Generating Dialogues for Virtual Agents Using Nested Textual Coherence Relations

Hugo Hernault[1,3], Paul Piwek[2], Helmut Prendinger[1], and Mitsuru Ishizuka[3]

[1] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
hugo@nii.ac.jp, helmut@nii.ac.jp
[2] NLG Group, Centre for Research in Computing
The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
p.piwek@open.ac.uk
[3] Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
ishizuka@i.u-tokyo.ac.jp

Abstract. This paper describes recent advances on the Text2Dialogue system we are currently developing. Our system enables automatic transformation of monological text into a dialogue. The dialogue is then "acted out" by virtual agents, using synthetic speech and gestures. In this paper, we focus on the monologue-to-dialogue transformation, and describe how it uses textual coherence relations to map text segments to query–answer pairs between an expert and a layman agent. By creating mapping rules for a few well-selected relations, we can produce coherent dialogues with proper assignment of turns for the speakers in a majority of cases.

1 Introduction

The task of automatically generating dialogues is a crucial step for the wide dissemination of agent-based multimodal presentations. Previous research has aimed at generating dialogues using various inputs. For example, in the systems described in [1] and [2], dialogues are generated from structured data (such as product databases and story plans, respectively).

However, the ever-growing collection of textual information found on the Web makes text an input of choice given its quantity, its availability, and the diversity of its content. Early work on dialogue based on text can be found in the automatic information presenters Web2TV and Web2TalkShow [3], where the output dialogue is intended to be humorous (exaggerated and distorted).

In our approach, the Text2Dialogue system based on [4], we want to implement the following three features:

- The input is plain, un-formatted text rather than a knowledge base.
- The transformation is meaning-preserving rather than intentionally humorous.
- The output is dialogue which is specified in a format that can drive the performance of a team of computer-animated virtual agents.

H. Prendinger, J. Lester, and M. Ishizuka (Eds.): IVA 2008, LNAI 5208, pp. 139–145, 2008.

The two interlocutors performing the dialogue are rendered as virtual agents, and assume the roles of expert (e.g. instructor) and layman (e.g. student). Presenting information as a dialogue is a popular means of conveying information. It can be witnessed in commercials, edutainment, and even video games.

Fig. 1. Two virtual agents having a dialogue in the "Second Life" 3D online virtual world

The Multimodal Presentation Markup Language MPML3D [5], based on MPML [6], is used to control the verbal and non-verbal behavior of the agent characters. Non-verbal behavior includes body gestures, posture, and eye gaze. Fig. 1 shows two MPML3D-controlled agents conversing in "Second Life", a 3D online virtual world.

In brief, there are two reasons why our approach is worth exploring. Firstly, it enables users – as content creators – to conveniently generate powerful multimodal presentations, and secondly, the system enables users – as content consumers – to "learn by observation" (vicarious learning). Studies such as those described in [7] have shown that dialogues convey information efficiently, and that one can learn effectively by watching them.

This paper presents a significant extension of previously published work [4] in that it introduces a method for generating dialogues from not just flat but also nested coherence relations in text.

The rest of the paper is organized as follows: The next section gives an overview of how the components of our system work. We describe the mechanisms used in the process of generating dialogues. Then, we address the issue of assigning turns to speakers. Finally, we briefly report on the results of a preliminary evaluation of the characteristics and quality of our generated dialogues.

2 The Text2Dialogue System

2.1 Text Analysis Component

First, our system uses a discourse analyzer to determine the coherence relations underlying the input text, i.e., the text's discourse or rhetorical structure. Compatible parsers are DiscourseAna [8] and SPADE [9]. Those parsers analyze the input text in terms of a specific theory of coherence relations known as Rhetorical Structure Theory (RST) [10]. In RST, a text is segmented into non-overlapping spans, connected by arrows that indicate the rhetorical (i.e., coherence) relations between the different spans. The endpoint of the arrow represents the "nucleus"

(the more important argument of the relation) whilst the origin represents the "satellite" (containing less important information).[1]

Here is a text snippet followed by its associated RST representation:

[Mexico exported an average of 1,296,800 barrels of crude oil a day at an average of $15.31 a barrel during 1989's first eight months for a total of $4.82 billion,]^{1A} [Petroleos Mexicanos S.A. said.]^{1B} [The state petroleum monopoly said]^{1C} [sales in the period gained 15%, and $262.4 million more than originally projected at an average of $10 a barrel on an export platform of 1,250,000 barrels a day.]^{1D} (wsj_1104 from the RST Discourse Treebank)

