From: AAAI Technical Report FS-96-04. Compilation copyright © 1996, AAAI (www.aaai.org). All rights reserved.

A dynamic lexical knowledge base for natural language processing

Caroline Barrière and Fred Popowich
School of Computing Science, Simon Fraser University
Burnaby, BC, Canada, V5A 1S6
{barriere,popowich}@cs.sfu.ca

Abstract

In this paper we present the design of a Lexical Knowledge Base (LKB) that uses Conceptual Graphs (CGs) to incorporate the information extracted from a children's dictionary. We show how CGs are used to build the concept hierarchy and to express the definitions of words. As an addition to the standard CG formalism, we introduce a notion of certainty on the facts extracted from the dictionary and present in the LKB. This allows us to assign weight to the different types of information (description, usage, examples) given by the dictionary definitions. A representation formalism like CGs, which are advocated as being closely related to natural language, facilitates the building of a dynamic LKB that can be augmented and updated during the processing of natural language text.

Introduction

The primary goals of this project are to extract information from a machine readable dictionary, to use that information to build a Lexical Knowledge Base (LKB) that will help the processing of natural language input, and to use some of the information from these processed texts to improve the LKB. In this paper, we will focus on the structure of the LKB and on how it can dynamically be modified.

We use Conceptual Graphs (CGs) (Sowa 1984) as the basis of our knowledge representation formalism since: (1) CGs allow us to have a uniform representation throughout the LKB, (2) CGs are in many aspects very close to Natural Language (NL) making the content of the LKB understandable by humans as well as making it easier to use NL for describing new information to be included in the LKB.

Our source of knowledge is the American Heritage First Dictionary[1] which contains 1800 entries and is designed for children of age six to eight learning the structure and the basic vocabulary of their language. Using a children's dictionary allows us to restrict our vocabulary, but still work on general knowledge about day to day concepts and actions.

The dictionary is only the starting point for building the LKB. Humans are constantly adapting and changing their models of the world based on new information, thus our LKB should similarly be dynamic and modifiable through the processing of new information.

We will discuss two main components of the LKB: the concept hierarchy and the word definitions. Then we will look at how each component can be updated by processing more text. But first, let us introduce our knowledge representation language.

Conceptual Graphs

CGs (Sowa 1984) were introduced as an attempt to reconcile the best of the "neat"and the "scruffy" approaches to AI; they present a logic-based formalism with the flexibility to express the background knowledge necessary for understanding natural language.

Here are some characteristics of CGs: (1) predicates and arguments from predicate logic are replaced by concepts and relations; (2) concepts are typed allowing selectional restrictions; (3) concepts allow referents to specify an individual or a set of individuals of a certain type; (4) a concept hierarchy can be defined on concepts, and used in graph operations; (5) coreference links identify all the references to a particular individual in multiple graphs; (6) graph manipulation algorithms are defined: *maximal common subgraph* which finds the largest subgraph that subsumes two graphs, *maximal join* which produces a graph combining the information of two graphs; (7) a treatment of quantification is incorporated; (8) their expressive power is equivalent to first order logic but they are argued to be more intuitive and readable, with a smooth mapping to natural language.

Figure 1 shows two sentences with their corresponding CG representations in linear form.

[1] Copyright ©1994 by Houghton Mifflin Company. Reproduced by permission from THE AMERICAN HERITAGE FIRST DICTIONARY.

```
Concept hierarchy:
    [something]←(is-a)←[animal]←(is-a)←[mouse]
                                ←(is-a)←[cat]
                                ←(is-a)←[dog]
            ←(is-a)←[object]←(is-a)←[furniture]←(is-a)←[sofa]

Sentences:
    The two cats chase the mice while the dog Tod is resting.
    A dog rests on a sofa.

    [chase] → (agent) → [cat:{*}@2]
            → (patient) → [mouse:{*}]
            → (during) → [rest] → (agent) → [dog:Tod]

    [rest]  → (agent) → [dog]
            → (location) → [sofa]

Maximal Common Generalization:
    [rest] → (agent) → [dog]

Maximal Join:
    [chase] → (agent) → [cat:{*}@2]
            → (patient) → [mouse:{*}]
            → (during) → [rest]→ (agent) → [dog:*Tod]
                                → (location) → [sofa]
```

