Technical Report

UCAM-CL-TR-23
ISSN 1476-2986

Number 23

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Two papers about the scrabble summarising system

J.I. Tait

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Two Papers about the Scrabble Summarising System

by J.I. Tait

This report contains two papers which describe parts of the Scrabble English summarising system. The first, "Topic Identification Techniques for Predictive Language Analysers" has been accepted as a short communication for the 9th International Conference on Computational Linguistics, in Prague. The second, "Generating Summaries using a Predictive Language Analyser" is an extended version of a discussion paper which will be presented at the European Conference on Artificial Intelligence in Paris. Both conferences will take place during July 1982.

University of Cambridge Computer Laboratory
Corn Exchange St.,
Cambridge CB2 3QG,
England.

Topic Identification Techniques for Predictive Language Analysers

1. Introduction

The use of prediction as the basis for inferential analysis mechanisms for natural language has become increasingly popular in recent years. Examples of systems which use prediction are FRUMP [DeJong79] and [Schank75a]. The property of interest here is that their basic mode of working is to determine whether an input text follows one of the systems pre-specified patterns; in other words they predict, to some extent, the form their input texts will take. A crucial problem for such systems is the selection of suitable sets of predictions, or patterns, to be applied to any particular text, and it is this problem I want to address in the paper.

I will assume that the predictions are organised into bundles according to the topic of the texts to which they apply. This is a generalisation of the script idea employed by [DeJong79] and [Schank75a]. I will call such bundles stereotypes.

The basis of the technique described here is a distinction between the process of suggesting possible topics of a section of text and the process of eliminating candidate topics (and associated predictions) which are not, in fact, appropriate for the text section. Those candidates which are not eliminated are then identified as the topics of the text section. (There may only be one such candidate.) This approach allows the use of algorithms for suggesting possible topics which try to ensure that if the system possesses a suitable stereotype for a text section it is activated, even at the expense of activating large numbers of irrelevant stereotypes.

This technique has been tested in a computer system called Scrabble.

2. Suggesting Candidate Topics

The discovery of candidate topics for a text segment is driven by the association of a set of patterns of semantic primitives with each stereotype. (For the purposes of this paper it is assumed that the system has access to a lexicon containing entries whose semantic component is something like that used by [Wilks77].) As a word is input to the system the senses of the word are examined to determine if any of them have a semantic description which contains a pattern associated with any of the system's stereotypes. If any do contain such a pattern the corresponding stereotypes are loaded into the active workspace of the system, unless they are already active.

3. Eliminating Irrelevant Candidates

In parallel with the suggestion process, the predictions of each stereotype in the active workspace are compared with the text. In Scrabble, the sentences of the text are first parsed into a variant of Conceptual Dependency (CD) representation ([Schank75b]) by a program described in [Cater80]. The semantic representation scheme has been extended to include nominal descriptions similar in power to those used by [Wilks77]. The predictions are compared with the CD representation structures at the end of each sentence; but nothing in the scheme described in this paper could not be applied to a system which integrated the process of parsing with that of determining whether or not a fragment of the text satisfies some prediction, as is done in [DeJong79].

It is likely that stereotypes which are not relevant to the topic of the current text segment will have been loaded as a result of the suggestion process. Since the cost of the comparison of a prediction with the CD-representation of a sentence of the text is not trivial it is important that irrelevant stereotypes are removed from the active workspace as rapidly as possible. The primary algorithm used by Scrabble removes any stereotype which has failed to predict more of the propositions in incoming the text than it has successfully predicted. This simple algorithm has proved adequate in tests and its simplicity also ensures that the cost of removing irrelevant stereotypes is minimised.

Further processing is subsequently done to separate stereotypes which were never appropriate for the text from stereotypes which were useful for the analysis of some part of the text, but are no longer useful.

4. An Example

Consider the following short text, adapted from [Charniak78]:

> Jack picked a can of tuna off the shelf. He put it in his basket. He paid for it and went home.

