AUTOMATIC ANALYSIS OF DESCRIPTIVE TEXTS

James R. Cowie
Computer Centre
University of Strathclyde,
Royal College, George Street,
Glasgow, G1 1XW. SCOTLAND

ABSTRACT

This paper describes a system that attempts to interpret descriptive texts without the use of complex grammars. The purpose of the system is to transform the descriptions to a standard form which may be used as the basis of a database system knowledgeable in the subject matter of the text.

The texts currently used are wild plant descriptions taken directly from a popular book on the subject. Properties such as size, shape and colour are abstracted from the descriptions and related to parts of the plant in which we are interested. The resulting output is a standardised hierarchical structure holding only significant features of the description.

The system, implemented in the PROLOG programming language, uses keywords to identify the way segments of the text relate to the object described. Information on words is held in a keyword list of nouns relating to parts of the object described. A dictionary contains the attributes of ordinary words used by the system to analyse the text. The text is divided into segments using information provided by conjunctions and punctuation.

About half the texts processed are correctly analysed at present. Proposals are made for future work to improve this figure. There seems to be no inherent reason why the technique cannot be generalised so that any text of semi-standard descriptions can be automatically converted to a canonical form.

I INTRODUCTION

A lot of useful information, covering many subject areas, is presently available in printed form in catalogues, directories and guides. Good examples are plants in "Collins Pocket Guide to Wild Flowers", aeroplanes in "Jane's All the World's Aircraft" and people in "Who's Who". Because this information is represented in a stylised form, it is amenable to machine processing to abstract salient details concerning the entity being described. The research described here is part of a long term project to develop a system which can 'read' descriptive text and so become an expert on the material which has been read.

The first stage of this research is to establish that it is indeed possible to abstract useful information from descriptive text and we have chosen as a typical example a text consisting of descriptions of wild plants. Our system reads this text and generates a formal canonical plant description. Ultimately this will be input to a knowledge-based system which will then be able to answer questions on wild plants.

The paper gives a limited overview of the recent work in text analysis in order to establish a context for the approach we adopt. An outline of the operation of the system is then made.

The analysis of our text proceeds in four separate stages and these are considered in conjunction with a sample text. The first stage attaches to each word in the text attributes which are held in either a keyword list or the system dictionary. This expanded text is then split up using conjunctions, punctuation marks and the keywords in the text to assign each segment of the text to a particular part of the plant. The third stage gathers up the descriptions for a particular part and abstracts properties from them. The final operation formats the output as required.

We then look at the more detailed operation of the system in terms of specific parts of interest. This covers the dictionary, skeleton structures, text splitting, text analysis and the limited word guessing attempted by the system.

Future developments are then considered. In particular the possibility of generalising the system to handle other topics. The actual implementation of the system and the use of PROLOG are examined and we conclude with some notes on the current utility of our system.

II BACKGROUND

Many research workers are interested in different aspects of text analysis. Much of the emphasis of this work depends on the use of sophisticated grammars to map to the internal representation. The work done by Schank (1973) and that of Sager (1981) are two contrasting examples of this interest. In addition to the research oriented work, some commercial groups are interested in the practicability of generating database input from text.

Although the internal details of the various systems are totally different the final result is some form of layout, script or structure which has been filled out with details from the text. The approach of the various groups can be contrasted according to how much of the text is preserved at this point and how much additional detail has been added by the system. DeJong (1979) processes newswire stories and once the key elements have been found the rest of the text is abandoned. Sager makes the whole text fit into the layout as here small details may be of vital importance to the end user of the processed text. Schank in his story understanding programs may actually end up with more information than the original text, supplied from the system's own world knowledge.

The other contrasting factor is the degree of limitation of the domain of interest of the text processors. The more a system has been designed with a practical end in view, the more limited the domain. Schank is operating at the level of general language understanding. DeJong is limiting this to the task of news recognition and abstraction, but only certain stories are handled by the system. Sager has reduced the range still further to a particular type of medical diagnoses.

Very recent work appears to be approaching text understanding from a word oriented viewpoint. Each word has associated with it processes which drive the analysis of the text (Small, 1981). We have also been encouraged in our own approach by Kelly and Stone's (1979) work on word disambiguation. The implication of which seems to be that word driven rules can resolve ambiguities of meaning in a local context.

