



Open Research Online

Citation

Piwek, Paul; Hernault, Hugo; Prendinger, Helmut and Ishizuka, Mitsuru (2007). T2D: Generating Dialogues Between Virtual Agents Automatically from Text. In: Intelligent Virtual Agents (Pelachaud, Catherine; Martin, Jean-Claude; Andre, Elisabeth; Chollet, Gerard; Karpouzis, Kostas and Pele, Danielle eds.), Lecture Notes in Artificial Intelligence, Springer, Berlin Heidelberg, pp. 161–174.

URL

<https://oro.open.ac.uk/12182/>

License

None Specified

Policy

This document has been downloaded from Open Research Online, The Open University's repository of research publications. This version is being made available in accordance with Open Research Online policies available from [Open Research Online \(ORO\) Policies](#)

Versions

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding

T2D: Generating Dialogues between Virtual Agents Automatically from Text

Paul Piwek¹, Hugo Hernault², Helmut Prendinger², and Mitsuru Ishizuka³

¹ NLG Group, Centre for Research in Computing
The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
p.piwek@open.ac.uk

² National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
hugo@nii.ac.jp, helmut@nii.ac.jp

³ Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
ishizuka@i.u-tokyo.ac.jp

Abstract. The Text2Dialogue (T2D) system that we are developing allows digital content creators to generate attractive multi-modal dialogues presented by two virtual agents—by simply providing textual information as input. We use Rhetorical Structure Theory (RST) to decompose text into segments and to identify rhetorical discourse relations between them. These are then “acted out” by two 3D agents using synthetic speech and appropriate conversational gestures. In this paper, we present version 1.0 of the T2D system and focus on the novel technique that it uses for mapping rhetorical relations to question–answer pairs, thus transforming (monological) text into a form that supports dialogues between virtual agents.

1 Introduction

Information presentation in dialogue format is a popular means to convey information effectively, as evidenced in games, news, commercials, and educational entertainment. Moreover, empirical studies have shown that for learners, dialogues often communicate information more effectively than monologue (see e.g. [5, 6]). The most well-known use of dialogue for information presentation is probably by Plato: in the Platonic dialogues, Socrates and his contemporaries engage in fictitious conversations that convey Plato’s philosophy. A more recent example is Douglas Hofstadter, whose Pulitzer prize winning book *Gödel, Escher, Bach* [8] consists of chapters which are each preceded by a dialogue that explains and illuminates concepts from mathematical logic, philosophy or computer science. Most information, however, is not available in the form of dialogue. Presumably the most common way of representing information is (monological) text, for instance on the web, where textual information is abundant in quantity and diversity. Moreover, huge amounts of information are captured in databases and, with the advent of the semantic web, ontologies.

The preparation of attractive and engaging multi-modal presentations using a team of virtual agents is a time-consuming activity that requires several skills regarding: (1) How to generate a coherent, meaningful dialogue; (2) how to assign appropriate gestures to the conversing agents; and (3) how to integrate media objects illustrating the dialogue into the presentation. Currently, most of these tasks can only be performed by a trained dialogue script writer. The wide dissemination of digital media content using life-like characters, however, would greatly benefit from an authoring tool that supports non-experts (for dialogue script writing) in generating multi-modal content.

In this paper, we focus on the issue (1) of generating coherent dialogue, and assume (monological) text as the input to dialogue generation. The next section provides an overview of and comparison with related work in this area. We then proceed to a description of version 1.0 of our implemented T2D system (Section 3). We relate the design of the system to a set of requirements that include robustness and extensibility. In Section 4 a walk-through example is described that illustrates how the system operates. Finally, Section 5 presents our conclusions and issues for further research.

2 Related Work

There are a number of studies that deal with the problem of automatically generating multi-modal dialogues between life-like animated agents. These differ, however, in the type of input they require and the techniques that are employed to map the input to multi-modal dialogue.