Assume that associated with the primitive pattern for food the system has stereotypes for eating in a restaurant, shopping at a supermarket, and preparing a meal in the kitchen. The lexicon entry for tuna (a large sea fish which is caught for food) will contain this pattern, and this will cause the loading of the above three stereotypes into the active workspace. The restaurant stereotype will not predict the first sentence, and so will immediately be unloaded. Both the supermarket and kitchen stereotypes expect sentences like the first in the text. When the second sentence is read, the supermarket stereotype will be expecting it (since it expects purchases to be put into baskets), but the kitchen stereotype will not. However the kitchen stereotype will not be unloaded since, so far, it has predicted as many propositions as it has failed to predict. When the third sentence is read, again the supermarket stereotype has predicted propositions of this form, but the kitchen stereotype has not. Therefore the kitchen stereotype is removed from the active workspace, and the topic of text is firmly identified as a visit to the supermarket.

It should be noted that a completely realistic system would have to perform much more complex processing to analyse the above example. In such a system additional stereotypes would probably be activated by the occurrence of the primitive pattern for food, and it is likely that yet more stereotypes would be activated by different primitive patterns in the lexicon entries for the words in the input text.

5. Conclusions

The technique described in this paper for the identification of the topic of a text section has a number of advantages over previous schemes. First, its use of information which will probably already be stored in the natural language processing system's lexicon has obvious advantages over schemes which require large, separate data-structures purely for topic identification, as well as for making the predictions associated with a topic. In practice, Scrabble uses a slightly doctored lexicon to improve efficiency, but the necessary work could be done by an automatic pre-processing of the lexicon.

Second, the scheme described here can make use of nominals which suggest a candidate topic, and associated stereotypes, without complex manipulation of semantic information which is not useful for this purpose. The scheme of [DeJong79], for example, would perform complex operations on semantic representations associated with "pick" before it processed the more useful word "tuna" if it processed the above example text.

Third the use of semantic primitive patterns has greater generality than techniques which set up direct links between words and bundles of predictions, as appeared to be done in early versions of the SAM program [Schank75a].

One final point. The technique for topic identification in this paper would not be practical either if it was very expensive to load stereotypes which turn out to be irrelevant, or if the cost of comparing the predictions of such stereotypes with the text representation was high. The Scrabble system, running under Cambridge LISP on an IBM 370/165 took 8770 milliseconds to analyse the example text above of which 756 milliseconds was used by loading and activating the two irrelevant stereotypes and 103 milliseconds was spent comparing their predictions with the CD-representation of the text. The system design is such that these figures would not increase dramatically if more stereotypes were considered whilst processing the example.

6. References

[Cater80]
Cater, A.W.S. Analysing English Text: A Non-deterministic Approach with Limited Memory. AISB-80 Conference Proceedings. Society for the Study of Artificial Intelligence and the Simulation of Behaviour. July 1980.

[Charniak78]
Charniak E. With Spoon in Hand this must be the Eating Frame. TINLAP-2. 1978.

[DeJong79]
DeJong, G.F. Skimming Stories in Real Time: an Experiment in Integrated Understanding. Research Report #158. Yale University Department of Computer Science, New Haven, Connecticut. May 1979.

[Schank75a]
Schank, R.C. and the Yale A.I. Project. SAM -- A Story Understander. Research Report #43. Yale University Department of Computer Science, New Haven, Connecticut. 1975.

[Schank75b]
Schank, R.C. Conceptual Information Processing. North-Holland, Amsterdam. 1975.

[Wilks77]
Wilks, Y.A. Good and Bad Arguments about Semantic Primitives. Communication and Cognition, 10. 1977.

Generating Summaries Using a Predictive Language Analyser

Abstract:
The paper describes a computer system capable of producing coherent summaries of English texts even when they contain sections which the system has not understood completely. The system employs an analysis phase which is not dissimilar to a script applier together with a rather more sophisticated summariser than previous systems. Some deficiencies of earlier systems are pointed out, and ways in which the current implementation overcomes them are discussed.

Keywords and Phrases: natural language, summarising, predictive analysis, unpredicted utterances.

