for editing of the initial type hierarchy. In this way the knowledge engineer might keep fresh his/her memory of existing KB labels.

Update Type Hierarchy opens an interface for automatic acquisition of new types from simple English sentences and their assertion in the KB hierarchy. Examples for input sentences are
(3.1) A subsidy is a financial instrument.
(3.2) A government is a financial institution.
The hierarchy is built top-down, i.e. a child is defined by IS_A relation to its parent.

Acquisition of Type Definitions is linked to an interface where the user types in English text. Although given in singular, all sentences are treated as universal statements (because the default is that the type definition encode statements valid for all instances). CGExtract considers all given sentences as one definition, i.e. Parasite processes the text as extended discourse. It is clear that definitions have to deal with already existing types; fortunately in our case missing types are diagnosticised as missing words by Parasite. Section 4.2 discusses checks of the semantic consistency of type definitions.

Acquisition of Conceptual Graphs opens an interface where the user types in English sentences to express existential statements, which are encoded as conceptual graphs. An example is given at Fig. 3a and 3b. The user enters free text and Acquisition of types, type definitions and graphs from English text and receives feedback after linguistic analysis. The idea is that after some experience and several iterations, the users will most probably learn the controlled English providing the desired KB insertions. Fig. 3a displays input of disconnected but correct sentences. Since the semantic analysis of the two sentences shows they are correct separate statements, two conceptual graphs are generated. However, the default is that CGExtract acquires one graph from each input fragment. CGExtract prompts the user to edit the input text in order to turn it to coherent discourse or to delete the redundant sentence. If the user corrects the input as shown at Fig. 3b, one connected graph will be acquired. Parasite recognises the coreference between 'municipal bond' in sentences one and two and CGExtract builds one instance of [MUNICIPAL_BOND] accordingly.

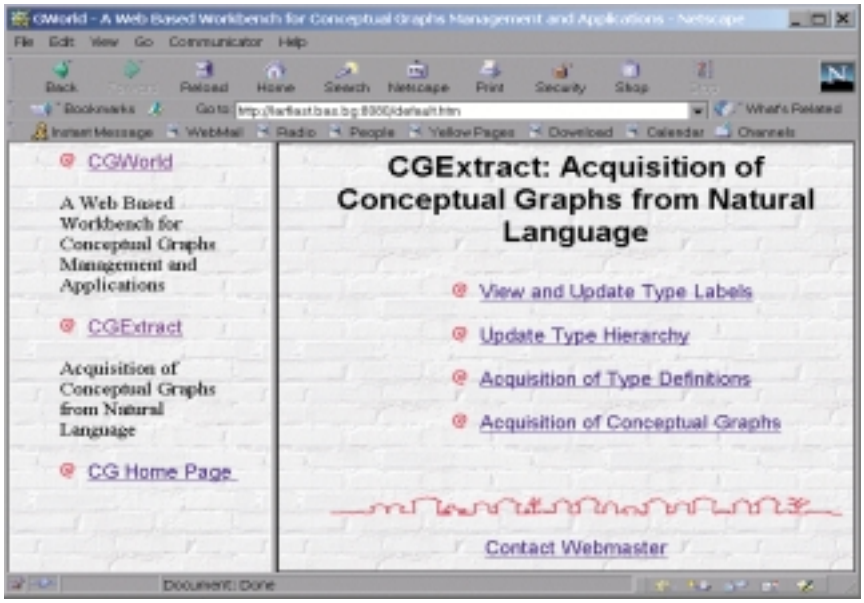

Fig. 2. The main CGExtract interface

4.2 Support of KB consistency

After acquisition of a new type definition or new graph, CGExtract tests the effects of its eventual KB assertion. Support of KB consistency involves the following tasks:

CGExtract checks whether the new generated definition/graph has a logical model under the available background knowledge, i.e. is it semantically correct and is the discourse coherent. This model is in fact an internal text representation, encoded as follows. First, the model places the concepts of the new graph/definition within the hierarchy of natural types; for instance, if [MUNICIPAL_BOND] appears in the new graph, then the model contains its parents [BOND], [DEBT_INSTRUMENT] and [FINANCIAL_INSTRUMENT] (these parents are not shown in Fig. 1 for brevety's sake).

Second, different objects have unique numbers (anchors) in the model; the properties that some of the objects possess are enumerated too, as well as the relations between them. Fig. 1 contains the unique numbers #356, #357, #358 etc. The system Parasite constructs this model (essentially a Herbrand model). After that CGExtract translates the model to Parasite's meaning postulate form and to CG Prolog format and proves whether the new graph contradicts to the already existing KB graphs.