Our own case is a purely practical attempt to generate large amounts of database building information from single topic texts. It should not be assumed however that a truly comprehensive syntax for a descriptive text would be simpler than for other types. The reverse may be true and the author of the descriptions may attempt to liven up his work with asides, unusual word-orders and additional atmospheric details.

Our system does not use sophisticated grammatical techniques. It is our contention that in the domain of descriptive texts we can make certain assumptions about the way the descriptive data is handled. These allow very crude parsing to be sufficient in most cases.

Similarly the semantic structures involved are simple. A description of an object consisting of several parts usually mentions the part and its properties in a single piece of text. The basic properties we are looking for - shape, colour, size - are all described by words with a direct physical relation or with a simple mental association. What we are really trying to do is tidy the description into a set of suitable noun phrases.

III OUTLINE OF THE SYSTEM

The text analysis system has been constructed on the assumption that much of the information held in descriptive texts can be extracted using very simple rules. These rules are analogous to the 'sketchy syntax' suggested by Kelly and Stone and operate on the text on a local rather than a global basis.

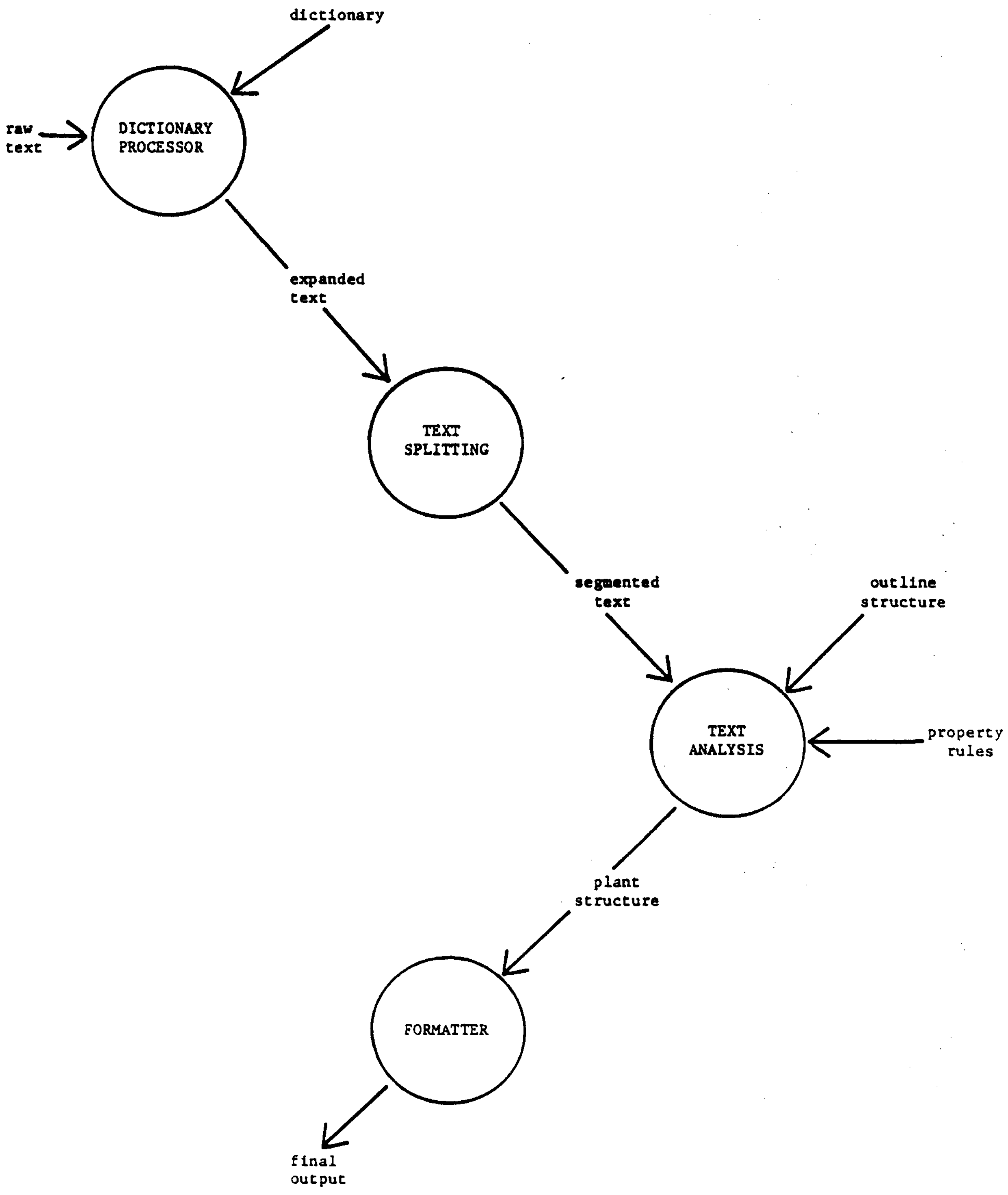
At the time of writing our system processes plant descriptions, in search of ten properties which we consider distinctive. Examples of these properties are the size of the plant, the colour of its flowers and the shape of its flowers. New properties can be added simply by extending the skeleton plant description.

