Strategies for Comparison in Encyclopædia Descriptions

Maria Milosavljevic and Robert Dale
Microsoft Research Institute
Macquarie University
North Ryde NSW 2113 Australia
{mariam,rdale}@mpce.mq.edu.au

February 7, 1996

Abstract

In the context of encyclopædia descriptions, comparisons are typically employed to distinguish similar entities, or to illustrate a property of an entity by referring to another commonly known entity which shares that property. Based on a corpus analysis, we define three types of comparisons and outline some strategies for applying these in the generation of encyclopædia descriptions. In particular, we describe how these comparison strategies are used within the PEBA-II hypertext generation system.

1 Introduction and Aims

In this paper, we outline some strategies for comparison which we use in PEBA-II, a hypertext generation system which produces encyclopædia descriptions of entities as World Wide Web (WWW) documents, based on an underlying taxonomic knowledge base. PEBA-II is part of a larger research programme built around the idea of an intelligent on-line encyclopædia, where the descriptions produced by the system vary for different users and at different times. Our work is grounded in the domain of animal descriptions, although similar issues arise in many other domains.

Comparisons are widespread within existing encyclopædia descriptions. In particular, when describing a new concept to a user, a comparison may be made with reference to other known concepts or ideas, enabling the hearer to more easily process and understand the new material (see [Milosavljevic 1996]). So, for example, if the user knows about the porcupine and requests a description of the echidna, then we might describe the echidna by highlighting both its similarities to and differences from the porcupine.

Clearly this requires us to make use of some notion of a user model, and in a way that is distinct from previous work in user-modelling in text generation (see, in particular, [Paris 1987]): our aim is to produce texts which introduce new concepts by reference to existing knowledge the user is assumed to have, thus employing the user model to greater advantage.

Making use of a user model in this way is particularly important in the context of the dynamic construction of hypertext documents from an underlying representation: by employing text generation techniques, we can produce context-dependent descriptions which vary depending on the information which has already been presented to the user, thus overcoming some of the limitations of hypertext documents which have been constructed simply by breaking an existing linear text into pieces. As has been noted by others (see, for example, [Reiter *et al* 1992]), the dynamic generation of hypertext also permits the user to effectively drive the text generation system, alleviating from the system some of the responsibility of reasoning about what to present to the user.

In Section 2, we provide an overview of the PEBA-II system; in Section 3, we provide a definition of comparison and identify three types of comparison on the basis of a corpus analysis; and in Section 4 we describe how the corresponding discourse strategies are implemented in PEBA-II. Section 5 ends the paper by pointing to some future research directions.

2 An Overview of PEBA-II

The architecture of the PEBA-II system is shown in Figure 2; the components are as follows.

The knowledge base that currently underlies the system has been hand-constructed from an analysis of animal encyclopædias and constitutes a taxonomy of the Linnaean animal classes with their associated properties.

The plan library consists of discourse plans which may be used by the text planning component. Currently, the system makes use of two high level discourse plans, which we name identify and compare-and-contrast. The identify discourse plan is used to describe an entity and the compare-and-contrast discoure plan is used to compare two entities. These discourse plans are similar in spirit but rather different in content to the similarly-named schemas used by McKeown [1985], with a number of the differences arising from the fact that we are generating hypertext pages.

A new discourse goal is generated by the user clicking on a hypertext link in the current document being viewed. Given this new goal, the text planning component selects any relevant information from the knowledge base and organises the information according to the current discourse plan. The leaves of the instantiated discourse plan are then realised via a sentence planning component which uses a small unification-based grammar developed for this domain; the realisation engine is Elhadad's [1992] FUF.

The output from the PEBA-II system is a document marked up using a subset of HTML commands. This document may be displayed using any WWW document renderer such as Mosaic or Netscape. The user poses new discourse goals to the system by clicking on any of the hypertext tags, and the cycle continues.[1]

The combination of text generation and hypertext has been explored by others, most notably in Moore's [1995] PEA and in Reiter *et al*'s [1992, 1995] IDAS. PEBA-II is closest in concept to the IDAS system; a more detailed description of PEBA-II can be found in [Milosavljevic, Tulloch and Dale 1996].