In Intelligent Multimedia Presentation (IMMP) systems the authoring process is automated by employing methods from artificial intelligence, knowledge representation, and planning (see [1] for an overview). An IMMP system assumes a so-called “presentation goal” and uses planning methods to generate a sequence of presentation acts. The generation of a presentation is based on dedicated information sources that encode information about presentation content and objects [2]. The difference with our proposal is that we do not require the formulation of planning operators, which assumes a background in artificial intelligence. Our proposal is solely based on existing material (currently text and, in future, possibly also associated graphics), and thus easy-to-use by non-experts and not suffering from the knowledge representation bottleneck.

Recently developed related systems include Web2TV and Web2Talkshow [12], and e-Hon [18].⁴ Web2TV uses two animated characters to readout a given text in a TV-style environment. Web2Talkshow transforms a (summary) of text

⁴ Here, we do not review work on tutorial dialogue systems. Some of the work in that area focuses on authoring tools for generating questions, hints, and prompts. Typically, these are, however, single moves by a single interlocutor, rather than an entire conversation between two or more interlocutors. Some researchers have concentrated on generating questions together with possible answers (i.e., multiple-choice test items), but this work is restricted to a very specific type of question-answer pairs (see, e.g., [11]).

from the web into a humorous dialogue between character agents. e-Hon transforms text into an easy-to-understand dialogue based on rephrasing content, and enriching it with animations. Web2Talkshow and e-Hon on the one hand, and our T2D system on the other, are similar in that they both aim to generate dialogues automatically from text. The differences lie in how text is mapped to dialogue: Firstly, Web2Talkshow and e-Hon analyze single sentences as the basis of the generated dialogue. E.g., Web2Talkshow takes declarative sentences of the form *X of the Y did Z* and transforms them into dialogue fragments of the form *A: Who is X. B: I know. X is one of the Y. A: That's right! He did Z*. The system looks for keywords called subject and content terms [12] that can fulfill the role of *X*, *Y*, and *Z*. Keywords are identified based on frequency counts and co-occurrence statistics, and presumably intended to reflect what the document is about. As a result, the approach seems to be based on, what (in linguistics) is called the *information structure* of a sentence. Information structure is orthogonal to *discourse structure*. The latter focuses on *relations* between spans of text (such as evidence, condition, justification, etc.), rather than aboutness, and applies both within and across sentence boundaries. T2D uses discourse rather than information structure to create dialogues. Secondly, whereas our aim is to faithfully render the content of the input text as a dialogue, a feature of Web2Talkshow is that it generates humorous dialogues, exploiting distortions and exaggerations of what is actually said in the input text. Furthermore, our approach is underpinned by systematic tests on a corpus of Patient Information Leaflets to verify that the mappings performed by T2D are indeed meaning-preserving and result in linguistically well-formed dialogues.

The investigations on automated generation of scripted dialogues described in [16, 14] provided some of the foundations for the current work. That research also investigates the combination of information from sources other than text. In one scenario [15], the principal information is an electronic health record, and supplementary information is drawn from thesauri, wikis, and ontologies.

3 System Description

The main starting point for our system is that it should be usable by non-experts to create multi-modal dialogue from text. We identify three requirements for such a system: robustness, extensibility, and variation/control.

Firstly, the system should be able to produce a dialogue regardless of the input text. In other words, the system should be ROBUST. Secondly, the system should be EXTENSIBLE. A given input text will normally be realizable as more than just one single dialogue. Since the general task of mapping text to dialogue is a very difficult one, any current system is unlikely to cover all possible mappings from text to dialogue. The system should, however, be easily extensible in order to cover new mappings. It should be straightforward to add new mappings and replace parts of the system, as and when new technologies and techniques for particular subtasks become available (e.g., text segmentation and discourse parsing). Finally, we require that our system allows for VARIATION and CON-

TROL of its outputs: An output dialogue should not contain repetitive structures that make it less appealing (e.g., ‘conversational ping-pong’ [7]). Ideally, choices for specific forms of expression should depend on the context and the purpose for which the dialogue is used. Here we will only discuss some very preliminary attempts to introduce variation, and leave issues of control for future work.