Figure 1: Conceptual graph representations

Concept hierarchy

Our concept hierarchy consists of a noun taxonomy and a verb taxonomy constructed from the definitions in our dictionary. We concentrate on verbs and nouns as they account for three quarters of the definitions. The relation (is-a) is used in a general way to represent the subclass/superclass relation, whether it is between nouns or between verbs. In (Barrière & Popowich 1996a), we present how to automatically build the noun taxonomy from the definitions of nouns in the dictionary.

In addition to the (is-a) relations described in the dictionary definitions, we also extract *covert categories*. Cruse introduced a notion of unlabeled categories that can be found using a sentence frame containing a variable X where we determine a set of items that X could be (Cruse 1986). For example, given the sentence frame, *John looked at the X to see what time it was*, we could generate the set {*clock, watch, alarm clock*} as possible values for X. Cruse calls these categories with no name, but for whose existence there is definite evidence, *covert categories*. Covert categories are often present in our dictionary, either because the label is not part of the vocabulary we want to teach the child, or because the label does not exist in the English language.

Although we have already presented a mechanism for finding a covert category using the CG formalism in (Barrière & Popowich 1996a), we now present those categories as ways of establishing selectional restrictions, which play an important part in the dynamic aspect of the LKB.

Finding a covert category consists of finding a graph that subsumes other graphs. For example, by projecting the graph [eat]->(object)->[something] on all the graphs in the LKB, we will find multiple graphs subsumed by that graph that contain a concept more specific than *something*. It is equivalent to using a sentence frame *John is eating something* and asking what *something* could be replaced by. The set of more specific concepts represents the category *food* as they are all subclasses of the class *food*. In that case, we can put a selectional restriction on a case role for the verb examined. Thus the selectional restriction of the "object" relation for *eat* is of class *food*. Furthermore, the class *food* is associated with the CG meaning *eating object* (the label we would come up with for the subsuming graph if the word *food* did not exist). We have a reciprocal relationship between the eating concept and the food concept, expressed through the interrelation between the selectional restriction and the covert category.Figure 2 shows some covert categories found, and we can see that a covert category can be subsumed by another covert category corresponding to a more general graph.

possible label	superclass	subsuming graph
carrier	something	[carry]->(agent)->[something:X]
vehicle	carrier	[carry]->(object)->[people] ->(agent)->[carrier:X]
farm animal	animal	[live]->(agent)->[animal:X] ->(on)->[farm]
farm animal giving food	farm animal	[raise]->(object)->[farm_animal:X] ->(for)->[food]

Figure 2: Covert categories

Definitions

In this part of the LKB, we store the graphs obtained from the definition of each word. The steps required to automatically transform a dictionary definition into a CG have been described in (Barrière & Popowich 1996b). We can see examples of definitions, in Figure 3. A definition contains up to three general types of information: a description, general knowledge or usage, and a specific example.

Since the LKB should be an ever changing source of knowledge, we decided to assign to the information extracted from our dictionary levels of certainty that could change as we analyze more information coming from texts.

We base our notion of certainty of relations on a notion of certainty introduced by Cruse where a semantic trait can be assigned five statuses of necessity to characterize the meaning of a lexical unit: criterial, expected, possible, unexpected, excluded (Cruse 1986).