3 Defining Comparisons

3.1 Data analysis

A corpus analysis has been conducted to identify how comparisons are used in encyclopædia articles, so that these techniques may be built into the PEBA-II system. In the first instance, we have concentrated on the domain of animal descriptions; we intend to widen the scope of this analysis to other domains in order to provide a more domain-independent theory of comparative forms. The two encyclopædias analysed were Microsoft Encarta [Microsoft 1995] and Groliers Multimedia Encyclopædia [Groliers 1992]; each encyclopædia yielded around 1200 animal entries, and from these we collated a subcorpus of sentences involving comparison. This subcorpus contains 1722 sentences from the Encarta corpus, and 1557 from the Groliers corpus.

The aim of the corpus analysis was to reverse-engineer the comparisons found in animal descriptions by studying the following questions:

- What entities are compared in descriptive texts and how do they relate to each other?

[1] The system is available on the Web at URL: `http://www.mpce.mq.edu.au/msi/peba.html`.

- Why are these particular entities compared? Why are some entities better comparators than others?
- How are entities compared in encyclopædia texts? What techniques do we need to build into a text generation system to be able to produce similar comparisons?

3.2 Some Definitions

3.2.1 Comparison

We will adopt the following definitions:

> A *comparative proposition* is a proposition whose purpose is to draw the hearer's attention to a difference or a similarity that two entities have for a shared attribute.
>
> A *comparison* is the linguistic realisation of a set of one or more comparative propositions, where the purpose of the set of propositions is to draw the hearer's attention to one or more differences or similarities between two entities.

We have identified three different types of comparative forms that appear in descriptive texts, which we refer to here as DIRECT COMPARISONS, CLARIFICATORY COMPARISONS, and ILLUSTRATIVE COMPARISONS. Of these three types, only the first has been explored in the context of natural language processing: McKeown [1985], McCoy [1988] and Maybury [1995] have all looked at various aspects of direct comparisons. Clarificatory and illustrative comparisons, on the other hand, have not been dealt with in previous work on language generation.

3.2.2 Direct Comparisons

A DIRECT COMPARISON is a comparison whose purpose is to compare two entities where neither entity is more important than the other. In the context of a language generation system like PEBA-II, direct comparisons arise when the user enters a request such as: *What is the difference between the Echidna and the African Porpupine?* PEBA-II generates the text shown in Figure 3 in response to such a query.

In this text, neither the echidna nor the porcupine are more important, and the purpose of the text is to determine their similarities and differences based on both their relationship within a taxonomy of animals (based on their lowest common ancestor) and their attributes. This is, of course, the same notion of comparison that is used in McKeown's [1985] TEXT system.

The key point here is that direct comparisons are user-initiated. More interesting from the point of view of user modelling are clarificatory and illustrative comparisons: here, the entity being described by the system is related to some other entity chosen by the system.

3.2.3 Clarificatory Comparisons

A CLARIFICATORY COMPARISON is a comparison whose purpose is to describe an entity by distinguishing it clearly from another entity with which it might be confused. The features distinctive to the comparator entity are not important.

The main difference between a clarificatory comparison and a direct comparison is that a clarificatory comparison is made within a text whose purpose is to describe one entity and not purely to provide a comparison between two entities. A clarificatory comparison serves to explain the features of the main entity being described (henceforth, we will refer to this entity as the FOCUSSED ENTITY); thus, it corresponds to the user entering a request such as *What is the echidna?* In such a case, instead of describing the echidna in isolation, the system may choose to describe it using a clarificatory comparison with the porpupine.

There are two reasons why a clarificatory comparison might be used:

- The focussed entity might be extremely similar to another entity, and therefore often confused with that entity. In this case, it is important that, when describing the focussed entity, it is sufficiently distinguished from the comparator entity.

- Alternatively, the comparator entity might already be known to the user; in such a case, a clarificatory comparison between these entities may aid the user's understanding of the focussed entity.

For example, consider the following text extracted from the animal corpus:

> Goat, common name for any of eight species of cloven-hoofed, horned mammals closely related to the sheep: The two differ in that the goat's tail is shorter and the hollow horns are long and directed upward, backward, and outward, while those of the sheep are spirally twisted. The male goats have beards, unlike sheep, and differ further by the characteristic strong odor they give off in the rutting season.