3.1 System design

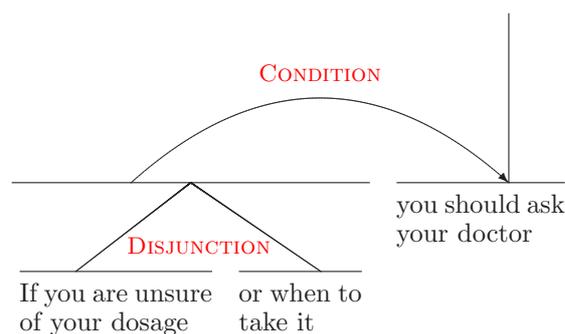
The system consists of three principal components:

1. **ANALYZER**: A component that analyses text in terms of Rhetorical Structure Theory (RST, [10]). Currently, it consists of the DAS Discourse Analyzing System [9] which builds RST structures (but without identifying nuclei and satellites), and a nucleus/satellite Identification Module;
2. **MAPPER**: Module that maps RST structures to DialogueNet structures (these are a specific subclass of RST structures that represent dialogue);
3. **PRESENTER**: A Module for translating DialogueNet structures to the Multimodal Presentation Markup Language (MPML3D) format [13]. MPML3D script specifies multi-modal dialogue performed by two 3D agents.

Both components (1) and (3) are partly of-the-shelf systems that can in principle be replaced with alternative solutions for discourse analysis and multi-modal presentation. Representations between components are exchanged in XML format. All this contributes to the **EXTENSIBILITY** of the system.

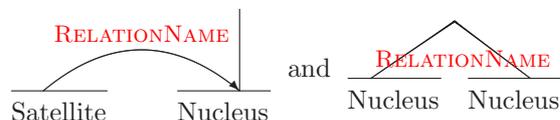
At the heart of the system sits the mapper from RST structures to DialogueNet structures. In the remainder of this section we introduce both RST and DialogueNet structures and then focus on the theoretical foundations underlying the T2D approach to mapping between such structures.

Rhetorical Structure Theory RST is the most widely used descriptive theory of discourse structure. A text is presumed to be segmented into units, e.g., independent clauses, and these occupy the terminal nodes in an RST structure. For example, the text ‘If you are unsure of your dosage or when to take it, you should ask your doctor’ receives the following analysis in RST:⁵



⁵ Reitter’s rst L^AT_EX package [17] is used for displaying RST trees.

This structure is built up of two relations that are instances of the following two generic schemas for building RST structures [10]:



In the schema on the left-hand there is a difference in status between the items that are related: one is more essential or prominent than the other. The more important span – the *nucleus* – is distinguished graphically from the *satellite*, by being the endpoint of the arrow. A relation with a distinguished nucleus is known as mononuclear. The notion of nuclearity plays a central role in the operational definitions of discourse relations. For example, *CONDITION* is defined in [4, p. 51] as: ‘In a *CONDITION* relation, the truth of the proposition associated with the nucleus is a consequence of the fulfillment of the condition in the satellite. The satellite presents a situation that is not realized.’ The schema on the right-hand applies to relations that do not have a single most prominent item. For example, *DISJUNCTION* is ‘[...] is a multinuclear relation whose elements can be listed as alternatives, either positive or negative.’ [4, p. 53].

DialogueNet DialogueNet (henceforth DN) structures are the subclass of RST structures that satisfy the following definition:

DEFINITION *DN Structure*: An RST structure R is a DN structure for a text T , if and only there exists a partitioning of T into a set of non-overlapping spans $\{T_1, \dots, T_n\}$, such that this set consists of pairs of spans $\langle T_x, T_y \rangle$ which are related in R by the RST *ATtribution* relation, with T_x the satellite of the *ATtribution* relation, in particular, a clause of the form *Speaker said*, and T_y being the nucleus.

Thus a DN Structure corresponds to a text of the form *Speaker₁ said P₁, Speaker₂ said P₂, ...*, where *Speaker₁*, *Speaker₂*, ... can be the same or different speakers. In the extreme, a DN structure can represent an internal monologue by a single speaker, or a conversation which has a different speaker for each span. Here, however, we deal mainly with DN Structures that have two alternating speakers.

