Automatic Knowledge Acquisition in Case-Based Text Generation

Virginia Francisco and Raquel Hervás and Pablo Gervás

Departamento de Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
{virginia,raquelhb}@fdi.ucm.es,pgervas@sip.ucm.es

Abstract. This paper discusses the effect that partially automating the construction of a case base has on an existing CBR system for template selection in natural language generation. Details are provided on the actual process of partial automation that has been applied to obtain the case base, together with the various ingredients of the original system that have required modifications to cope with the semi-automatically built case base. This includes resorting to an existing taxonomy instead of recreating a specific one for the particular domain under consideration.

1 Introduction

The task of obtaining reusable cases from plain text faces the challenge of processing the text in an intelligent way to access some of its underlying conceptual content. Several efforts in the past have addressed this problem from the point of view of accessing the semantic information conveyed by text, to be used in CBR systems that deal with information. The present paper addresses a slightly different problem, in the sense that it aims to access the syntactic information hidden in the text. A given sentence is processed in search of its elementary syntactic structure: its main verb, the noun that acts as subject of the verb, and the nouns and prepositions involved in any prepositional complements that modify the action represented by the verb. These data are then represented as a vector of attribute value pairs, and they constitute a structured version of the sentence. This is used as the problem statement of a case. Additional information is extracted from the original sentence in the form of a template for the syntactic structure corresponding to the sentence, with gaps indicated for the places where particular noun phrases can be inserted in different uses of the same syntactic structure. This constitutes the solution part of the case. These cases are then used by a CBR template-based natural language generator for deciding how to convey particular actions received as a conceptual input represented by similar vectors of attribute value pairs. An important disadvantage of this approach is that if the cases and the templates are generated manually, the task becomes onerous in terms of time and effort.

This paper summarises our work in the development of a method for automated construction of textual cases from a corpus within the context of PRINCE,

a template-based solution for natural language generation that relies on reusing fragments of text extracted from typical texts.

2 Problems and Possible Solutions for CBR Template Selection for NLG

This section provides a brief outline of the CBR template selection method for natural language generation that has been described elsewhere [1, 2], together with a description of the main problems found in that approach, and introductions to the two techniques that are applied in this paper to solve the problems.

2.1 The Original System

Hervás and Gervás [1] presented a case-based reasoning solution to perform heuristic lexicalisation in the context of a NLG application. Lexicalisation is the task of selecting adequate text templates for realizing messages describing actions in a given domain. Our method relied on reusing fragments of text extracted from typical texts in a given domain, having applied to them a process which identifies the part of them which is common to all users, and leaving certain gaps to be filled with details corresponding to a new use. For instance, to express the information that *The swindler came to the city*, a template such as _ *came to* _ may be used, filling in the gap with appropriate strings for *the swindler* and *the city*.

Applying a case-based solution presents the advantage that the information needed to solve the problem can be obtained from the original examples of appropriate use that gave rise to the templates. By associating a case with each template, with case attributes consisting of conceptual descriptions of the arguments that were used for the template in the original instance, a case-based reasoning solution can be employed to select the best template for realizing a particular message. For the previous example, a case such as the one given below may be generated and associated to the template _ *came to* _ :

```
LEX:      ACTOR:    OBJECT:
come to   Swindler  city
```

The case base employed is stored in a Case Retrieval Net [3]. This model is appropriate for the problem under consideration, because on one hand our cases consist of attribute-value pairs that are related with one another, and on the other hand the queries posed to the module will not always be complete. To find a lexical tag for a given action, the CRN is queried with the class of elements involved in the action. For each attribute-value pair in the cases an Information Entity (IE) is created. For each case, a node is created which holds references to the information entities that are contained. When introducing an IE, if that entity has already appeared in another case it is not duplicated. Instead, another association is created between the new case and the existing information entity.

The vocabulary of the NLG system is divided in two parts: templates for actions and lexical tags for concepts. The templates for the actions are stored in the cases. The lexical tags for concepts are stored in the general vocabulary of the generation system, that also contains general information about pronouns and other linguistic items.

As nodes are inserted to form the net, a measure of similarity must be established between them. A taxonomy of concepts must be used to deal with this task. Similarity between concepts is computed in terms of the distance traversed over the taxonomical structure to reach one from the other. More details may be found in [1, 2].

