Representing Knowledge for Planning Multisentential Text

Jose Coch, Raphael David GSI-Erli / 1, place des Marseillais 94227 Charenton-le-Pont Cedex, FRANCE e-mail: jose.coch@erli.gsi.fr, raphael.david@erli.gsi.fr

Abstract

The AlethGen tool has been used to build a system for automatically generating replies to complaints, for a European mail-order company. In the current version, letters are generated in French.

AlethGen consists of several modules: the text deep structure planner (or conceptual planner), the text surface structure planner (or rhetorical planner), the computation of pronouns and quantifiers, and the linguistic generator proper (globally inspired by the Meaning-Text Theory) which determines the concrete realization of sentences.

The paper mainly describes AlethGen's conceptual and rhetorical planners and focuses on the knowledge they use and produce.

1. Goals and general approach

The main function of the texts we had to generate consisted indeed in replying to a complaint, thus in arguing and justifying a decision. This implies, on the one hand, a good management of the argumentative aspects, and on the other hand, the need to generate rather complex texts in a perfectly consistent style.

The general approach of the conceptual planning is relation-based, as described in the Rhetorical Structure Theory or RST (Mann & Thompson, 1988).

However, as shown in (Elhadad & Mc Keown, 1990), previous works in text generation systems (including RST) "have generally used a notion similar to rhetorical relations to describe the connection between propositions. They make a one-to-one mapping from these relations to connectives for generation (for example, the "opposition" relation would be realized by the connective "but"). In this approach it is difficult to distinguish between similar connectives (e.g. [...] but vs. although)."

In our work, we assume that there are two different levels of text structure. The **deep level** is represented by unordered basic relations, and the **surface level** is represented by (ordered) lists of atomic events and rhetorical operators.

2. Example

Here is a letter generated by AlethGen and its translation in English:

Cher Monsieur,

J'ai bien reçu votre message minitel du 22 mars 1993 concernant la commande du disque XXCC et du logiciel YYBB. Le disque a été expédié le 3 février 1993 par la Poste, il aurait donc dû parvenir à votre domicile. Je vais me renseigner pour savoir ce qui a pu se produire.

Je ne peux pas renouveler cette commande car malheureusement, nous ne pouvons pas vous livrer avant le mariage. Dans votre intérêt, je préfère annuler cette commande. Elle ne fera l'objet d'aucun remboursement puisque vous aviez prévu de régler à la livraison.

Je vous prie, Cher Monsieur, ...

Dear Sir,

I have received your Minitel message, dated March 22, 1993, concerning the order of an XXCC disc and a YYBB software. The disc was sent by post on February 3, 1993; you should therefore have received it by now. I will make inquiries to find out what may have happened.

Unfortunately I cannot renew the order, as we are unable to deliver before the wedding. In your interest, it would be preferable that I cancel your order. It will not be refunded as you had decided to pay on delivery.

Yours faithfully,

In fact, the customer has sent a complaint letter because he has not received the items he had ordered. However, he maintains his order provided he receives the items before his wedding. The items that were sent are considered lost because of the dates. As the wedding is to be celebrated soon, it will not be possible to get the items before that date. The order must therefore be cancelled.

The deep text structure is a list of two trees (in fact the introduction and the last phrases of the letters are computed from the input with a trivial system):

• CONSEQUENCE (

BECAUSE (delivery_parcel: modality=past counterfactual,

parcel_sent: lost),

inquiry)

 CONSEQUENCE (condition_fulfillment_impossible, CONSEQUENCE (AND (NOT renewal_of_the_order, cancellation), no_refund))

The surface text structure is a list of events with rhetorical features and surface operators (note that the English (human) translation of the letter has a slightly different surface text structure).

parcel_sent:

```
lost

parcel_delivery:

modality1=past counterfactual

operator="donc" ("therefore/so")

inquiry

<new paragraph>

non renewal_of_the_order

"car" ("because/as")

condition_fulfillment_impossible

modality2=ne_pas_pouvoir (to be unable)

modality1=malheureusement (unfortunately)

cancellation

modality1=préfère (better/preferable)

no_refund
```