Mapping from RST to DialogueNet structures Mapping RST to DialogueNet structures can be decomposed into two tasks. Firstly, we need to introduce the aforementioned *ATtribution* relations into the input RST structure. On its own, this would, however, not lead to very natural dialogues; rather, we would end up with a presentation of the input text by two or more speakers. To create a proper dialogue, we also need to introduce instances of the RST *QUESTION-ANSWER* relation into the input RST structure. Questions are characteristic of dialogue. They move the dialogue forward by at the same time introducing new topics and making requests for information. This raises the issue

of how to introduce QUESTION-ANSWER relations into RST structures. Take the following flat representation of an RST structure: (1) **CONDITION(P, Q)**, where bold face indicates the nucleus. A question-answer pair corresponding with this structure is: (2) QUESTION-ANSWER(What if P, **Q**). We use this example to illustrate two problems. Firstly, how do we arrive at the question-answer pair, and, secondly, given our commitment to information preserving mappings, what is the formal correlate of information preservation from (1) to (2)? (1) and (2) are supposedly carrying the same information, but they do not even have an RST relation in common.

To address this problem, we use a tool from mathematical logic, called λ -abstraction. One problem with (2) is that it obscures the fact that there is an underlying CONDITION relation. Instead, let us write

$$(3) \text{QUESTION-ANSWER}(\lambda x. \text{CONDITION}(P, x), \mathbf{Q}).$$

Thus, in (2) we replaced ‘What if P’ with $\lambda x. \text{CONDITION}(P, x)$. The latter is a formal representation of the former. The question is now analyzed as an abstraction over one of the arguments of the CONDITION-relation. Abstraction is the sister of application. If we apply a lambda expression $(\lambda x. M)$ to another expression N , the result is defined as follows: $(\lambda x. M)N \mapsto M[x := N]$. This now allows us to explicate in what sense (1) and (3) are equivalent. For that purpose, our formal interpretation of the QUESTION-ANSWER relation is application, and consequently: $\lambda x. \text{CONDITION}(P, x) \mathbf{Q}$ can be related to **CONDITION(P, Q)**. The use of abstraction and application to represent the equivalence between question-answer pairs and declarative sentences has been independently proposed by several researchers (see [3]).

Apart from the technical benefit of being able to express information equivalence on RST structure precisely, abstraction also provides us with a generic tool for generating question-answer pairs from declarative sentences and larger units. The general formula for question formation over a subexpression E of P is: $P \mapsto \lambda x. P' E$, where $P' = P[E := x]$. This allows us to generate various types of question-answer pairs via abstraction over different parts of the input, e.g.:

- (A) Over the first argument of a relation: If it rains, the tiles get wet. \mapsto Under what circumstances do the tiles get wet? If it rains.
- (B) Over the second argument: If it rains, the tiles get wet. \mapsto What if it rains? Then the tiles get wet.
- (C) Over a relation (higher-order): John is at home because I saw his car outside. \mapsto What is the relation between John being at home and his car being outside? The latter is evidence for the former.
- (D) Over a subexpression of a simple proposition: John is at home. \mapsto Where is John? At home.

Note that the mapping in (D) corresponds with that proposed in [12]. Our approach provides the formal underpinning for that work, and also presents

a significant generalization of it, showing its relation to many other declarative to question-answer pair mappings.

So far, we have implemented mappings for some of the most common relations – CONDITIONAL, CONCESSION, ELABORATION, SEQUENCE, and DISJUNCTION – and we are continually adding mappings for new relations. It is straightforward to add mappings for additional discourse relations to T2D. We have developed a generic format for specifying such mappings. Our methodology for adding mappings and evaluating them on naturally occurring text is described in the next section.

ROBUSTNESS of the current version of T2D is limited by the performance of the underlying DAS parser. An evaluation of DAS's performance is described in section 3.4. DAS failed for 39% of the inputs that it was presented with. Failure took different forms: a) DAS crashed or produced no analysis, b) ill-formed input, as a result OCR errors,⁶ led to an incorrect analysis by DAS, or c) the input was well-formed but DAS nevertheless produced an incorrect analysis. Currently, for the 39% of cases where DAS fails, our system produces no or an incorrect output. We are working on a number of strategies to address this problem: Firstly, we are exploring whether, when DAS crashes or produces no mapping, running it only on carefully selected subspans of the input might still yield useful results. Secondly, we intend to do some preprocessing of the input to check for OCR errors. Finally, for those cases where there was a well-formed input but an incorrect analysis, we are investigating post-processing on the DAS output to spot these (e.g., we observed that for the cases where DAS produces an incorrect analysis, the resulting tree often contains unnecessarily many nestings).