In [2], we considered the possibility of extending the case base by using a combination of lexical information extracted from WordNet [4] and a reference corpus of texts from the target domain. The cases were extracted manually from a corpus of 109 classic fairy tales obtained from Internet web sites presenting collections of fairy tales in English, including of Aesop's fables, Afanasiev's collection of Russian fairy tales, and tales by Andersen, the Grim brothers, and Perrault. The total number of sentences in the corpus was 9852 sentences. Sentences from the corpus were taken as examples, from which cases were obtained. For each case, the concepts appearing in the example found in the corpus - which become information entities in the Case Retrieval Net used to store the cases - must also be inserted into the knowledge base of the NLG application. To ensure appropriate performance, they must be inserted at the correct place in the taxonomy that organises the knowledge base, because the system uses the relative positions in this taxonomy to calculate similarity between the query and the cases during retrieval.

The experimental results showed that the use of the case-based reasoning paradigm for the task of lexicalisation is a good approximation whenever enough information is available in the case base to express in an acceptable form any new request. If queries beyond the scope of the input were tried, the system performed poorly. Much of the degradation in performance could be attributed to poor coverage provided by the case base. However, the task of enlarging the case base faced a substantial knowledge acquisition bottleneck. For the process of building the case base to be practically feasible, a degree of automation was required. Two main obstacles were found. One important problem was the fact that the number of sentences was too large for manual processing to be practical. Some means of automating case extraction was needed. Another important problem was the fact that insertion into the taxonomy of the knowledge base was also done manually, and the number of concepts involved had become too large to contemplate this approach.

2.2 Dependency Analysis for Case Extraction

The basic idea of the dependency analysis is that the syntactic structure of a sentence is described in terms of dependency relations between pairs of words (a parent and its child). These relations compose a tree (the dependency tree).

Nowadays there are some free automated dependency-based analysers for several languages: English, Swedish, Turkish ... MINIPAR is one of the most successful analysers. It was developed by Dekang Lin [5]. MINIPAR analyses English texts with high accuracy and efficiency in terms of time. An example of the dependency tree generated by MINIPAR for the sentence *A girl lived in a house* is given in Figures 1 and 2:

```
0(
E0  ()      fin   C     *
1   A       ~     Det   2   det       (gov girl)
2   girl    ~     N     3   s         (gov live)
3   lived   live  V     E0  i         (gov fin)
E2  ()      girl  N     3   subj      (gov live)
4   in      ~     Prep  3   mod       (gov live)
5   a       ~     Det   6   det       (gov house)
6   house   ~     N     4   pcomp-n   (gov in)
)
```

Fig. 1. Example of dependency tree for the sentence *A girl lived in a house*

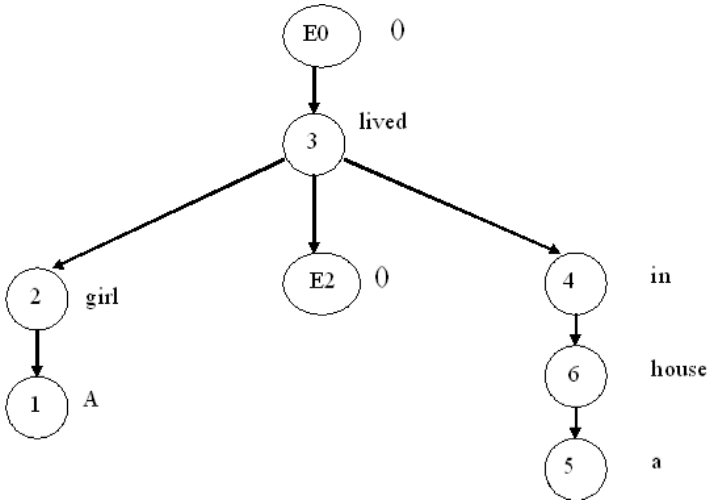

Fig. 2. Example of the graphical representation of the dependency tree for the sentence *A girl lived in a house*

2.3 Using an Existing Taxonomy to Compute Similarity

Currently, a number of efforts in the area of language engineering are aimed to the development of systems of basic semantic categories (often called "upper-level ontologies"), to be used as main organizational *backbones*, suitable to impose a structure on large lexical repositories. Examples of such systems are the PENMAN Upper Model [6], the Mikrokosmos ontology [7], and the WordNet [4] upper structure. Machine learning techniques have been used to build *mapping dictionaries*, lexicons of elementary semantic expressions and corresponding natural language realizations [8].