1. Introduction

There have been a number of recent attempts to build natural language processing systems which produce summaries of texts by recognising the topic of their input, exploiting a set of expectations about the contents of texts related to that topic to analyse the input, and then using the particular way their predictions were satisfied to fill out a template summary for such texts. Prime amongst such summarising systems is FRUMP ([DeJong79]). A severe disadvantage of these systems is that they cannot incorporate unexpected information in the input text into their summaries. This paper describes some techniques which can be used to incorporate such information into summaries whilst still retaining many of the advantages of predictive processing.

If the system's analysis processing is to remain primarily predictive, or top-down, its understanding of the unpredicted parts of the text will inevitably be rather shallow. This shallowness presents problems for the summary generation process since it must rely on linguistic, rather than world, knowledge to integrate unexpected text segments into the summary. The techniques adopted have also proved useful when generating summaries of texts which concern more than one topic.

The ideas presented here have been tested in computer system called Scrabble.

2. Summarising Predicted and Unpredicted Text Segments

It is the central contention of this work that those parts of a text which are unexpected are of interest precisely because they are unexpected; and that a good summary should reflect the contents of those parts of a text which are of most interest. Therefore a good summary should reflect (amongst other things) those parts of the text which were unexpected given the text's topic. Of course, the summary must also contain enough contextual information to form a complete, coherent and comprehensible text in itself.

A further contention is that a practical automatic summarising system should not produce summaries which misrepresent the input text, even at the cost of failing to reduce the volume of the original, or failing to produce a summary at all.

DeJong's FRUMP demonstrated that reasonably good summaries could often be produced (with great computational efficiency) using predictive text analysis techniques. It worked by selecting a template summary representation from amongst a fairly large number of possibilities using bottom-up processing techniques on the beginning of the input text. It then used predictions associated with the summary template to examine the input text for fillers for empty slots in the template in order to form a complete representation which could be converted into English, or a number of other natural languages. However, FRUMP was designed to completely skip unpredicted segments of the input text, and thus such segments never appeared in summaries; nor did sections of text which dealt with topics for which FRUMP had no suitable template or predictions. Furthermore it would often see into a text for which it had no suitable predictions a topic for which it was prepared, and hence produce an entirely misleading summary.

The SAM system described in [Schank75a] had essentially similar problems with unpredicted text segments.

It seems unlikely that we can give predictive analysers suitable sets of predictions for all the input texts they may meet in any but the most limited domains. Therefore devices are required to deal with unpredicted text segments in a motivated way if one is to use as one's primary means of condensing the original text techniques like those of FRUMP: that is the reduction of those parts of the input text to a brief indication that the input text deals with that topic, with some details of how it deals with it.

The approach adopted in Scrabble is as follows. All input material is examined and all unpredicted material is transferred from the analyser text representation to the summary representation as unreduced Conceptual Dependency structures (see [Schank75b]). (The CD-structures are extracted from the input text by a program described in [Cater82]). Together with indications of the textual and temporal relationships between predicted and unpredicted text segments, the filled-in summary templates (which are also CD-structures) constitute a framework into which the unpredicted material may be fitted. The initial summary representation thus formed may then be processed to produce a coherent English summary.

It is crucial for this approach that Scrabble does not take as its input natural language. Rather all input material is converted in CD-structures before Scrabble begins to process it. In particular, the final stage of processing, to convert the elements of the summary into a coherent whole, relies on the unpredicted text segments being presented in an easily manipulable form. CD-structures are admirable for this purpose, although there are other representation languages which would be equally suitable.

3. Producing the Summary

The Scrabble summariser employs a particular assumption about the coherence of the original text to perform the integration of the CD-structures representing the unpredicted parts of the input text with the CD-templates corresponding to the predicted parts.

This assumption is that if an object in the input is referred to loosely, for example by means of a pronoun, in a text segment, then if there is a closely preceding text segment with an identified topic, the object will co-refer with the central object of that topic, unless, of course, there is contrary information.

This rule is applied both to the unpredicted CD-structures and filled-in CD-templates derived from predicted portions of the input text, allowing the replacement of residual (CD) anaphora markers by more specific co-referring objects.(The templates have the likely central object of their associated topic marked). Thus in the example of the following section, once Fred and Bill have been identified as the central object of the visiting-the-zoo topic, they are assumed to be the central objects of the eating-at-a-restaurant topic as well.