The mapper, which is the main topic of this paper, is successful for almost all of the inputs (see section 3.4). Although we are deriving the mappings from a specific corpus, we anticipate that they are portable to other text genres, since they are defined in terms of domain-independent RST and syntactic constraints. There might, however, be problems with specific genres. For instance, narratives will typically be annotated mainly in terms of TEMPORAL-AFTER relations, which makes for rather uninteresting dialogue.

VARIATION is addressed by allowing for multiple mappings for one and the same discourse relation. Currently, how to deal with CONTROL of variation is still an open issue. We are planning to investigate contextual factors that might determine the choice between different mappings.

3.2 Authoring of mapping rules

In this section we describe our methodology for authoring mapping rules by discussing a particular discourse relation, i.e., CONDITION. Evaluation is dealt with in Section 3.4. The development is empirically driven. We start out with a large collection of instances of the discourse relation in question. For this purpose, we use the PIL corpus,⁷ a corpus consisting of 465 Patient Information

⁶ The PIL corpus was created by scanning a collection of paper leaflets.

⁷ Available at: http://mcs.open.ac.uk/nlg/old_projects/pills/corpus/

Leaflets. We identified conditionals in the corpus by searching it with the regular expression $[I| i]f\backslash b$. This yielded a total of 4214 instances. We took a random sample of 100 sentences. Manual examination of the sample sentences led to their classification into two main categories: (1) nucleus of the conditional in (negative) imperative form, and (2) nucleus of the conditional in declarative form with modal auxiliary. For each case, we distilled separate mapping rules that are paraphrased below:

MAPPING RULES: Condition with Imperative Nucleus

CONDITION(P,Q) & imperative(P) \implies
 Layman: Under what circumstances should I P^* ?
 Expert: If Q .

CONDITION(P,Q) & neg-imperative(P) \implies
 Layman: Under what circumstances should I not P^* ?
 Expert: If Q .

where P^* is $P[I:=you,you:=I,my:=your,your:=my,mine:=yours,yours:=mine]$

MAPPING RULE: Condition with Nucleus in Declarative Form with Modal Auxiliary

CONDITION(P,Q) & declarative-modal-aux(P) \implies
 Layman: Under what circumstances *flip*(P^*)?
 Expert: If Q .

P^* is $P[I:=you,you:=I,my:=your,your:=my,mine:=yours,yours:=mine]$, and *flip*(X) is a function that performs the “interrogative flip” [19] inverting subject and auxiliary.

Here is an example for conditions where the nucleus is in (positive) imperative form. Given the input text ‘If you experience any other unusual or unexpected symptoms consult your doctor or pharmacist’, CONDITION(P,Q) is instantiated as CONDITION(*consult your doctor or pharmacist*, *you experience any other unusual of unexpected symptoms*). Syntactic analysis of the two clauses with the Machine Syntax parser⁸ tells us that P is in imperative form, i.e., P is the nucleus and Q is the satellite.

When the mapping rule provided above is applied, we obtain:

Layman: Under what circumstances should I consult my doctor or pharmacist?
 Expert: If you experience any other unusual or unexpected symptoms.

Depending on the application, dialogue contributions are assigned to more specific role pairs of type Expert–Layman, such as Instructor–Student, Boss–Assistant, and so on.

⁸ <http://www.connexor.com/>

An example of a condition with a declarative nucleus is ‘It should not produce any undesirable effects if you (or somebody) accidentally swallows the cream’. This is represented as `CONDITION(it should not produce any undesirable effects, you (or somebody) accidentally swallows the cream)`. The nucleus contains a modal auxiliary (“should”).

After applying the interrogative flip, the resulting dialogue is:

Layman: Under what circumstances should it not produce any undesirable effects?

Expert: If you (or somebody) accidentally swallows the cream.

In order to introduce VARIATION into the dialogues, we also prepared alternate mappings. For example, for conditions we use the following additional mapping, which is independent of the form of the nucleus.