WordNet is by far the richest and largest database among all resources that are indexed by concepts. For this reason, WordNet has been chosen as initial lexical resource for the development of the module presented in this paper.

WordNet is an on-line lexical reference system organized into synonyms sets - or *synsets* -, each of them representing one underlying lexical concept, linked by semantic relations like synonymy or hyponymy. This organization makes it possible to use WordNet as a knowledge source. The hyponymy/hypernymy relation can be considered equivalent to the "isa" relation, and it induces a taxonomical hierarchy over the set of available concepts.

3 A New Proposal Based on Automated Case Extraction

Initially, both the construction of the cases from the corpus sentences and the insertion of the concepts in the knowledge base were carried out by hand. Instead of this time-consuming process we have used the MINIPAR dependency analyser to automatically obtain the case base from the corpus sentences. Changes in the original implementation of the CBR lexicalisation module have been required to deal with the new information obtained.

3.1 Building the Case Base with MINIPAR

In order to obtain the case base automatically we have developed a method based on MINIPAR. MINIPAR gives a dependency tree for every sentence, and based on this tree we select every verb and the words related to it. This section explains the process followed to obtain the different cases involved in a text and their templates. Firstly MINIPAR processes the texts and generates a dependency tree for each of the sentences. Each tree is analysed in order to obtain the following elements:

- The verbs involved in each sentence. The process looks for every node in the tree marked as a verb during lexical analysis. Each of the verbs found in the sentence will give rise to a new case. The stems of the verbs as identified by MINIPAR are stored in the LEX attribute field of the case.
- The nodes which depend on each of the verbs. Once we have the children of every verb we process them in search for the rest of the elements required to build the cases and templates:

- Subject of the verb. MINIPAR identifies for each verb a special node that is marked as subject of that verb during lexical analysis. The stem of the subject node is stored in the ACTOR attribute field of the case.
- Object of the verb. In a similar way, MINIPAR identifies objects of the verb. The stem of the object node is stored in the OBJECT attribute field of the case.
- Prepositions. MINIPAR identifies with a special label the prepositions that appear in the sentence. Each preposition found in the sentence gives rise to a new field in the case. This new field is labelled with the preposition itself as name of the attribute.
- Words related to prepositions. MINIPAR indicates dependency relations for every word. The nouns that act as head of the nodes related to the prepositions identified in the previous step are used as values for the preposition attributes discovered in the previous step.

An example of dependency tree and the case and template generated for the sentence *The prince killed the dragon in the cave* is given in Figure 3.

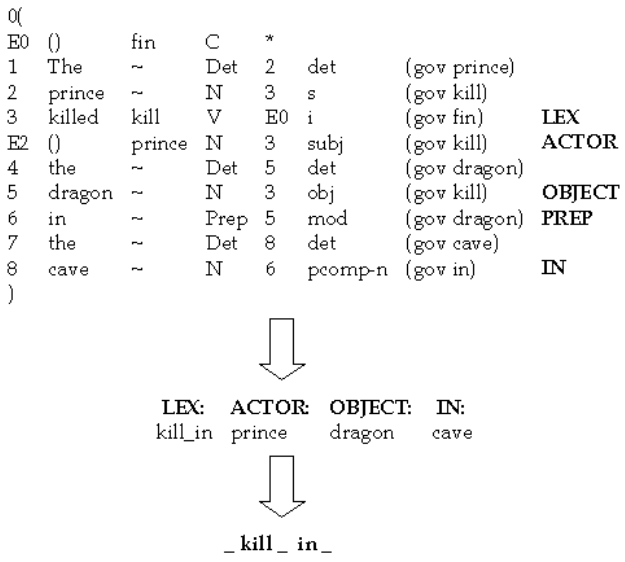

Fig. 3. Dependency tree, template and related case for *The prince killed the dragon in the cave*

In order to evaluate the feasibility of our proposal we carried out some preliminary tests over four tales: “The princess and the pea”, “The fox and the crow”, “The tortoise and the hare” and “The Emperor’s new suit”. These tales

contain 196 sentences in total which generates 323 cases. To evaluate the automated generation of cases we have generated the cases for every tale and then we have checked the correctness of each of the cases. Figure 4 shows the percentage of success - the percentage of correct cases. Looking at the graph we can see that the average percentage of success is over 60%. This is not related to the acurracy of MINIPAR but to the fact that our first approximation to the problem only uses the basic elements of the resulting dependency tree, as described above.