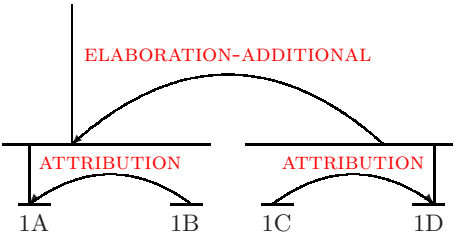

Fig. 2. RST representation of example text

2.2 Mapper Component

For each rhetorical relation, we define one or more mapping rules that determine how to generate a corresponding fragment of dialogue. For the ATTRIBUTION relation, one of the most common relations making up 13.87% of all relations in the RST Treebank [11], we created the following rule:

MAPPING RULE: **Attribution, principal rule**

ATTRIBUTION$(P, Q) \Longrightarrow$
L: *What did + getSubject(P+Q) + getMainVerbLemma(P+Q)?*
E: *AddIfNotPresentIn(Q, That) + Q*

An example dialogue generated from the sentence ATTRIBUTION("John said", "he likes apples.") would be: "L: What did John say? E: That he likes apples."

In an RST structure, relations are nested most of the time, and we have to find systematic rules to decide how and when to perform a mapping. For instance, ATTRIBUTION takes plain text for P and Q, and therefore the mapping rule given above cannot be applied if there are other mapping rules that need to be applied to the rhetorical relations underlying P or Q. Thus, we define the notion of *relation-incompatibility* to express that certain mapping rules can not be applied at the same time:

[1] RST also has multi-nuclear relations, i.e., relations in which none of the arguments is more prominent than any of the other.

> Relation-incompatibility: We say that a relation R_1 is incompatible with a relation R_2 if, when R_1 is a child of R_2 in a RST tree, the mapping rules for R_2 cannot be applied.

There are more than 50 rhetorical relations [11]. Some of them, such as Elaboration-Additional and List are very common, whereas others (such as Topic-Shift or Analogy) are more specific and occur rarely. Defining mapping rules for all relations would be a time consuming task. Instead we need a way to classify and compare relations, in order to decide which are more essential for the dialogue. Therefore, we define the notion of *relation importance*:

> Relation importance: We use a function $Imp : \mathcal{R} \rightarrow \mathbb{N}$ that associates a natural number to all relations. Unimplemented relations are given the importance 0.

This allows us to compare relations based on their importance, and enables us to filter out non-important relations: unimplemented and secondary relations will not be mapped as they have the lowest importance. Depending on the domain of application of our system, we might want to emphasize specific relations. Setting a high importance for these relations will ensure that they are always mapped. Finally, this mechanism allows us to tune the verbosity of the system. For instance, when all relations are given the same priority, they will all be mapped to query–answer pairs.

With the notions of relation importance and relation compatibility defined, we are now in a position to describe the mapping algorithm of our system. Our dialogue mapper parses the RST tree in a top-down fashion and maps the text segments into question-answer pairs. At each recursive call, we decide to perform mapping of a child relation if

1. The child is more or as important as its parent relation, and
2. The child is not incompatible with its ancestors, and
3. The child relation is implemented (i.e., has at least one mapping rule).

Else, if no mapping is performed, the child relation's subtree will be flattened as a single line of text and spoken by the expert.

The algorithm, when applied to the tree represented in Fig. 2, returns:

a. *L*: What did Petroleos Mexicanos S.A. say?
b. *E*: That Mexico exported an average of 1,296,800 barrels
c. of crude oil a day at an average of $15.31 a barrel
d. during 1989's first eight months for a total of $4.82 billion.
e. *E*: Should I tell you more?
f. *L*: Yes, please.
g. *L*: What did the state petroleum monopoly say?
h. *E*: That sales in the period gained 15%, and $262.4 million
i. more than originally projected at an average of $10 a barrel
j. on an export platform of 1,250,000 barrels a day.

2.3 Assigning Turns to Speakers

In the previous dialogue, we can notice some discrepancies due to improper turn assignment between the two interlocutors (see the transition from line f. to g.). Indeed, when the mapping produced by a relation ends with a certain agent asking a question, the other agent is expected to produce an answer. A solution is to define an alternate-speaker rule for each relation. For instance, our alternate-speaker rule for the ATTRIBUTION relation is:

MAPPING RULE: **Attribution, alternate-speaker rule**

ATTRIBUTION$(P, Q) \Longrightarrow$
E: RemoveIfPresentIn(Q, That) + Q
L: Who getMainVerbFromSentence(P+Q) + that?
E: getSubjectFromSentence(P+Q) +
generateWordForm(do, getMainVerbMorphoTagsFromSentence(P+Q))

The problematic passage in the previous dialogue is now resolved as:

f. *L*: Yes, please.
g. *E*: Sales in the period gained 15%, and $262.4 million
h. more than originally projected at an average of $10 a barrel
i. on an export platform of 1,250,000 barrels a day.
j. *L*: Who said that?
k. *E*: The state petroleum monopoly did.