In this text, the focussed entity (the goat) is very similar and might often be confused with the comparator entity (the sheep). A reader who is familiar with the comparator entity will also more easily form a mental picture of what the focussed entity is like.

There are a number of interesting research issues here:

- How do we make clarificatory comparisons which take into account any incorrect inferences which the user might make? For example, if the user requests a description of the echidna and the system describes the echidna by informing the user of its similarities with the porcupine and not their differences, then the user could be led to believe that the two animals are more similar than they are in reality. The text shown above very carefully describes both similarities and differences.

- How is a comparator entity selected? The most appropriate comparator for the echidna is the porcupine, but the two entities are not closely related within the Linnaean taxonomy of animal classes. The reason for the choice of comparator entity here lies in the fact that both animals possess sharp spines, but this is the only distinguishing property the animals share.

A user model is advantageous here since the salience of different attribute types will vary from person to person. For example, if external appearance is the most salient attribute, then we would want to compare the echidna to the porcupine. If, on the other hand diet is considered a more salient feature, then we might compare the echidna to the anteater. The geographical location of the user can also play an important role: for example, Australians are not necessarily aware of the existence of squirrels, and some North Americans might only know of the existence black squirrels.

3.2.4 Illustrative Comparisons

An ILLUSTRATIVE COMPARISON is a comparative proposition whose purpose is to describe one or more attributes of an entity by refering to the same attribute(s) of another entity with which the user is familiar. In most cases, only one attribute is at issue, and it is this single common property which is important.

The difference between an illustrative comparison and a clarificatory comparison is that the comparator entity, although of a similar type (in this case, an animal), may only share one attribute with the focussed entity, and is not necessarily similar in any other way to the focussed entity.

Here are some illustrative comparisons from our corpus:

- Powerful and aggressive animals *about the size of a large dog*, baboons have strong, elongated jaws, large cheek pouches in which they store food, and eyes close together.

- [Aye-aye] are *about the size of a large cat* and have long, bushy tails, a shaggy brown coat, and large ears.

- Slightly larger than chinchillas, the mountain viscachas have long, *rabbitlike ears* and a long *squirrel-like tail.*

In each of these sentences, an illustrative comparison is made so that the reader can more easily grasp the concept being described. Instead of describing the size and proportion of the viscacha's ears in absolute terms, a reference to the rabit's ears makes it easier for the reader to understand what the ears really look like.

4 Implementing Comparison Strategies

Above, we identified three particular types of comparisons that are present in our corpus. In PEBA-II, each corresponds to a particular discourse strategy for generating a hypertext page. In this section, we describe how these strategies are implemented within PEBA-II.

4.1 Choosing Amongst the Strategies

We are faced with two interdependent questions: when do we decide to describe an entity by comparing it to another entity, and how do we decide which type of comparison to use?

Recall from earlier that PEBA-II can address two different discourse goals: requests to describe some specified entity, and requests to compare two specified entities. The latter discourse goal corresponds, of course, to the category of direct comparisons we identified above. As we noted earlier, direct comparisons are thus user-initiated. We are more interested here, of course, in how PEBA-II decides when it is appropriate to use either a clarificatory comparison or an illustrative comparison. Each becomes an option when PEBA-II has been asked to describe some specified entity. A clarificatory comparison is generated whenever the entity to be described is known to have a POTENTIAL CONFUSOR: our implementation of this strategy is currently very simple, and is described in Section 4.3. Illustrative comparisons are the focus of the current work, and we describe our approach to these in Section 4.4.

4.2 Direct Comparisons

As mentioned earlier, the PEBA-II system allows the user to request one of two actions: to describe a single entity or to compare two entities. A direct comparison is generated by PEBA-II whenever the user requests a comparison between two entities. Using a corpus-derived property classification system, the discourse plan used here pairs up those attributes which are of a similar type and compares their values. An example WWW page generated using this strategy is shown in Figure 3.

4.3 Clarificatory Comparisons

The purpose of a clarificatory comparison is to ensure that the reader does not confuse the entity being described with some other entity. Such confusions are possible when the entity being described is similar in relevant respects to some other entity.

