COLING 82, J. Horecký (ed.)
North-Holland Publishing Company

FORWARD AND BACKWARD REASONING IN AUTOMATIC ABSTRACTING

Danilo FUM °, Giovanni GUIDA, Carlo TASSO
Istituto di Matematica, Informatica e Sistemistica, Universita' di Udine, Italy.
° Laboratorio di Psicologia E.E., Universita' di Trieste, Italy.

The paper is devoted to present a new approach to automatic abstracting which is supported by the development of SUSY, an experimental system currently being implemented at the University of Udine (Italy). The original contribution of the research reported is mostly focused on the role of forward and backward reasoning in the abstracting activity. In the paper the specifications and basic methodologies of SUSY are introduced, its architecture is illustrated with particular attention to the organization of the basic algorithms, and an example to support the novel approach proposed is described.

INTRODUCTION

For its theoretic and practical implications automatic abstracting has recently emerged as one of the most promising and interesting research topics in the field of natural language studies covered by computational linguistics, artificial intelligence, and psycholinguistics. In this paper we present the first results of a research project aimed at developing a new approach to automatic abstracting which is supported by the development of SUSY (SUmmarizing SYstem), an experimental system which is currently being implemented on VAX-11/780 at the University of Udine (Italy). The system is conceived to accept in input a natural language text (a scientific paper in the current application) together with the user's requirements and to produce as output a summary of the specified kind. SUSY relies on two basic assumptions:
- to ground parsing, summarizing, and generation activities mostly on the semantics of the language, and to avoid any kind of reasoning merely based on syntactic or structural properties which are not adequate for an intelligent and effective summarizer;
- to take strongly into account recent results of psycholinguistic research (Kintsch, 1974; Kintsch and van Dijk, 1978) as a conceptual background and a valid standpoint for designing a general purpose summarizing method.

The most relevant and original features of SUSY consist, in our opinion, in the remarkable flexibility of the system which allows the user to obtain different abstracts depending on his particular goals and needs, and in the strategies used to summarize (i.e., forward and backward processing) that simulate at a certain level of abstraction those utilized by humans.

SPECIFICATIONS AND BASIC METHODOLOGIES

In defining SUSY's specifications we have tried to implement at a certain level of abstraction an important human feature: the capability to generate summaries of different content and extent depending on the user's goals. The system is therefore able to process a text following the two principles of variable-length processing and of user-taylored abstracting. With variable-length processing we mean the capability to generate. starting from the same text. summaries of different length. complexity. and level of abstraction depending on the user's requirements. With user-taylored abstracting we mean the capability to generate summaries of different content depending on the user's goals and needs.

Together with the input text. SUSY can therefore receive in input the user's requirements describing with more or less details the organization. content. and extent of the output summary. This is done through a summary schema which can be interactively supplied at the beginning of the session. The user can also provide the system with a text schema which is constituted by a set of suggestions on how the input text can be interpreted. The text schema has a twofold motivation: to help the system in capturing from the input text only the most relevant parts. and to increase summarizing effectiveness.

Turning now our attention to the methodological aspects of SUSY. we notice that. in general. the summarizing activity can be performed in two distinct and complementary ways. The first one. or meaning-based. is grounded on the comprehension of the text to be summarized: in this case the summarizer has to capture the most important information contained in the text. The second possible way is structure-based and it does not rely on the meaning of the text but rather on its structure: the summary is obtained by eliminating. without understanding. parts of the text (for example adjectives. relative sentences. etc.) which a priori are considered less relevant. Both these ways can be combined with the two basic methodologies we have conceived for the system, i.e. forward and backward processing.

With the term forward processing we mean the capability to understand the whole natural language text and to produce in output. possibly through the iterative application of summarizing rules. the desired summary. This is clearly a bottom-up approach which constantly focuses on the input text. In backward processing. on the other hand. the focus is on the summary schema. The system works now top-down. searching for those parts of the text that can be utilized to build up the summary according to the specifications contained in the summary schema. In the SUSY sistem we have chosen to implement both forward and backward processing within a meaning-based approach.

SYSTEM ARCHITECTURE AND BASIC ALGORITHMS

The architecture of the system is organized in two main parts: the first one is devoted to collect the user's requirements and suggestions and to perform a preprocessing activity on them, the second one implements the actual parsing. summarizing. and generation activities.