MAPPING RULE: **Alternate Mapping Rule for Conditional**

`CONDITION(P,Q) & nucleus(P) \implies`

Layman: What if Q^* ?

Expert: Then P .

Here Q^* is $Q[I:=you, you:=I, my:=your, your:=my, mine:=yours,yours:=mine]$

When applied to ‘It should not produce any undesirable effects if you (or somebody) accidentally swallows the cream’, this mapping rule yields the following dialogue fragment:

Layman: What if I (or somebody) accidentally swallows the cream.

Expert: It should not produce any undesirable effects.

Note that this dialogue fragment appears to be more natural than the fragment produced by the other mapping (see above). One further strand of research we intend to pursue, is to develop a version of the system that creates and compares alternative mappings and selects the best one based on independent criteria (e.g., a measure of fluency).

3.3 Algorithm

The mapping algorithm performs prefix-parsing on the RST tree. The final dialogue is composed of sub-dialogues generated by recursively parsing the tree. The transitional words or expressions spoken by the Expert or the Layman are decided on the basis of the relation type of the current node. For instance, parsing an ELABORATION node will result in the concatenation of the dialog generated when parsing the left-hand child node, then the sentences ‘Expert: Should I tell you more? – Layman: Yes, please.’, followed by the Expert speaking the dialogue generated when parsing the right-hand child node. This algorithm, while simple, provides a good level of flexibility. It allows us to incrementally enrich the list of relation types for our system. Hence, we can evaluate the MAPPER on progressively richer samples every time a previous mappings have been validated.

3.4 Preliminary Evaluation

We conducted separate evaluations on both the DAS discourse parser and the MAPPER from the RST tree to DialogueNet. To evaluate DAS, we took a random sample of one hundred sentences from the PIL corpus. When compared to the analyses of a single human judge⁹, DAS achieved correct discourse parse results for 61%. The failed parse outcomes for the remaining 39% can be divided into four categories: (1) well-formed input, but incorrect analysis (40%), (2) ill-formed input (as a result of OCR errors; the corpus was created by scanning a large number of leaflets) and incorrect analysis (19%), (3) no mapping performed (19%), and (4) DAS crashes (22%).

In order to evaluate the MAPPER we took another random sample of one hundred condition sentences from the PIL corpus which we manually annotated in terms of RST by one of the authors. The dialogue was correctly mapped, according to a single judge, in 92% of the cases. In the 8% of remaining cases, the generated dialog was incorrect for one of the following reasons: (1) the structure of the sentence was not correctly analyzed due to incorrect output from the Machine Syntax parser (4%), or (2) the mapping rules were not precise enough (4%).

4 Walk-through of Example

In this section, we describe the operation of T2D on a multi-sentence text by looking in detail at a specific input text and the DialogueNet and multi-modal dialogue that T2D can produce. We take three sentences from the PIL corpus that are also discussed in [14]: (i) *To take a tablet, you should first remove it from the foil and then swallow it with water.* (ii) *Your doctor will tell you the dosage.* (iii) *Follow his advice and do not change it.*

The text is input into our ANALYZER component as plain text, and processed by the DAS Discourse Analyzing System. DAS outputs an XML file consisting of tagging structures for RST relations, which encodes the RST tree corresponding to the input text. The Identification Module for nucleus/satellite determines the relative importance of two clauses between which a (mononuclear) rhetorical relationship holds, by syntactic analysis. For instance, if the relation is CONDITION, the nucleus is identified by the occurrence of a verb in imperative form or the presence of a modal auxiliary (see Section 3.2). The output of the ANALYZER can be visualized by an RST tree, as shown in Fig. 1. Sentence (i) and sentences (ii) and (iii) are connected by the multinuclear SEQUENCE relation, which is often used when no more specific relationship can be identified. The satellite of a MEANS relation specifies '[...] a method, mechanism, instrument, channel or conduit for accomplishing some goal.' [4, p. 62]. ELABORATION is a common way to modify the nucleus by providing additional information.

Next, the MAPPER Module is called to transform the RST tree into a DialogueNet structure. An example dialogue reads as follows:

⁹ In this preliminary study, we used a single judge. We are planning further studies with two judges to assess interjudge agreement.