3. Knowledge

3.1. Rules for conceptual planning

If p and q are events, one can have a rule (cause p consq q) in the knowledge base. If the input has p & q, one can generate something like "p caused q" or "q, because p". All the possible values are given in the following table:

р	q_	one can generate	
Т	T	"q because p"	
F	F	nothing (does not apply)	
F	Т	nothing (does not apply)	
Τ	F	"~q though p"	

The last line is the "exception" to the rule.

Rules can be more complex, and have a condition in addition to the cause: (cause p cond r consq q). For example,

cause: addressee_wants_cancel (his order)

cond: not_delivered (the articles have not been...) consq: cancellation (of the order)

Clearly, not_delivered is not a <u>cause</u> for cancellation but only a <u>condition</u>, because the reason for cancellation is addressee_wants_cancel.

р	r	q	one can generate
Τ	T	Т	"q because p"
Τ	F	Т	"q because p though ~r"
F	T/F	T/F	nothing (does not apply)
Т	T	F	"~q though p"
Т	F	F	"~q though p because ~r"

So the knowledge for planning the deep structure of the text is a set of domain-dependent conceptual rules.

3.2. Rhetorical rules

Given a simple relational tree consisting of an operator as well as two events, it is easy to see that several surface orders and several surface operators are possible. For instance, the tree BECAUSE (You ask me for X, I see to it so that X) may have the following linguistic realizations:

- I see to it so that X, because you asked me for Y

- You asked me for Y, so I see to it so that X.

The rhetorical module chooses concrete operators, modalities, and order according to certain attributes, e.g. the fact that the addressee is aware of an event or not, the fact that it is in favour of the addressee or not, etc.

4. Implementation

The system was written in C++ under Unix and effectively runs on RS/6000 and Sun SparcStation.

In the current version, there are 130 conceptual rules for 50 models of events, and 75 rhetorical rules.

In any case, the system generates a whole letter on a complex problem (15-20 sentences) in less than 2,5 seconds.

5. Conclusion

The distinction between deep and surface structure of the text allows the system to generate relevant letters from an argumentative point of view.

Besides, letters may deal with a rather large number of various topics (payment of lost or damaged items, etc.). Our concern was therefore to factor the information as much as possible, in other words we wanted to make certain pieces of knowledge portable. For that purpose, it is interesting here to remind the distinction between domain knowledge and domain communicative knowledge (DCK) proposed in (Kittredge and al., 1991). For example a police crime report and a detective novel can use the same domain knowledge but the DCK is quite different. We think that it is preferable to call this knowledge "Style Dependent Knowledge", because two detective novels in two different "domains" (for example a political murder and a crime of passion) could have the same style.

In our system, conceptual rules are clearly domaindependent. The communicative knowledge used by the rhetoric module to compute the surface structure of the text is style-dependent but it is domain-portable.

References

Cerbah Farid (1992). "Integrating qualitative reasoning and text planning to generate causal explanations" in *Proceedings of 1992 International Computational Linguistics Conference.*

- Coch José, and David Raphaël (1994). "Une application de génération de textes" in Actes de TALN'94. France.
- Elhadad Michael, and Mc Keown, Kathleen. R. (1990). "Generating connectives" in *Proceedings of 1990 International Computational Linguistics Conference*.
- Elhadad Michael (1992). "Generating coherent argumentative paragraphs" in *Proceedings of 1992 International Computational Linguistics Conference*.
- Kittredge Richard, Korelsky Tanya, and Rambow Owen (1991) "On the need for domain communication knowledge" in *Computational Intelligence* V7, N4, Canada.
- Mann William C. and Thompson Sandra. A. (1988). "Rhetorical Structure Theory: Towards a functional theory of text organization" in *Text* 8(3).