A semantic trait in the CG formalism is represented by a relation/concept pair. For example, [A]->(X)-

```
CEREAL
Cereal is a kind of food. [description]
[cereal]->(is-a)->[food]

Many cereals are made from corn, wheat or rice. [usage]
[cereal]->(made-of)->[corn]->(or)->[wheat]
                          ->(or)->[rice]

Most people eat cereal with milk in a bowl. [usage]
[eat]->(agent)->[person:{*}]
    ->(object)->[cereal]->(with)->[milk]
                         ->(in)->[bowl]

COOK
To cook is to heat food to make it ready to eat. [description]
[cook]->(is-a)->[heat]->(object)->[food]
                      ->(goal)->[make]->(object)->[food]..
                               ->(att)->[ready]->(to)->[eat]

We cooked the turkey in the oven for six hours. [example]
[cook]->(agent)->[we]
     ->(in)->[oven]
     ->(object)->[turkey]
     ->(for)->[hour: @6]
```

Figure 3: Some dictionary definitions and their CG representation

>[B] means that B is a semantic trait of type X for A. Therefore [A]->(X:certainty level)->[B] represents the certainty level involved in that particular relation.

We will see how the three types of information (description, usage, specific examples) involve concepts at different levels of certainty.

Description

In a noun description containing genus/differentia information, the genus gives the superclass of the noun and is frequently used for noun taxonomy construction (Byrd *et al.* 1987; Klavans, Chodorow, & Wacholder 1990; Barrière & Popowich 1996a). The "is-a" relation between an object and its genus is at a criterial level. The differentia part of the definition contains different attributes that differentiate the word defined from the other words in the same superclass. The presence of those semantic traits in the definition is usually essential to the meaning of the word defined, which gives them a relation at the criterial level with that word. A verb description is similar to a noun in that it gives a general verb class, and instead of listing particular attributes it gives particular case roles, such as an instrument or manner that are specific to that verb.

For example, both descriptions *A ball is a round object*, and *To bite means to cut with your teeth* have two criterial relations, as shown below:

```
[ball]->(is-a:criterial)->[object]
     ->(att:criterial)->[round]
[bite]->(is-a:criterial)->[cut]
     ->(instrument:criterial)->[teeth].
```

Usage

The usage part of the definition gives information about typical situations, or answers to questions such as how to use an object, what it is made of, or what it looks like. Such information can be assigned different levels of certainty. Some keywords present in the definition are good indicators of the certainty level. They usually express a position in a range of quantity, such as *all, some, most, many, few, no* or a range of frequency such as *always, sometimes, never, often*. The CG representation will include the necessity level corresponding to the keyword, or we will assume the level "expected" if generic information suggesting a typical usage is given without a keyword.

Specific examples

In the example part of the definitions, the concepts presented are only possibly interacting with the noun or verb defined in a particular situation. The situations presented are very precise, and therefore the information given is of limited interest by itself but it might reveal some interesting aspects when we cumulate multiple examples as we dynamically update our LKB.

The examples can be seen as different daily experiences. We keep some individual experiences in memory, but when we find a generalizing pattern, we can discard the individual ones, and keep the general pattern that encompasses all the separated examples. On the other hand, when we find one particular example that expresses a different view then the one we hold, we tend to keep it to show as a contradictory example.

We will do the same here with the examples extracted from the dictionary, by trying to find similarities and possible generalizations. In the LKB, the individual examples are kept separately from the generic information given by the usage part. When a generalization pattern is found, it is placed with the generic information, and the examples are discarded.

For doing so, we use the same process as the one used for finding covert categories. We try to find a graph that subsumes multiple other graphs. We find all the concepts that unify with a more general concept in the subsuming graph, and using the concept hierarchy we find the superclass for that set of possible unifiers. This establishes a selectional restriction on that case role for that particular verb.

For example, in *John baked his pie for an hour*, the object of *bake* is *pie* which is a subclass of *food*. In the dictionary, we find that *cakes, bread, loaves, cookies* and *pizza* are being baked, and they are all subclasses of food. We assign a selectional restriction to the object of [bake] by giving the relation an *ex-*

pected certainty level, here [bake]->(object:expected)->[food]. This more general graph showing the selectional restriction will replace the multiple specific example graphs we have. In fact, the selectional restriction should correspond to the covert category meaning "baked food", but as this unlabeled category is built from examples, we don't have an exhaustive list of the items for which *food* is their superclass and that would have *baked food* as their more immediate superclass.