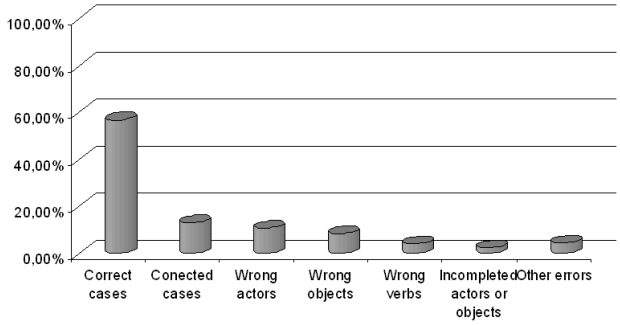

Fig. 4. Percentage of success

Analysing instances where the process produced incorrect cases indicated five main reasons for failure:

- **Nested cases.** There are some cases that have as object or as actor another case. The current representation does not allow nesting of cases, so these subcases are not being recognized. An example is the sentence "It was a princess standing out there in front of the gate". Here, the main verb is "to be", its subject is "it", and the object for the verb "to be" is the whole sentence "a princess standing out in front of the gate", itself another case.
- **Actor mistakenly identified.** In some cases the actor is not identified or the word MINIPAR points as subject is not the correct one. An example is the sentence "I challenge anyone here to race with me", where MINIPAR has decided that the subject for the verb "race" is the word "here", although the correct choice is "anyone".
- **Object mistakenly identified.** In some cases the object is not identified as object in the lexical analysis. An example is the sentence "The emperor's new suit is ready now", where MINIPAR has taken as object of the verb "to be" the word "now". The correct choice would have been "ready".
- **Verb mistakenly identified.** In some sentences the verb is not well identified by MINIPAR. An example is the sentence "The swindlers sat before the empty looms", where "loom" has been identified as a verb.
- **Default parsing of conjunctions.** MINIPAR always parses conjunctions as clause conjunctions, which ledds to incorrect parsing of NP conjunctions

.An example is the sentence "he did not admire the exquisite pattern and the beautiful colours", where "pattern" has been chosen as the unique object of the verb "admire". However, "colour" is also an object of the verb.

Figure 5 shows the relative contribution to the total error of each source of failure in terms of percentages of the total number of processed cases.

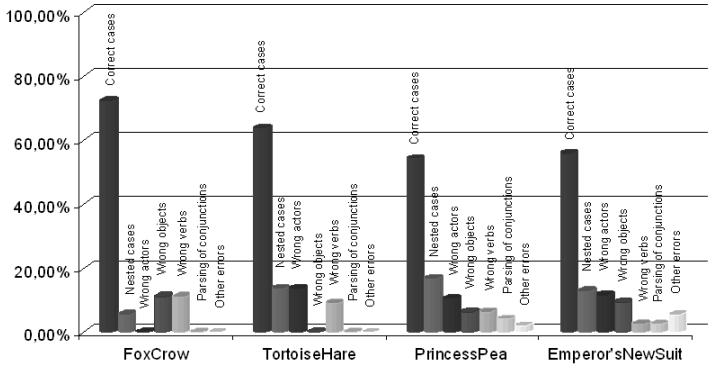

Fig. 5. Percentage of wrong cases group by reasons

3.2 Template Selection Using the Case Base

The hyponymy/hypernymy relation of WordNet can be seen as a "isa" relation. WordNet can therefore be used as a taxonomy over which to automatically calculate the similarity between the concepts appearing in the cases. This requires some additional measures when creating the case base, to ensure that all elements appearing as arguments anywhere in the case base are adequately covered by WordNet. A preliminary filter is applied to the automatically generated cases, so that if one of the elements of a case is not found in WordNet, the case is discarded. From our initial corpus 323 cases were generated, and 172 of them were discarded using WordNet, being our final case base formed by a total of 151 cases.

The similarity between two entities is calculated by taking into account the distance between them and using Formula 1.

$$sim(c1, c2) = 1 - (distance(c1, c2)/20) \tag{1}$$

The distance between two concepts is calculated by finding their first shared ancestor or hypernym, and adding up the distance between this ancestor and each of the concepts. It can be seen as the number of nodes we have to pass when going from one of the concepts to the other. It is also necessary to have

a similarity value for each entity with itself. This value is always the maximum possible, because the distance between the entity and itself is 0. Examples of similarities between concepts can be seen in Table 1.