Example 1. A Sample Analysis

SMALL BUGLOSS.
An erect bristly annual, up to a foot high, with wavy lanceolate leaves and small blue flowers which are the only ones of their family to have their corolla-tube kinked at the base; calyx with lanceolate teeth, hardly enlarging but much exceeding the fruit. Habitat: Widespread and locally frequent in open spaces on light soils. April onwards.

TOPIC	COMPONENT PARTS	PROPERTY NAMES	PROPERTY VALUES
plant			
	general		
		name	small bugloss
		size	a foot high
	flower		
		colour	blue
		shape	noinfo
		size	small
	leaf		
		shape	wavy lanceolate
		size	noinfo
		colour	noinfo
	habitat		
		geog-location	widespread
		season	april onwards

Figure 1. System Outline



The texts being processed are plant descriptions as found in McClintock and Fitter (1974). The system has been built to handle this topic and it attempts to fill out various properties for selected parts of a plant. A skeleton description is used to drive the processing of the text. This indicates the parts of the plant of interest and the properties required for each part.

The structure which we presently use is shown in Example 1 after it has been filled out by processing the accompanying description. It should be noted that if the system cannot find a property then the null property 'noinfo' is returned.

An outline of how a description is processed by the system and converted to canonical form is given in Figure 1. There are four distinct stages in the transformation of the text.

A. Dictionary processor.

The raw text is read in and each word in the text is checked in a dictionary/keyword list. Each dictionary entry has an associated list of attributes describing both syntactic and semantic attributes of that word. These attributes are looked at in more detail in section IV. If a word in the text appears in the dictionary it is supplemented with an attribute list abstracted from the dictionary.

The keywords for a text depend on which parts of the object we are interested. Thus for a plant we need to include all possible variants of flower (floret, bud) and of leaf (leaflet) and so on. Fortunately this is not a large number of words and they can be easily acquired from a thesaurus.

The output from this stage is a list of words and attached to each word is a list of the attributes of this word.

B. Text splitting.

The expanded text is then burst into segments associated with each keyword. We identify segments by using 'pivotal points' in the text. Pivotal points are pronouns, conjuntions, prepositions and punctuation marks. This is the simplifying assumption which we make which allows us to avoid detailed grammars. The actual words and punctuation marks chosen to split the text are critical to the success of this method. It may be necessary to change these for texts by a different author as each author's usage of punctuation is fairly idiosynchratic. Within a given work however fairly consistent results are obtained. The actual splitting of the text is covered more fully in section IV C.

C. Text analysis.

We now have many small segments of text each with an attached keyword. This keyword indentifies the text as describing a particular part of the plant. Text segments are gathered together for a particular keyword. This may pull together text from separate parts of the original description

This new unit of text is then examined to see if any of the words or phrases in it satisfy the specific property rules required for this part of the plant. If found the phrases are inserted into appropriate parts of the structure.

D. Formatter.

The ultimate output of the system is intended as input to a relational database system developed at the University of Strathclyde. At the moment the structure is displayed in a form that allows checking of the system performance.

IV SYSTEM DETAILS

A. The Dictionary

The dictionary is the source of the meanings of words used during the search for properties. Two other word sources are incorporated in the system, a list of keywords which is specific to the subject being described and a list of words which may be used to split the text. This second list could probably be incorporated in the dictionary, but we have avoided this until the system has been generalised to handle other types of text.