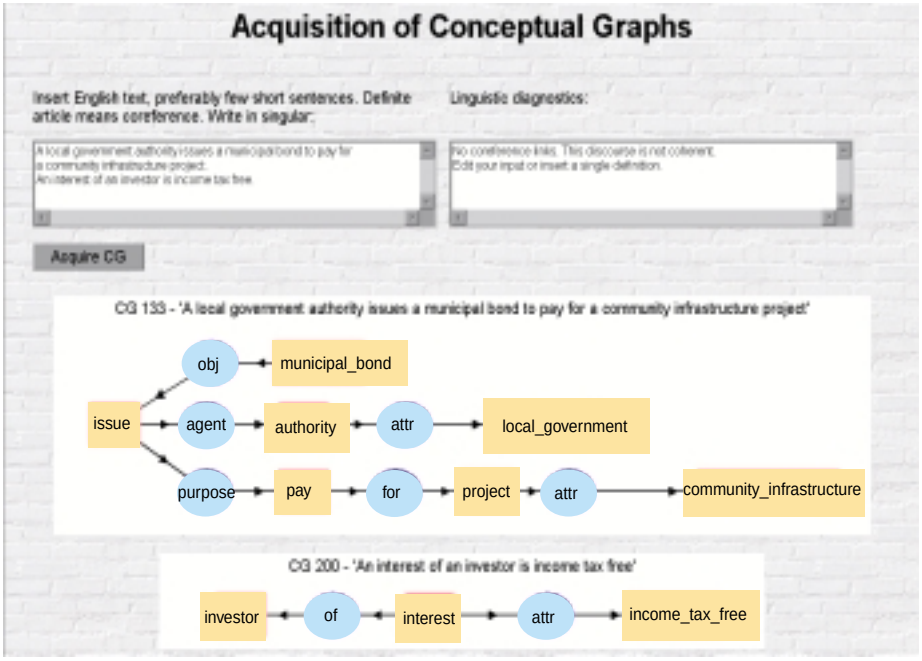

Fig. 3a. Non-coherent input to CGExtract

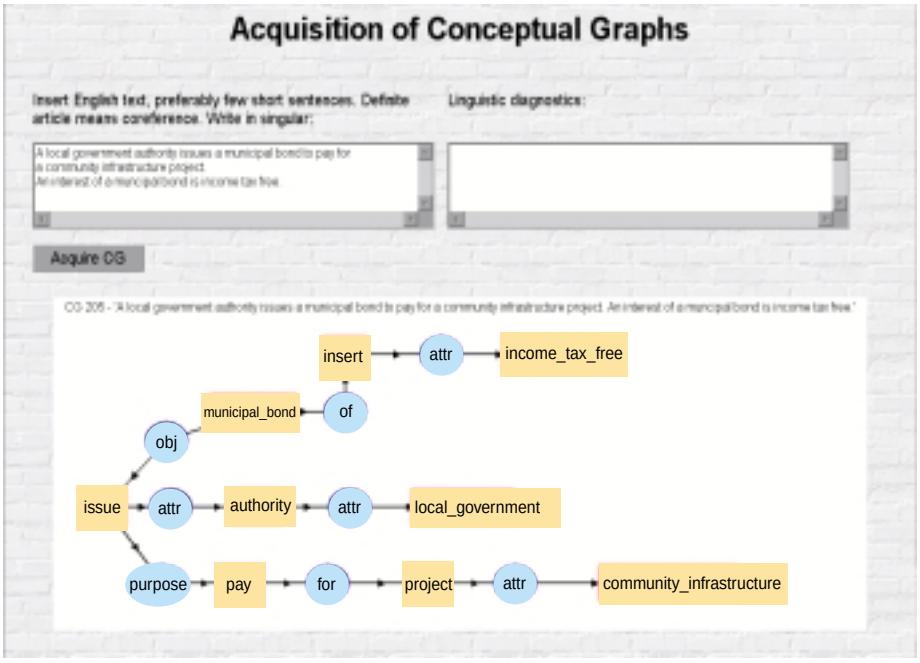

Fig. 3b. Building one connected graph from two input sentences

After constructing the model, Parasite checks whether the new graph or type definition yields loop definitions with already asserted KB statements. For instance, consecutive entry of the sentences:

(4.1) A municipal bond has an interest.

(4.2) If a municipal bond has an interest it is exempt from income taxes.

will cause indication of a loop for the second definition. This operation for checking consistency avoids multiple definitions of partially overlapping statements in the knowledge base.

Additional function tested at present is the insertion of new meaning postulates, i.e. translation of the newly acquired graphs to Parasite's meaning postulates, in order to improve the knowledge resource of the NLU component for more sophisticated acquisition.

5 Conclusion

The idea to develop the workbench CGExtract appeared in the Larflast project, where Sofia team has to construct a larger KB of CG in the financial domain. Using CGExtract, relatively simple graph can be easily acquired; the semantic checks of KB consistency is the most useful functionality of the workbench. At the same time it is clear that rather complex graphs (like e.g. the birthday party example [19]) can be acquired only if our restricted English is turned to formalised English by further constraints. Despite the limitations, CGExtract is a very useful tool in restricted domains.