Updating the LKB from text

As we said earlier, the LKB is built dynamically and should continue to evolve as we process new text. The two parts described, the concept hierarchy and definitions, are subject to modification by new incoming information. To continue updating our LKB, after our definitions have all been analyzed, we will look for simple texts, such as children's stories. We assume we are processing simplified text, like that written for children that only includes concepts from the dictionary, or new concepts that we can link to the existing ones via the concept hierarchy or definitions. The sentences should be short and simple as the examples found in the dictionary. So, the process of updating the LKB from test is essentially the same as the one that updates the LKB as dictionary definitions are processed.

Concept hierarchy When analyzing a text that contains concepts not already in the knowledge base, the necessity levels established on the relations can help us find a temporary place for a new concept in the concept hierarchy. For example, if we have the fact [eat]->(instrument:expected)->[utensil] in the knowledge base, and the graph [John]<-(agent)<-[eat]->(instrument)->[chop-sticks] is extracted from a text, we can infer the following taxonomic link: [chop-stick]->(is-a)->[utensil].

The concept hierarchy is usually helpful when we need to compare concepts, as it tends to organize them into branches of related items. When we want to combine new with existing information, we need to find the similarity between concepts to see if they can be mapped together. If we want to compare *banana* and *orange* and they are both subsumed by *fruit*, we can make an attempt at finding a generalization from their respective information and apply it to *fruit*. But sometimes the concept hierarchy is not helpful in finding the semantic similarity between two words. For example, when using the concept hierarchy to establish the similarity between *pen* and *crayon*, we find that one is a subclass of *tool* and the other of *wax*, both then are subsumed by the general concept *something*. We have reached the root of the noun tree in the concept hierarchy and this would give a similarity of 0 based on the informativeness notion (Resnik 1995).

Graph subsumption, which was used to find covert categories and to find generalizations by comparing multiple examples, can be used as a general method for comparing information within the LKB and for comparing incoming information from text with the existing information in the LKB. Continuing the same example with *pen* and *crayon*, the CGs associated with those two words are subsumed by the same graph [write]->(instrument)->[]. In fact this graph, if it subsumed multiple graphs within the LKB, might already be assigned a label and be considered to be a covert category. It expresses the idea of a "writing instrument". The dictionary used here is small enough to extract many covert categories to start with, but as we process more text, we can use this method dynamically to find similarity between pairs, or sets of words and then update our concept hierarchy by assigning labels to those subsuming graphs and putting them at the appropriate place in the concept hierarchy.

Definitions The certainty levels that we introduced within the CG representation will play an important part in the changes to occur in the LKB. As we said earlier, the examples can be seen more or less as different daily experiences. We keep the general pattern that encompasses all the separated examples, and when we do find one particular example that expresses a different view, it is kept as a contradictory example.

In a previous example, we assigned a selectional restriction on the object of *bake* as:

[bake]->(object:expected)->[food]

If we then encounter an example saying: *Susan baked her clay animals in the oven* which does not follow the norm, we include it as a counter example:

[Susan]<-(agent)<-[bake]->(object)->[animals]->(made-of)->[clay]

We can establish a threshold for the number of examples to be encountered before we actually move the information from the example part to the usage part. This means deleting multiple examples, and adding a generic information with a selectional restriction expressed as a "possible" or "expected" relation. For example, if we see more examples, in which we bake clay pots, then clay figurines, then we can replace the examples we were keeping with:

[bake]->(object:expected)->[food]
->(object:possible)->[object]->(made-of)->[clay]