The dictionary entry for each word consists of three lists of attributes. The first contains it's part of speech, a flag indicating the word carries no semantic information and some additional attributes to control processing. For example the attribute "take-next" indicates that if a property rule is already satisfied when this word is reached in the text then the next word should be attached to the property phrase already found. Thus the word "-" carries this property and pulls in a successive word.

The second list contains attributes whose meaning would appear to be expressible as a physical measure of some kind:- "touch-roughness", "vision-intensity". Many of the words used in descriptions can be adequately categorised by a single attribute of this type. Thus the word red is an "adjective" with a physical property "vision-colour".

The third contains those which require physical measures to be mapped and compared to internal representations or which deal with the manipulation of internal representations alone:- "form-shape", "context-location". Words using these attributes generally tend to be more complex and may have multiple attributes. Thus the word field has as attributes "context-location" and "relationship-multiple-example" whereas the word Scotland also carries "context-location" but is qualified by "relationship-single-example".

We realize this division is delimited by an extremely fuzzy border, but when the search for a basis for word definition was made this helped the intuitive allocation of attributes. Sixty five different attributes have been allocated. Only sixteen of these are used in the rules for our current list of properties.

The size of the dictionary has been considerably reduced by including the algorithm, given by Kelly and Stone (1979), for suffix removal in the lookup process.

B. Skeleton Structure

The structure we wish to fill out is mapped directly to a hierarchical PROLOG structure with the uninstantiated variables, shown in the structure in capital letters, indicating where pieces of text are required. The PROLOG system fills in these variables at run time with the appropriate words from the text. Each variable in a completed structure should hold a list of words which describe that particular property. Thus a partial plant structure is defined as:-

```
plant(
        general(
                size(G1),
                name(G2),
                ),
        flower(
                colour(F1),
                shape(F2),
                ),
        ).
```

This skeleton is accompanied by a set of keyword lists. Each list being associated with one of the first levels of the structure. Thus a partial list for 'flower' might be:-

```
keyword(flower,1).
keyword(bud,1).
keyword(petal,1).
keyword(floret,1).
```

The number indicates which item on the first level of the structure is associated with these keywords.

C. Text Splitting

The fundamental assumption we make for descriptions of objects is that the part described will be mentioned within the piece of text referring to it. Thus conjunctions and punctuation marks are taken to flag pivotal points in the text where attention shifts from one part to another.

We assume initially that we are describing the general details of the plant, so the text read up to the first pivotal point belongs to that part of our structure, keyword level 0. Each subsequent piece of text found assigns to the same keyword until a piece of text is found containing a new keyword. This becomes the current keyword and following pieces of text belong to this keyword until yet another keyword is found.

D. Property Rules

We now gather together the pieces of text for a part of the structure and look for properties as defined in the skeleton structure. A property search is carried out for each of the property names found at level two of the structure. The property rules have the general form:-

```
Set 'property' to NO
repeat{
 examine attributes of next word
  if(suitable modifier attributes)
    then keep word
  if(suitable property attributes)
    then  keep word and set 'property' to YES
  if(no suitable attributes and 'property'is NO)
    then throw away any words kept so far
  if(no suitable attributes and 'property' is YES)
    then exit repeat
  if(no more words)
    then exit repeat
  }
if('property' is YES) then return words kept
if('property' is NO) then return 'noinfo'.
```

E. Special Purpose Rules

We are trying to avoid rules specifically associated with layout which would need redefinition for different texts. However the system does assume a certain ordering in the initial title of the descriptions. Thus the name of the plant is any adjectives followed by a word or words not in the dictionary. It is intended to add rules to detect the Latin specific name of the plant. We have excluded these from our current texts. These will in all probability be based on a similar rule of ignorance, reinforced by some knowledge of permissible suffices.

F. Specially Recognised Words

Certain words are identified in the dictionary by the attributes "take-next" and "take-previous". They imply that if a property rule is satisfied at the time that word is processed then the successor or predecessor of that word and the word itself should be included in the property. The principal use of this occurs in hyphenated words. These are treated as three words; word1, hyphen, word2. The hyphen carries both "take-next" and "take-previous" attributes. This often allows attachment of unknown words in a property phrase. Thus "chocolate-brown" would be recognised as a colour phrase despite the fact that the word chocolate is not included in the dictionary.