The system then attempts to construct a suitable textual order for the now complete set of CD-structures and filled-in CD-templates of the summary representation. If there is sufficient temporal information in the input text, for example if it concerns scripty material, it will arrange the parts in the apparent temporal order in which the events described in the text occurred. Otherwise it chooses an order which is intended to reflect the order in which the material was presented in the original text.

Together with the rule for resolving anaphora the ordering rule allows reasonably natural output summary texts to be produced even though the system has not genuinely understood parts of the input text, at least for the (admittedly rather small number of) texts the Scrabble computer system has processed. Furthermore, the system operates without the (expensive) depth of processing used by [Lehnert81].

Two passes are now made over the ordered set of CD-structures and filled-in CD-templates.

First, they are blocked up into units which will form sentences in the output summary text. During the analysis of the input any unpredicted material is associated with the stereotype which is closest to it textually. So, for example, if an unpredicted text segment occurs between two segments predicted by the same stereotype [1] the unpredicted material will be associated with that stereotype. A variety of simple heuristics are used to deal with more complex cases. The summarising

---

[1] The use of the word stereotype here is the same as that in the previous paper in this report.

process examines this data structure and attempts to construct sentential CD-structures in which each filled-in CD summary template is conjoined its associated unpredicted CD-structures. For the first two such structures (in the textual order of the original text) a marker representing "but" is used; the unpredicted CD-structures are conjoined by a representation for "and" if necessary. Any additional unpredicted CD-structures are marked so that the corresponding sentences in the output summary occur directly after the sentence generated from the structure containing the filled-in CD-template.

Second, nominals are marked which should be pronominalized in the output where this would make the summary text more readable, for example if the subjects of consecutive sentences are the same.

Finally, the CD summary representation is converted into an English text. A program described in [Cater82] is used to do this.

4. An Example Text and Summary

Two topics are identified in the following text: a visit to a zoo, and eating at a restaurant. The method used for this identification is described in in this document. The parts of the text which the system did not expect to occur in texts concerning either topic are underlined. The parts predicted for the zoo topic are enclosed in curly brackets ({}). The parts predicted for the restaurant topic are in square brackets ([]).

The text:

> {Fred and Bill went to the zoo. They saw the elephants and fed the monkeys peanuts. After they had looked at the lions, [they went to the restaurant. They could see the zebra and giraffes from their table}. After they ate their meal] they realised they didn't have any money. They had to wash dishes [before they could leave].

Ignoring temporal information, the filled-in CD-template for the zoo topic is:

```
((EVENT
    (ACTOR GROUP#1)
    (ACT PTRANS)
    (OBJECT GROUP#1)
    (TO ZOO1)))
```

GROUP#1, whose English manifestation is "Fred and Bill", is marked as the central object of this topic. For the restaurant topic the filled-in template (again ignoring temporal information) is:

```
((EVENT
    (ACTOR DUMMY-UNKNOWNS11)
    (ACT PTRANS)
    (OBJECT DUMMY-UNKNOWNS11)
    (TO RESTAURANT1)))
```

DUMMY-UNKNOWNS11, which represents the "they" who went to the restaurant,

is marked as the central object of this topic.

The two unmatched clauses in the text are represented by:

```
((CAUSE
    (ANTECEDENT
       (EVENT
          (ACTOR DUMMY-HUMAN6)
          (ACT DO)
          (TIME
             (NAMED TIMEPOINT13)
             (COMPARISON (BEFORE *NOW*)))) )
    (RESULT
       (CAUSE
          (ANTECEDENT
             (EVENT
                (ACTOR DUMMY-UNKNOWNS10)
                (ACT PTRANS)
                (OBJECT *WATER*)
                (FROM DUMMY-PLACE4)
                (TO DISHES1)
                (TIME
                   (COMPARISON (AFTER TIMEPOINT8))
                   (NAMED TIMEPOINT13)
                   (COMPARISON (BEFORE *NOW*)))) )
          (RESULT
             (STATE
                (STATENAME CLEANNESS)
                (THING DISHES1)
                (VAL (HIGHERBY 3))
                (TIME
                   (NAMED TIMEPOINT15)
                   (COMPARISON
                      (AFTER
                         (NAMED TIMEPOINT13)
                         (COMPARISON (BEFORE *NOW*))))
                 ))) ))) )
```