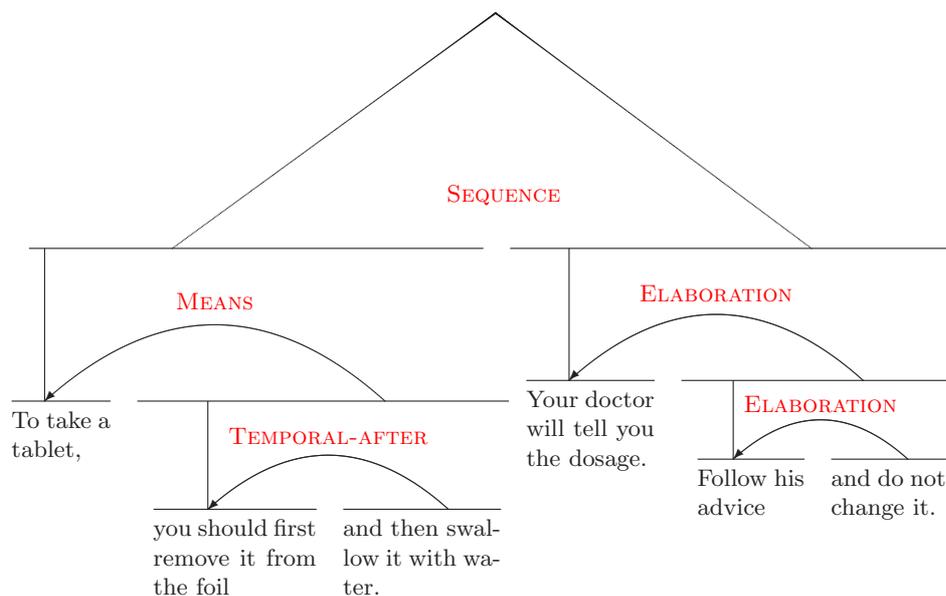


Fig. 1. RST tree of sample input sentences.

- (1) Layman: How should I take the tablet?
- (2) Expert: You should first remove it from the foil and then swallow it with water.
- (3) Expert: Your doctor will tell you the dosage.
- (4) Expert: Should I tell you more?
- (5) Layman: Do I have to follow his advice?
- (6) Expert: Yes.
- (7) Expert: And do not change the dosage.

The nucleus of the MEANS relation sentence (i) is mapped to a question, dialogue contribution (1), explicating the intention of this relation. The answer, contribution (2), can be organized as a TEMPORAL-AFTER relation. However, (2) is not turned into a question-answer pair, since this relationship (similar to SEQUENCE) does not justify the formation of a question. This can be contrasted to the situation in sentence (iii). Here the nucleus of the ELABORATION relationship is turned into a question with an *induced* answer (“Yes”), followed by the satellite information, contribution (7). Note that anaphora resolution was applied in (7) for disambiguation. Besides induced answers, we also implement induced questions, such as dialogue contribution (4). Both types are intended to smoothen the course of the dialogue. We are currently investigating a principled method to introduce them into the dialogue.

Finally, the purpose of the PRESENTER Module is to translate the DialogueNet structure into MPML3D [13], our Multimodal Presentation Markup Language for highly realistic 3D agents (see Fig. 2). The agents were created



Fig. 2. Multi-modal dialogue.

by a professional Japanese character designer for “digital idols”. They can perform around thirty gestures, express facial emotions, and speak with proper lip-synchronization. While MPML3D provides an easy-to-use, intuitive, and powerful scripting language for the definition of agent behavior, conversational gestures and gaze behavior have to be added manually. Since our T2D system is intended as a fully automated system, we are currently conducting extensive research in also automating this process.

5 Conclusions

We have developed a first working prototype of our Text2Dialogue system. In this paper, we presented both the theoretical grounding of the mapping that the system performs from Rhetorical Structure Theory structures to DialogueNet structures, and the system implementation. We introduced several requirements (ROBUSTNESS, EXTENSIBILITY, and VARIATION and CONTROL) and described how these are addressed. We also reported on the evaluation of the mapping rules that the system uses. In our future work, we aim to extend the system to mappings for further discourse relations, and to increase the naturalness of the dialogue by special devices, such as inserting induced questions. In this way, we want to advance the ease of generating high-quality multi-modal contents for non-professional and expert digital content creators alike.