Words which actually name the property being sought after carry a "take-previous" attribute. Thus "coloured" when found will pull in the previous word e.g. "butter colour" although the word butter may be unknown or have no specific dictionary attribute recognised by the rule.

V FURTHER DEVELOPMENTS

In the short term, the size of the dictionary and the rules built into the system must be increased so that a higher proportion of descriptions are correctly processed. Another problem which we must handle is the use of qualifiers referring to previous descriptions e.g. 'darker green' or 'much less hairy than the last species'. We intend to tackle this problem by merging the current canonical description with that of plants referred to previously

It would appear from work that has been carried out on dictionary analysis (Amsler, 1981) that a less intuitive method of word meaning categorization may be available. If it proves possible to map from a standard dictionary to our set of attributes or some related set then the rigour of out internal dictionary would be significantly improved and a major area of repetitive work might be removed from the system.

It is also intended to extend the suffix algorithm to handle prefixes and to convert the part of speech attribute according to the transformations carried out on the word. This has not proved important to us up to the present but future uses of the dictionary may depend on its being handled correctly.

In the longer term we intend to generalise the system to cope with other topic areas. In particular, we intend to provide a user interface to allow the system to be modified for a specific topic by user definitions and examples.

The potential also exists for mapping from our word based internal representation to a more abstract machine manipulable form. This may be the most interesting direction in which the work will lead.

VI IMPLEMENTATION

The code for the system is written in PROLOG (Clocksin and Mellish, 1981) as implemented on the Edinburgh Multi Access System (Byrd,1981). This is a standard implementation of the language, with the single enhancement of a second internal database which is accessed using a hashing algorithm rather than a linear search. This has been used to improve the efficiency of the dictionary search procedures.

PROLOG was chosen as an implementation language mainly because of the ease of manipulation of structures, lists and rules. The skeleton plant and keyword lists are held as facts in the PROLOG database. The implementation of the suffix stripping algorithm is a good example of the ease of expressing algorithms in PROLOG. The mapping from the original to our code being almost one to one.

In addition the implementation on EMAS allows large PROLOG programs to be run. The interpretive nature of the language also means that trace debugging facilities are available and new pieces of code can be easily incorporated into the system.

VII CONCLUSIONS

Initial indications suggest that for about 50% of descriptions, all ten properties are correctly evaluated and for about 30%, 8 or 9 properties are correct. The remaining 20% are unacceptable as less than 8 properties are correctly determined by the system.

We anticipate that increasing the knowledge base of the system will significantly increase its accuracy.

The very primitive 'sketchy syntax' approach appears to offer practical solutions in analysing descriptive texts. Furthermore, there seems to be no intrinsic reason why a similar method could not be used to analyse temporal or causal structures. There will always be segments of text that the system cannot cope with and to achieve a greater degree of accuracy we will need to allow the system to consult with the user in resolving difficult pieces of text.

The structured nature of the system output allows the possibility of building a complex database system. A data base system based on the raw text alone has no ability to distinguish to which part of an object any property belongs as its searches are made on the basis of keywords alone without taking contextual information into account.

VIII ACKNOWLEDGEMENTS

I would like to thank the director of the Computer Centre Mr. Grant Fraser for making available time to carry out this work and my supervisor Dr. Ian Sommerville for his help in the development of the system and in the writing of this paper.

IX REFERENCES

Amsler, Robert A. A Taxonomy for English Nouns and Verbs. Proc. 19th Annual ACL, 1981, 133-138.

Byrd, Lawrence, ed. A User's Guide to EMAS PROLOG. D.A.I. Occasional Paper No. 26. Department of A.I. Edinburgh University, 1981.

Clocksin, William F. and Christopher S. Mellish. Programming in PROLOG. Heidelberg: Springer-Verlag, 1981.

DeJong, Gerald F. Skimming Stories in Real Time. Research Report 158. Department of Computer Science, Yale University, 1979.

Kelly E. and P. Stone. Computer Recognition of English Word Senses. Amsterdam: North-Holland, 1979.

McClintock, David and R.S.R. Fitter. The Collins Pocket Guide to Wild Flowers. London: Collins, 1975.

Sager, Naomi. Natural Language Information Processing. Reading, Mass.: Addison-Wesley, 1981.

Schank, Roger C. and Kenneth M. Colby, eds. Computer Models of Thought and Language. San Francisco: Freeman, 1973.