We could try to generate such clarificatory comparisons from first principles: when we have to describe some entity e, we could search the knowledge base for entities which share properties with e, and then use some mechanism to determine whether there is any chance that the two entities might be confused. We could then phrase our description of e to make sure that we distinguish e from such potential confusors. For example, in describing the rabbit, it may be important to distinguish it from the very similar hare in order to avoid confusion. There are clearly ideas we might use here in McCoy's [1988] work on correcting a user's misconceptions; however, the real issue here lies in determining whether such a misconception might arise.

There are problems with such an approach: searching the knowledge base in this way would be a very costly process; it assumes a rather more complete knowledge base than we may be able to rely on; and, most important of all, it assumes that we can determine likelihood of confusability on the basis of some metric—but it is not at all clear what such a metric might be.

Our current solution to these problems is to sidestep them entirely: for each entity that has a potential confusor — for example, sheep and goats — we specify this explicitly in the knowledge base by means of a clause of the following form:

- `(hasprop sheep (potential-confusor goat))`

Then, whenever we have to describe the sheep, we know immediately that it has a potential confusor in the goat, and invoke a discourse strategy that makes an explicit comparison between the two entities. The resulting text includes a comparison with the goat but is aimed at describing the sheep and hence goes further than a direct comparison between the sheep and goat.

Hard-coding potential confusors might be considered an 'easy way out', although it is our view that this is one of many places in NLG where there is benefit in adopting solutions that make use of precomputed information in preference to working things out from first principles. For example, singling out potential comparator entities in this way is no different in principle to explicitly marking in the knowledge base those properties which are distinguishing characteristics, a tactic that both McKeown [1985] and we ourselves use. We have adopted this philosophy for various design decisions made in the development of PEBA-II, so that, for example, we also make use of a phrasal lexicon as a repository of precomputed mappings from semantic units to multi-word lexico-syntactic resources [Becker 1979]. Again, a similar philosophy underpins the use of precomputed lists of preferred attributes in the work on the generation of referring expressions reported in [Reiter and Dale, 1992]. Our position is that such methods are a virtue, not a vice.

4.4 Illustrative Comparisons

Currently, most of our attention is focussed on the third category of comparisons, those we have termed illustrative comparisons. These are cases where one or more attributes of an entity being described are compared to those of a common object with which the reader is assumed to be familiar. For the present discussion, we will concentrate on the attribute of size, and the mechanisms used to produce illustrative comparisons that indicate the size of the entity being described. This is probably one of the easiest properties to deal with; it remains to be seen to what extent the mechanisms we propose will generalise to other attributes.

For illustrative comparisons, there are two questions to be answered:

- How do we decide whether an illustrative comparator should be introduced?

- How do we decide which comparator to choose when there are multiple candidates?

We could perform these comparisons using a similar approach to that which we adopted for clarificatory comparisons: for each entity–attribute pair we could specify some entity that can be used as a comparator. Thus, we might have clauses in the knowledge base that look like the following:

- `(hasprop baboon (illustrative-comparator size dog))`

However, this would be unwieldy: part of the justification for taking this approach in the case of clarificatory comparisons is that we would expect a relatively small subset of the entities in the knowledge base to have potential confusors, and so the cost of explicitly encoding a representation of these potential confusors is not too great. However, virtually any entity–attribute pair might be described using an illustrative comparison, and so we need some way of generalising the processing here.

We do this by making use of the notion of a COMMON COMPARATOR SET. This is a set of entity types that can be compared against for illustrative purposes. For the moment, a common comparator set is defined for each attribute we might wish to describe; there may be some scope for interesting generalisations later. We focus here on the size attribute: for this, our common comparator set is the set

- ⟨human, dog, cat⟩

Note that the common comparator set for any given entity and attribute is

- domain specific: different comparator sets for size will be appropriate in different domains;
- user specific: it is likely that different comparator sets will be appropriate for different users; in the current implementation, we only used illustrative comparisons for users who have declared themselves naive; and
- in principle extensible, both directly and indirectly: we can imagine the user explicitly being allowed to specify a set of comparator objects, or we could dynamically extend the set used on the basis of the ongoing discourse history.

There may be ways of building or precompiling a common comparator set automatically using the knowledge base and information from a user model, but for the moment we assume that it has been preconstructed. Given an entity e we want to describe and some attribute a of the entity we want to communicate, we use the algorithm in Figure 1.