Table 1. Similarity examples using WordNet

Concept	Hypernyms list	Distance	Similarity
princess	`[ aristocrat leader person organism living thing object entity ]`	1	0.95
knight	`[ male_aristocrat aristocrat leader person organism living thing object entity ]`		
king	`[ sovereign ruler person organism living thing object entity ]`	5	0.75
knight	`[ male_aristocrat aristocrat leader person organism living thing object entity ]`		
princess	`[ aristocrat leader person organism living thing object entity ]`	-	0
Cinderella	`[ character imaginary_being imagination creativity power knowledge psychological_feature`		

The concepts "princess" and "knight" have a high similarity because both of them are aristocrats, and they share most of their ancestors. The concepts "king" and "knight" are similar because they both are persons, but this shared ancestor is very general and their similarity is not as high as in the previous example. The concepts "princess" and "Cinderella" are a special example, because they do not share any of their ancestors. Because of that, similarity 0 is directly assigned to them. This is due to the fact that, when locating Cinderella within the taxonomy, WordNet assigns more weight to the fictional nature of Cinderella than to the physical nature of the character described.

Each of the IEs is related to the cases to which it belongs with a certain value of relevance. In the implemented module we have chosen that all the elements in a case has relevance 0.5.

4 Conclusions and Future Work

Overall, our conclusions are that dependency analysis provides a good first approximation for extracting automatically the information needed for case-based template selection. Full coverage of the initial corpus is not a priority since texts to be generated need not match those in the corpus precisely. Even with the current restrictions imposed by the internal representation, the success rate for that stage of the process is close to 60%. This indicates that a substantial portion of the corpus can be converted into cases. The use of WordNet as a taxonomical knowledge base is a poor solution, for this particular domain, since more

than half of the cases extracted from the tales in the initial sample had to be discarded because the elements appearing in them were not covered by WordNet. However, this poor result may be influenced by the large number of proper nouns that appear in fairy tales. A possible addition to the system would be a knowledge base that includes proper nouns as well as general concepts. Further work will consider alternative language analysis tools and lexical resources.

One of the points to take into account in the future versions is the resolution of pronominal references. In the current version the pronoun are taken as value of the different fields (actor, object, ...). A method for anaphora resolution must be developed in future versions in order to solve this problem.

The use of WordNet has not proved as succesful as it promised to be. The particular organization of its concepts leads to strange results - such as Cinderella and princess not being similar at all. The use of alternative knowledge bases must be considered.

Another issue that requires attention is extending the internal representation format to cope with nested cases. MINIPAR provides sufficient information to identify nested structures but the current version of the CBR process does not support nested cases. This modification is not trivial since it requires careful reconsideration of our associated processes of retrieval and adaptation.

The original similarity measure was normalised over the deepest branch of the taxonomy. The similarity being employed in the current version establishes a normalising upper limit independent of the depth of WordNet as a taxonomy. This should be corrected in subsequent versions.

References

1. Hervás, R., Gervás, P.: Case Retrieval Nets for Heuristic Lexicalization in Natural Language Generation. In Cardoso, A., Bento, C., Dias, G., eds.: Progress in Artificial Intelligence (EPIA 05). Number LNAI 1036, (Springer-Verlag)
2. Hervás, R., Gervás, P.: Case-based reasoning for knowledge-intensive template selection during text generation. In: Proc. of the 8th European Conference on Case-Based Reasoning, Springer-Verlag (2006)
3. Lenz, M., Burkhard, H.D.: Case Retrieval Nets: Basic Ideas and Extensions. In: KI - Kunstliche Intelligenz. (1996) 227–239
4. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38** (1995) 39–41
5. Lin, D.: Dependency-based evaluation of MINIPAR. In: Proc. of Workshop on the Evaluation of Parsing Systems, Granada, Spain (May 1998)
6. Bateman, J.A., Kasper, R.T., Moore, J.D., Whitney, R.A.: A General Organization of Knowledge for Natural Language Processing: the PENMAN upper model (1990)
7. Mahesh, K.: Ontology development for machine translation: Ideology and methodology. Technical Report MCCS-96-292 (1996)
8. Barzilay, R., Lee, L.: Bootstrapping lexical choice via multiple-sequence alignment. In: Proc. of the EMNLP'02. (2002) 164–171