Acknowledgements We would like to thank *Huong Le Thanh* for making the DAS system available to us, and *Abdul Ahad* and *Christian Pietsch* helping us

with the installation of DAS. We would also like to acknowledge the helpful comments and suggestions of the three anonymous IVA07 reviewers.

References

1. E. André. The generation of multimedia presentations. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*, pages 305–327. Marcel Dekker, Inc, 2000.
2. E. André, T. Rist, S. van Mulken, M. Klesen, and S. Baldes. The automated design of believable dialogue for animated presentation teams. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 220–255. The MIT Press, Cambridge, MA, 2000.
3. R. Bäuerle and T. Zimmermann. Fragesätze. In A. von Stechow and D. Wunderlich, editors, *Semantics. An International Handbook of Contemporary Research*, pages 333–348. Mouton de Gruyter, Berlin/New York, 1991.
4. L. Carlson and D. Marcu. Discourse tagging reference manual. Technical Report ISI-TR-545, ISI, September 2001.
5. R. Cox, J. McKendree, R. Tobin, J. Lee, and T. Mayes. Vicarious learning from dialogue and discourse: A controlled comparison. *Instructional Science*, 27:431–458, 1999.
6. S. Craig, B. Gholson, M. Ventura, A. Graesser, and the Tutoring Research Group. Overhearing dialogues and monologues in virtual tutoring sessions: Effects on questioning and vicarious learning. *International Journal of Artificial Intelligence in Education*, 11:242–253, 2000.
7. R. Davis. *Writing for Dialogue Scripts*. A & C Black Ltd, London, 1998.
8. D. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, USA, 1979.
9. H. T. Le and G. Abeyasinghe. A study to improve the efficiency of a discourse parsing system. In *Proceedings 4th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-03)*, Springer LNCS 2588, pages 101–114, 2003.
10. W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.
11. R. Mitkov, L. A. Ha, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering: Special Issue on using NLP for Educational Applications*, 12(2):177–194, 2006.
12. A. Nadamoto and K. Tanaka. Complementing your TV-viewing by web content automatically-transformed into TV-program-type content. In *Proceedings 13th Annual ACM International Conference on Multimedia*, pages 41–50. ACM Press, 2005.
13. M. Nischt, H. Prendinger, E. André, and M. Ishizuka. MPML3D: a reactive framework for the Multimodal Presentation Markup Language. In *Proceedings 6th International Conference on Intelligent Virtual Agents (IVA-06)*, Springer LNAI 4133, pages 218–229, 2006.
14. P. Piwek, R. Power, D. Scott, and K. van Deemter. Generating multimedia presentations. From plain text to screenplay. In O. Stock and M. Zancanaro, editors, *Multimodal Intelligent Information Presentation*, Text, Speech, and Language Technology, pages 203–225. Springer, 2005.

15. P. Piwek, R. Power, and S. Williams. Generating scripts for personalized medical dialogues for patients. Technical Report 2006/06, Department of Computing, Faculty of Mathematics and Computing, The Open University, UK, 2006.
16. P. Piwek and K. van Deemter. Towards automated generation of scripted dialogue: some time-honoured strategies. In *Proceedings 6th Workshop on the Semantics and Pragmatics of Dialogue (EIDLOG-02)*, pages 141–148, 2002.
17. D. Reitter. Rhetorical theory in LaTeX with the rst package. URL: <http://www.reitter-it-media.de/>.
18. K. Sumi and K. Tanaka. Transforming E-contents into a storybook world with animations and dialogues using semantic tags. In *Online Proceedings of WWW-05 Workshop on the Semantic Computing Initiative (SeC-05)*, 2005. URL: <http://www.instsec.org/2005ws/>.
19. C. L. Tenny and P. Speas. The interaction of clausal syntax, discourse roles, and information structure in questions. In *ESSLLI 2004 Workshop on Syntax, Semantics and Pragmatics of Questions*, Université Henri Poincaré, France, 2004.