The first part of the system constitutes an interactive interface centered around a main module called schema builder. This module is devoted to engage a bounded scope dialogue with the user in order to collect his suggestions about the structure and content of the texts to be summarized, and his requirements on the summary to be generated. This information is embedded in two different frameworks called working text schema and working summmary schema which contain the user's suggestions and requirements, respectively. The schemas will constitute a fundamental input for the following phases of the system operation. The working schemas are defined by the user, under the continuous guidance of the schema builder, through three different activities:

- choosing the most appropriate schema from a library of basic text and summary schemas or from a library of working text and summary schemas which contain the schemas utilised in previous summarizing sessions;
- tuning a selected schema by assigning (or reassigning) same parameters contained in it;
- defining a fully new (basic) schema.

It is understood that working schemas are not requested to be always defined at the same level of detail and completeness; they are allowed to embed more or less information according to the adequacy and richness of the specifications supplied by the user. For both text and summary schemas there exist default values to be utilized when the user is unable or unwilling to supply its own specifications.

The second part of the system is devoted to the parsing, summarizing, and generation activities. These are conceived in SUSY as three sequential steps which communicate through precisely defined data interfaces representing intermediate results of the processing.

The parser constructs the internal representation of the input text on which the summarizer will afterwards perform its activity. The operation of this module is based on a semantics-directed parsing algorithm which aims to supply a full understanding of the input text along the following two main lines:

- the text is parsed in a uniform way, independently of any expectation that could be possibly made (by considering the current working schemas) about the relevance of the different parts of the text in relation with the summary to be produced;
- the parsing is performed at a generally high level of abstraction, without decomposing objects into very elementary semantic primitives (Schank, 1975) but only considering the basic attributes and relations which are necessary for the summarizing task.

The semantics directed parsing algorithm utilises two kinds of information: the elementary knowledge about words and simple constructs contained in the vocabulary, and a set of semantic rules that specify the basic properties and relations of the elementary semantic entities which are supposed to play a role in the application domain in which the system operates (Guida and Tasso, 1982).

The internal representation constructed by the parser shares many features with that proposed by Kintsch (Kintsch, 1974; Kintsch and van Dijk, 1978) and is constituted by a sequence of labelled linear propositions each one conveying a unit of information. Every proposition is composed by a predicate with one or more arguments. Predicates and arguments can be considered as concepts or types to which the words in the input text (tokens) refer. The same type may be instantiated by different tokens which are therefore considered as synonims. Arguments can be types or labels of propositions and, in any case, they play precise semantic roles (agent, object, patient etc.). Every predicate imposes some constraints (linguistic or derived from the world knowledge possessed by the system) on the number and nature of its arguments. The proposions are connected to each other through shared terms in such a way to represent an actual network structure.

The activity of the summarizer has been split, according to the basic methodology illustrated in the previous section, in two sequential steps: a forward one performed by the weighter and a backward one implemented by the selector. The weighter is devoted to organize the internal representation, which is originally a flat and homogeneous network, into a structured framework in which the different levels of relevance and detail of the single propositions are clearly defined. This is obtained by assigning an integer weight to each proposition in such a way to generate a weighted network called weighted representation. The weighter utilizes for its operation the working text schema and a set of general purpose weighting rules. The selector is devoted to prune the weighted internal representation in such a way to obtain the selected representation i.e. the internal representation of the desired summary. It takes into account the working summary schema and operates through a set of general purpose selecting rules. The pruning it performs is generally not uniform with respect to the weights attached to the weighted representation, but it is biased and tuned by the requirements contained in the summary schema.

It is easy to recognize that weighting is indeed a forward activity which mainly focuses on the input text, while selecting represents a backward process which is generally directed by the consideration of the summary to be generated. Let us outline that the completeness and depth of the weighting and selecting activities strongly depend on the quality and richness of the text and summary schemas, respectively. Generally, these steps are not equally balanced and, in some cases, one of them may even be nearly void, as text schema or summary schema may be almost empty or even missed. In such cases we obtain a pure forward or backward strategy.

The last step of the system operation is the actual generation of natural language summary that is performed by the generator. Its activity is organised in two phases:

- retrieval from the input text of the basic linguistic elements (words, phrases, whole sentences etc.) necessary to compose the summary;
- appropriate assembly of these elements into a correct and acceptable text.

In the second phase it utilizes a set of sentence models which supply the most basic and usual rules for constructing correct sentences in a simple and plain style.

AN EXAMPLE

Owing to space restrictions we present in this section only a short working example of SUSY's performance, focusing on the most relevant features of the internal representation and of the weighting and selecting activities.