Discussion

We presented two main parts of a LKB being built from a children's dictionary: the concept hierarchy and the definitions. We showed that: (1) the CG formalism can be used to represent all parts of our LKB, (2) the CG corresponding to a sentence is in itself readable, which is important for making the information contained in the LKB understandable by humans (3) the CG formalism can be augmented to better represent the relative importance of the multiple facts given in the dictionary by assigning certainty levels to the relations between concepts (4) our LKB is dynamic, with graph subsumption forming the basis of the update mechanism; we can modify the LKB using its own information or information coming from simple texts by making comparisons and finding generalizations. This constant evolution, which demands of our LKB that it is able to process text and to incorporate its information to update itself, constitute our knowledge reasoning.

The idea of constructing a LKB using concepts and labeled links can be traced back to (Quillian 1968), where a small number of words and their dictionary definitions were encoded by hand. Here we try to perform all the steps automatically, from the analysis of the dictionary definitions to the construction of their CG representation, to the building of the concept hierarchy including the covert categories, and to the update of the LKB by the analysis of new text. As some problems will be hard to solve (such typical NLP problems as word sense disambiguation, propositional attachment, anaphora resolution, which were dealt with manually in Quillian), the overall system will need some human intervention, and therefore be semi-automatic. Our C++ implementation using a Conceptual Graph platform(Haemmerle 1995) is too preliminary to allow us to present results of our experiments.

Working with a children's dictionary makes us concentrate on a project of a manageable size, while using a non-trivial subset of English which emphasizes non-technical, daily usage words. This is an interesting vocabulary to model common situations occurring in everyone's life.

We presented the concept hierarchy which is automatically built from the analysis of the dictionary definitions. We introduced the covert categories which are created by processing information already in the LKB and which can be used in complement to the concept hierarchy in semantic similarity measures. We saw how the different parts of a definition: description, usage and example are assigned a different weight and play a different role in the LKB. The description gives information usually essential to the definition of the word, and it also contains the information used for building the concept hierarchy. The usage part is the richest of all parts of the definition, giving some selectional restrictions. The specific examples give the best illustration of the dynamic aspect of the LKB; processing them makes us change our expectations about typical situations, take note of exceptions to rules, etc.

Humans are constantly adapting their models of the world, often using NL to achieve this goal. To mimic this process, the LKB should be dynamic and easily modified through the processing of additional NL input. In our project, the dictionary is the starting point for building the LKB. The same method used to process the dictionary definitions can be used to process new text, assuming this text is written for children or it is made of sentences containing a simple vocabulary and written in a simple way. Further research has to be done to establish a reasonable updating mechanism of the information by comparing and modifying the certainty levels assigned in the LKB.

References

Barrière, C., and Popowich, F. 1996a. Building a noun taxonomy from a children's dictionary. In *Euralex'96*.

Barrière, C., and Popowich, F. 1996b. Concept clustering and knowledge integration from a children's dictionary. In *Proc. of the 16 th COLING*.

Byrd, R.; Calzolari, N.; Chodorow, M.; Klavans, J.; Neff, M.; and Rizk, O. 1987. Tools and methods for computational lexicology. *Computational Linguistics* 13(3-4):219–240.

Cruse, D. 1986. *Lexical Semantics*. Cambridge University Press.

Haemmerle, O. 1995. *CoGITo: une plate-forme de developpement de logiciels sur les graphes conceptuels*. Ph.D. Dissertation, Universite Montpellier II, France.

Klavans, J.; Chodorow, M. S.; and Wacholder, N. 1990. From dictionary to knowledge base via taxonomy. In *Proceedings of the 6th Annual Conference of the UW Centre for the New OED: Electronic Text Research*, 110–132.

Quillian, M. 1968. Semantic memory. In Minsky, M., ed., *Semantic Information Processing*. MIT Press, Cambridge, MA.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14 th IJCAI*, volume 1, 448–453.

Sowa, J. 1984. *Conceptual Structures in Mind and Machines*. Addison-Wesley.