representing "they had to wash dishes, and:

```
((STATE
    (STATENAME MLOC)
    (MOBJECT
       (STATE
          (STATENAME POSS)
          (THING MONEY1)
          (VAL DUMMY-UNKNOWNS9)
          (TIME
             (NAMED TIMEPOINT8)
             (COMPARISON (BEFORE *NOW*)))
          (TRUTH FALSE)))
    (CERTAINTY 8.1E-1)
    (INCP DUMMY-UNKNOWNS8)
    (TIME
       (NAMED TIMESPAN1)
       (COMPARISON (BEFORE *NOW*))
       (TS
          (COMPARISON (AFTER TIMEPOINT3))
          (NAMED TIMEPOINT8)
          (COMPARISON (BEFORE *NOW*)))) ))
```

representing "they realised they didn't have any money". Both the structures are associated with the restaurant topic, since they occur between two text segments expected to occur in texts about that topic.

These four summary elements are ordered so that the zoo template is first, the restaurant template second, then the structures representing "they realised they didn't have any money" and finally "they had to wash dishes". The two unpredicted text segments are placed in this order because it is the order in which they occur in the input text. All the remaining ordering is done on the basis of explicit temporal information in the input text.

Next, the system attempts to remove the residual anaphora markers from the summary representation. It is observed that DUMMY-UNKNOWNS11, which is plural and may refer to anything, can co-refer with the central object of the textually preceding topic, that is GROUP#1. Therefore GROUP#1 (representing "Fred and Bill" is placed in the filled-in template for the restaurant topic, displacing DUMMY-UNKNOWNS11. Next the two structures representing unpredicted parts of the input text are examined for anaphora markers which may co-refer with the central object of the restaurant topic, now GROUP#1. Three such markers are found, DUMMY-UNKNOWNS10, DUMMY-UNKNOWNS9 and DUMMY-UNKNOWNS8, each of which is replaced by GROUP#1.

Once this replacement has taken place two sentential structures are formed around the filled-in templates, and all the occurrences of GROUP#1 but the first are re-pronominalised. Finally the structures are passed over to the English generator producing:

> Fred and Bill went to a zoo. They went to a restaurant but they realised they that had no money and they had to wash dishes.

5. Conclusions

This paper has presented a method of exploiting the advantages of predictive language analysis when condensing text segments which make predictable statements about their topics. The method used does not have the disadvantage of previous systems, that material which was unusual or concerned topics for which the system was unprepared never occurred in the summary produced. Although this makes the system dependent on a sophisticated semantic parser, and therefore less robust than say, the system of [DeJong79], it will rarely produce a summary which seriously misrepresents the input text. Such avoidance of misrepresentation seems likely to be a requirement for any practical automatic summarising system.

Acknowledgements: I would like to thank Dr. B. K. Boguraev and H. Alshawi for reading and commenting on earlier drafts of this paper.

6. References

[Cater82]
Cater, A.W.S. Analysis and Inference for English. Technical Report No. 19. University of Cambridge Computer Laboratory, Cambridge. 1982.

[DeJong79]
DeJong, G.F. Skimming Stories in Real Time: an Experiment in Integrated Understanding. Research Report #158. Yale University Department of Computer Science, New Haven, Connecticut. May 1979.

[Lehnert81]
Lehnert, W.G. Plot Units and Narrative Summarisation. Cognitive Science Vol. 5 No. 4. pp. 293-331. 1981.

[Schank75a]
Schank, R.C. and the Yale A.I. Project. SAM -- A Story Understander. Research Report #43. Yale University Department of Computer Science, New Haven, Connecticut. 1975.

[Schank75b]
Schank, R.C. Conceptual Information Processing. North-Holland, Amsterdam. 1975.