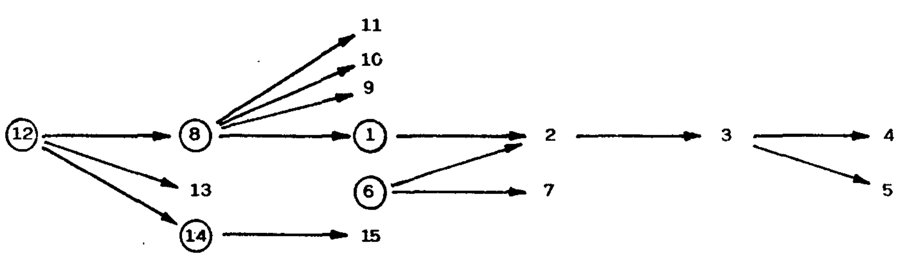
The input text in this example is a slightly adapted version of the first sentence of an article entitled "Fast Breeder Reactors" taken from Meyer (1975).

"The need to generate enormous additional amounts of electric power while at the same time protecting the environment is one of the major social and technological problems that our society must solve in the next future."

The parser maps this text into the internal representation:

1. NEED (2)
2. GENERATE ($,POWER)
3. QUANTITY OF (POWER.LOTS)
4. MORE (3)
5. ELECTRIC (POWER)
6. WHILE (2.7)
7. PROTECT ($,ENVIRONMENT)
8. PROBLEM (1)
9. BIG (8)
10. SOCIAL (8)
11. TECHNOLOGICAL (8)
12. MSOLVE (SOCIETY.8)
13. OUR (SOCIETY)
14. TIME OF (12,FUTURE)
15. NEXT (FUTURE)

The internal representation is then passed to the weighter in order to attach. following the suggestions contained in the text schema, an integer weigth to each proposition. As a result the weighted representation is obtained. which is graphically expressed as a network:



We mention here the three most relevant rules applied by the weighter to generate this network:

W.RULE1. IF a proposition i is referred to by a different proposition j, THEN assign weigths w such as $w(i) < w(j)$.

W.RULE2. IF the predicate of a proposition i is constituted by a modifier AND (the proposition i is referred to by a proposition j OR the proposition i has among its arguments one which has already appeared in a preceding proposition j) THEN assign weigths w such as $w(j) < w(i)$.

W.RULE3. IF a proposition i has among its arguments one which has already appeared in a preceding proposition j AND W.RULE2 is not applied. THEN assign weigths w such as $w(i) < w(j)$.

Let us note that modifiers in our approach are constituted by types that grammatically can be classified as adjectives or adverbials, and types such as TIME OF, QUANTITY OF, LOCATIVE OF and so on.

The weighted representation is then supplied to the selector which chooses a certain number of propositions that will constitute the selected internal representation of the summary and will be passed to the generator in order to produce the final output summary. This choice is driven by the specifications contained in the summary schema.

In our example. through the application of the selecting rule:

S.RULE1 Choose the n most weighted propositions discarding the leaves.

where n is a parameter which takes into account the length of the desired summary (in the example, n=5), we can select the propositions that appear encircled in the network.

These propositions are eventually passed to the generator which gives the final output summary:

"The society must solve in the future the problem of the need to generate power while protecting the environment."

The specifications given by the user through the text schema and the summary schema may of course activate different weighting and selecting rules and thus generate different summaries.

CONCLUSION

At the end of the paper. let us mention some of the most promising research directions for future activity:

- to develop a new parsing algorithm which. taking into account the text and summary schemas. allows the generation of a variable-depth internal representation;
- to implement a more advanced weighter which attaches weights not only to propositions but also to their elementary components;
- to expand the knowledge representation method adopted for constructing the internal representation into a more sophisticated language suitable to express. whenever requested. very elementary semantic primitives which allow limited deduction and reasoning capabilities.

REFERENCES

1. GUIDA, G. and TASSO, C., NLI: A robust interface for natural language person-machine communication. International Journal of Man-Machine Studies (1982), in press.
2. KINTSCH, W., The Representation of Meaning in Memory (Lawrence Erlbaum Ass., Hillsdale: N.J., 1974).
3. KINTSCH. W. and VAN DIJK, T., Toward a model of text comprehension and production. Psychological Review. 85 (1978) 363-394.
4. MEYER. B., The Organization of Prose and Its Effects on Memory (North-Holland, Amsterdam, 1975).
5. SCHANK, R.C. Conceptual Dependency Theory (North Holland. Amsterdam, 1975).