To describe attribute a of entity e (the focussed entity)

- Identify comparator set S_a for attribute a
- Val = median value of a for e
- For each $e_i \in S_a$, identify location of Val on the range of values that e has for a
- Choose best match:
    - choose e whose median value for a is closest to Val
    - if this doesn't select uniquely from amongst the comparator set then choose e whose range for a is closest to Val

Figure 1: Choosing a comparator object

The procedure used here for finding the best match is one that in our current experiments looks acceptable, although it is likely to be applicable only for a relatively narrow range of attributes. There are a number of obvious deficiencies, all of which we are currently exploring:

- Properties are not independent: for example, when dealing with size, we also need to take account of similarity of body-form in determining which entity makes the best comparator.
- The user's degree of familiarity with the potential comparators can help in making a choice.
- The degree of relatedness between the two entities can also play a role in choosing the best comparator.

So far, however, the results of this method seem promising. Note that the use of a common comparator set in conjunction with the algorithm specified here means that we can separate the domain-specific aspects of the computation from the domain-independent aspects; in principle, the aim is that the comparator set specifies domain-specific information, but the algorithm itself is domain independent.

As always, our methodology is to pursue solutions that first assume a considerable amount of precompiled knowledge and then introduce generalisability and flexibility through subsequent parameterisation, rather than beginning with a very limited coverage solution that works from first principles. It is our view that this methodology is the only one that is likely to be successful for broad coverage, practical NLG systems.

5 Conclusions and Future Work

In this paper, we have:

- described the PEBA-II system, as an example of a system which integrates natural language generation and hypertext in the provision of user-tailored information;
- defined some notions relevant to the study of comparison; and
- looked at the concept of illustrative comparison in detail, with the aim of defining a mechanism for generating such comparisons that embodies a clear distinction between domain-dependent and domain-independent information.

For future work, we intend to elaborate upon and extend further the techniques described here. In particular, we intend to make use of these notions in the generation of comparisons which take account of the discourse history; some examples of this phenomenon are discussed in [Dale and Milosavljevic 1996].

References

Robert Dale and Maria Milosavljevic [1996] *Authoring on Demand: Natural Language Generation in Hypertext Documents.* To appear in *Proceedings of the First Australian Document Computing Conference.*

Microsoft [1995] *Microsoft (R) Encarta'95 Encyclopædia.* Copyright (c) 1994 Microsoft Corporation. Copyright (c) 1994 Funk and Wagnall's Corporation.

Groliers [1992] *The New Grolier Multimedia Encyclopædia.* Copyright Grolier Incorporated, (c) 1987-1992 Online Computer Systems, Inc.

Kathleen McCoy [1988] Reasoning on a Highlighted User Model to Respond to Misconceptions. *Computational Linguistics* 14(3): 52-63.

Kathleen McKeown [1985] *Text Generation - Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press.

Mark T Maybury [1995] Using Similarity Metrics to Determine Content for Explanation Generation. *Expert Systems with Applications* 8(4): 513-525.

Maria Milosavljevic [1996] Introducing New Concepts Via Comparison: A New Look at User Modeling in Text Generation. In *Proceedings of the Fifth International Conference on User Modelling, Doctoral Consortium*, pages 228-230.

Maria Milosavljevic, Adrian Tulloch and Robert Dale [1996] Text Generation in a Dynamic Hypertext Environment. In *Proceedings of the Nineteenth Australasian Computer Science Conference.*

Cecile Paris [1993] *User Modelling in Text Generation.* Pinter Publishers.

Ehud Reiter and Robert Dale [1992] A Fast Algorithm for the Generation of Referring Expressions. In *Proceedings of Coling–92*, Nantes, France, August 1992.

Ehud Reiter, Chris Mellish and John Levine [1992]

Automatic Generation of On-line Documentation in the IDAS Project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, March–April 1992.

Ehud Reiter, Chris Mellish and John Levine [1995] Automatic Generation of Technical Documentation. *Applied Artificial Intelligence*, Vol 9(2), pp 259–287.

Figure 2: The architecture of the Peba-II system

Figure 3: A direct comparison as generated by PEBA-II