At present we plan further developments in the following directions:

(i) Elaborated features for definition of the type hierarchy. The type hierarchy can be updated not only by adding new leaves to the hierarchy tree (which is the present solution), but also by adding new types between existing parent and children nodes. This assumes more sophisticated NLdescription of the new type and its position among the others; then the is_a definition will be rather complex and will most probably be represented by several sentences using extended discourse to describe all parents and children. We plan to develop an interactive augmentation of the hierarchy;

(ii) Dialog in cases of ambiguities, especially in cases when more than one graph can be acquired from the given input. The system will show graphical representations of all possible graph translations of the given text and will discuss them. The user can choose one of them or skip all of them and enter new text. In the first situation after choosing the correct graph representation CGExtract resumes the processing.

(iii) Help with samples of controlled English.

(iv) CGWorld will be supplied with functions for better automatic positioning of the acquired graphs, i.e. will generate a plane representation with minimal crossed incoming and outgoing arcs. We have to note that the present drag-and-drop interface will be preserved for further manipulations of the graph to be used by the user if needed.

6 Acknowledgements

We are grateful to Prof. Allan Ramsay for his kind co-operation within the Larflast project and more especially, for giving up the system Parasite for further extensions.

References

1. Cryan, M and A M Ramsay, A Normal Form for Property Theory, 14th Conference on Automated Deduction, 1997, Springer Lecture Notes in Computer Science 1249, pp. 237-251
2. Cyre, W. Knowledge Extractor: A Tool for Extracting Knowledge from Text. In Lukose, Delugach, Keeler, Searle and Sowa (Eds.), Proc. ICCS-97, Seattle, USA, LNAI 1257, pp. 607-610.
3. Cyre, W. Capture, Integration and Analysis of Digital system Requirements with Conceptual Graphs, IEEE Transactions on Knowledge and Data Engineering, Vol. 9, No 1, February, 1997.
4. Dobrev, P. and K. Toutanova, CGWorld - a web based workbench for conceptual graphs management and applications, Proceedings of ICCS-2000, Darmstadt, Germany, Shaker Verlag, pp. 243-256
5. Dracos, N. Controlled Languages, In Machine Translation; an Introductory Guide, 1995, CBLU, University of Leeds, see http://clwww.essex,ac.uk/ doug/book/node101.html
6. Fuchs, G. and R. Levinson. The CG Mars Lander. In Lukose, Delugach, Keeler, Searle and Sowa (Eds.), Proc. ICCS-97, Seattle, USA, LNAI 1257, pp. 611-614.
7. Levinson, R. Symmetry and the Computation of Conceptual Structures. In Ganter and Mineau (Eds.), Proc. ICCS-2000, Darmstadt, Germany, LNAI 1867, pp. 496-509.
8. Mann, G., Control of a Navigating, Rational Agent by Natural Language, PhD thesis, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, 1996
9. Martin, Ph. Conventions and Notations for Knowledge Representation and Retrieval. In Ganter and Mineau (Eds.), Proc. ICCS-2000, Darmstadt, Germany, LNAI 1867, pp. 41-54.
10. Parasite: http://www.co.umist.ac.uk/staff/ramsay.htm, Parasite is free for academic applications.
11. Ramsay, A. , Theorem Proving for Intensional Logic, Journal of Automated Reasoning 14, 1995, pp. 237-255
12. Ramsay A., Does It Make Any Sense? Update Semantics as Epistemic Reasoning, see http://www.co.umist.ac.uk/staff/ramsay.htm
13. A.-M. Rassinoux, R. H. Baud, J.-R. Scherrer. A Multilingual Analyser of Medical Texts. In: W. Tepfenhart, J. Dick, J. Sowa (eds.), Conceptual Structures: Current Practices. Proc. ICCS94, LNAI 835, pp. 84-96.
14. Schroeder, M. Knowledge Based Analysis of Radiology Reports using Conceptual Graphs. In: H. Pfeiffer, T. Nagle (eds.), Conceptual Structures: Theory and Implementation, Proc. 7th Annual Workshop, July 1992, LNAI 754.
15. Sowa, J. Controlled English, www.bestweb.net/ sowa/misc/ace.html
16. Sowa, J. Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA, 1984.

17. Sowa, J. and E. Way. Implementing a Semantic Interpreter using Conceptual Graphs. IBM Journal R&D, Vol. 30 (1), 1986, pp. 57-69.
18. Sowa, J. Using a Lexicon of Canonical Graphs in a Semantic Interpreter. In: Martha Evens (Ed.), Relational Models of the Lexicon, Cambridge University Press, 1988, pp. 113-137.
19. Sowa, J. Knowledge Representation: logical, philosophical and computational foundations, Brooks/Cole, 2000
20. P. Velardi, M. Pazienza, M. DeGiovanetti. Conceptual Graphs for the Analysis and Generation of Sentences. In: IBM J. Res. and Develop. Vol. 32 (2), March 1988, pp. 251-267.