2.4 Preliminary Evaluation

To evaluate our system's "Mapper Component", we applied it to RST-hand-annotated texts from the RST Treebank [11] and collected statistics. We implemented mapping rules for 3 of the most common relations: ELABORATION-ADDITIONAL (2 rules), ATTRIBUTION (2 rules), CIRCUMSTANCE (1 rule). We gave the same importance to all 3 relations. We ran the system on all 347 annotated newspapers articles contained in the RST Treebank, and obtained the following results:

31.8% of relations in the text are mapped. This is almost equal to the cumulated occurence rate of the 3 implemented relations, which make up 33.3% of all relations. The relatively small difference can be explained by the fact that nested relations sometimes lead to incompatibilities that prevent mapping.

The mean length of a turn in the resulting dialogues is 13.28 words (SD=32.28, median=5.0). For the layman specifically, the mean length of a turn is 2.71 words (SD=1.42, median=2.0). For the expert, the mean length of a turn is 18.88 words (SD=38.77, median=12.0). The dialogues generally have a reasonable size, and the layman's turns are short because they typically consist of sentences such as "Yes, please." or "What else?". Looking at the number of turns per dialogue, we found that the mean was 46.98 (SD=43.39, median=32.0). In summary, on average each text was rendered as a dialogue consisting of a good number of turns of a reasonable length.

In order to also evaluate the quality of the dialogues, we manually analyzed a sample of randomly selected generated dialogues. On 100 extracts of our generated dialogues, the first author evaluated their quality based on 1) whether the dialogues were syntactically sound, 2) whether they conveyed the meaning of the original text, and 3) whether they made sense when taken in context of the surrounding dialogue. The results were: 87% of our dialogues fulfilled 1), 75% fulfilled 1) and 2), and 60% satisfied 1), 2), and 3).

3 Conclusions

We have presented the latest advances on our Text2Dialogue system. Using abstract mapping rules and an algorithm based on relation incompatibility and relation importance, we can automatically generate dialogues from text. While the first version of the system [4] was confined to non-nested relations, the version described in this paper can also handle nested rhetorical structures in text. Our algorithm deals with incompatible nested relations by using a preference ordering on relations to decide which relation to realize. The problem of properly assigning turns to speakers has been addressed using alternate-speaker rules for each relation. Our evaluation of the latest prototype indicates that the proposed algorithm results in reasonably organized dialogue for a majority of the test inputs.

In the future, we will be working on (1) implementing more mapping rules for frequently occurring relations, (2) improving the syntax of the generated dialogues, and (3) investigating how to improve the coherence of the generated dialogues by considering context.

Acknowledgements

The first author was supported by an International Internship Grant from NII under a Memorandum of Understanding with the Institut National Polytechnique de Toulouse, and a “Strategic Project” Grant from NII.

References

1. André, E., Rist, T., van Mulken, S., Klesen, M., Baldes, S.: The automated design of believable dialogues for animated presentation teams. In: Embodied Conversational Agents, pp. 220–255. MIT Press, Cambridge (2000)
2. Cavazza, M., Charles, F.: Dialogue generation in character-based interactive storytelling. In: Proceedings of the AAAI First Annual Artificial Intelligence and Interactive Digital Entertainment Conference, Marina Del Rey, California, USA (2005)
3. Nadamoto, A., Tanaka, K.: Complementing your TV-viewing by web content automatically-transformed into TV-program-type content. In: Procs. 13th Annual ACM Intl. Conf. on Multimedia, pp. 41–50. ACM Press, New York (2005)

4. Piwek, P., Hernault, H., Prendinger, H., Ishizuka, M.: T2D: Generating dialogues between virtual agents automatically from text. In: Pélachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 161–174. Springer, Heidelberg (2007)
5. Nischt, M., Prendinger, H., André, E., Ishizuka, M.: MPML3D: A reactive framework for the Multimodal Presentation Markup Language. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 218–229. Springer, Heidelberg (2006)
6. Prendinger, H., Descamps, S., Ishizuka, M.: MPML: A markup language for controlling the behavior of life-like characters. Journal of Visual Languages and Computing 15(2), 183–203 (2004)
7. Craig, S., Gholson, B., Ventura, M., Graesser, A.: the Tutoring Research Group: Overhearing dialogues and monologues in virtual tutoring sessions. Intl. Journal of Artificial Intelligence in Education 11, 242–253 (2000)
8. Le, H.T., Abeysinghe, G.: A study to improve the efficiency of a discourse parsing system. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 101–114. Springer, Heidelberg (2003)
9. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Procs. HLT/NAACL 2003, Edmonton, Canada (2003)
10. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243–281 (1988)
11. Carlson, L., Marcu, D.: Discourse tagging reference manual. Technical Report ISI-TR-545, ISI (September 2001)