The CODA System for Monologue-to-Dialogue Generation

Svetlana Stoyanchev
Centre for Research in Computing
The Open University
Walton Hall, Milton Keynes, UK
s.stoyanchev@open.ac.uk

Paul Piwek
Centre for Research in Computing
The Open University
Walton Hall, Milton Keynes, UK
p.piwek@open.ac.uk

Abstract

This paper describes an implemented monolingual Text-to-Text generation system. The system takes monologue and transforms it to two-participant dialogue. The system uses mappings between discourse relations in text and dialogue acts in dialogue. These mappings are extracted from a parallel monologue and dialogue corpus.

1 Introduction

This paper describes the CODA system,[1] a Text-to-Text generation system that converts text parsed with discourse relations (Mann and Thompson, 1988) into information-delivering dialogue between two characters. By information-delivering dialogue, we mean dialogue (akin to that used by Plato) that is used primarily to convey information and possibly also to make an argument; this in contrast with dramatic dialogue which focuses on character development and narrative.

Several empirical studies show that delivering information as dialogue, rather than monologue, can be particularly effective for education (Craig et al., 2000; Lee et al., 1998) and persuasion (Suzuki and Yamada, 2004). Information-delivering dialogue also lends itself well for presentation through computer-animated agents (Prendinger and Ishizuka, 2004).

With most information locked up in text (books, newspapers, leaflets, etc.), automatic generation of dialogue from text in monologue makes it possible to convert information into dialogue on demand.

In contrast to previous Text-to-Dialogue systems (Piwek et al., 2007), the CODA system is data-driven and modular. The system is composed of three modules: *Dialogue Modeller, Verbalizer,* and *Dialogue Merger*.

The *Dialogue modeller* determines appropriate dialogue act sequences that can be used for converting a segment of input text containing a single discourse relation into dialogue. The module is data-oriented in that the mappings it uses between discourse structure and dialogue act sequences have been derived from the CODA parallel monologue/dialogue corpus (Stoyanchev and Piwek, 2010).

The *Verbalizer* converts text segments together with a specification of the target dialogue act types into dialogue utterances.

The Dialogue modeller and verbaliser components overgenerate possible outputs for each discourse relation in monologue. The *Dialogue Merger* component selects one of the proposed outputs for each text segment of the input and merges them into a single coherent dialogue.

2 System Design

In this section we describe the three components of the system: dialogue modeller, verbalizer, and dialogue merger.

Before we look at each of the modules, we, however, first need to specify more precisely what the

[1] CODA stands for COherent Dialogue Automatically generated from text (see http://computing.open.ac.uk/coda/). The CODA project is funded by the UK's Engineering and Physical Sciences Research Council under Grant EP/G020981/1.

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 335–337,
Portland, Oregon, June 17-18, 2011.

Input	MANNER-MEANS [In September, Ashland settled the long-simmering dispute] [by agreeing to pay Iran $325 million.]
Dialogue Modeller	1. (ComplexQ; Explain) 2. (Explain; ComplexQ; Explain) 3. (Explain; YesNoQ; Explain)
Verbalizer DA Seq1	A: How did Ashland settle the long-simmering dispute in September? B: By agreeing to pay Iran $325 million.
Verbalizer DA Seq2	A: In September, Ashland settled the long-simmering dispute. B: How? A: By agreeing to pay Iran $325 million.
Verbalizer DA Seq3	A: In September, Ashland settled the long-simmering dispute. B: By agreeing to pay Iran $325 million? A: Correct.
Dialogue Merger	Select one of the DA sequences based on overall dialogue

Table 1: Example of the output from each component

input for our system is. The system expects text that has already been annotated with a discourse structure. There have been recent encouraging advances in the automatic parsing of discourse structure, e.g., see duVerle and Prendinger (2009), but the state-of-the-art is not yet at a point where it provides sufficiently reliable inputs for our purposes. To demonstrate the functionality of our system without relying on still imperfect discourse parsing, we use the RST-parsed Wall Street Journal corpus as input (Carlson et al., 2001).

Throughout the remainder of this section, we use the outputs for each of the modules in Table 1 as a running example.

2.1 Dialogue Modeller

The *Dialogue Modeller* component takes as input a snippet of monologue text annotated with discourse structure. For each input Discourse Relation structure (DR), the dialogue modeller outputs a set of dialogue act (DA) sequences appropriate for expressing the same information, but now in dialogue form.

The *Dialogue modeller* uses a configuration XML file to look up possible DA sequences for the input

DA sequence
YesNoQ; Explain
YesNoQ; Yes; Explain
Explain; ComplexQ; Explain
ComplexQ; Explain
Explain; YesNoQ; Resp-Answer-Yes
Explain; Contradict
Factoid-Info-Req;Factoid-Resp;Explain
Exlain; Resp-Agree;Explain

Table 2: Dialogue act sequences

discourse structure. In the current system configuration we extract these mappings from the CODA parallel corpus of professionally authored dialogues and parallel monologues. We use the eight most frequent DA sequences (see Table2) that occur on the dialogue side of discourse relations in the parallel dataset. Each discourse relation is mapped to one or more DA sequences with a score indicating frequency of this mapping in the CODA corpus.

The dialogue modeller can be customised with mappings from other sources such as a different corpus, manually authored mappings or a mapping arrived at through experimental methods.

The current version of the dialogue modeller supports input with only one level of discourse structure annotation. As a result, all input structures contain parts made of two segments and one discourse relation between these segments. In the future work, we plan to implement a dialogue modeller that accepts more complex (nested) discourse structures.

2.2 Verbalizer

The verbalizer is rule-based and has three types of rules: discourse relation (DR)-specific, generic, and canned. All of the rules take as input a monologue segment and a target dialogue act. DR-specific rules also use the discourse relation and segment nuclearity of the input segment.[2] The verbalization rules are ordered according to their priority with DR-specific rules having a higher priority.

Generic and DR-specific rules use the CMU question generation tool (Heilman and Smith, 2010) in combination with syntactic and lexical manipulation rules. Canned text rules are used to generate *AnswerYes*, *Agree* and *Clarify* dialogue acts by proba-

[2]Nucleus is the more salient segment in a relation.

bilistic selection from a set of utterances extracted from the CODA corpus. For example, the *Agree* dialogue act is verbalized as one of the statements: *I agree with you; I agree; I couldn't agree more; I completely agree; Absolutely; Very true; Right; True*. Probabilistic selection from a list allows us to generate non-repetitive dialogues. The system is extendible, such that new rules can be easily added to the implementation.

2.3 Dialogue Merger

The Dialogue Merger component takes as input verbalized dialogue act sequences. The tasks of the Dialogue Merger include: 1) selecting the best verbalized sequence and 2) assigning speaker roles (TEACHER or STUDENT) to dialogue turns.

We aim to create diverse dialogues, in particular, by avoiding repetitive use of the same dialogue act sequences. This is achieved as follows. Selection of DA sequence is incremental, considering one relation at a time. For each relation, the dialogue merger selects a dialogue act sequence that has been successfully verbalized by the *verbalizer* and which, so far, has been used the smallest number of times (out of all the sequences that have been used up to this point).

Although in the original authored dialogues, both TEACHER and STUDENT ask questions and give explanations, in our preliminary experiments observers made negative comments about mixing initiative between the STUDENT and the TEACHER in the generated dialogues. In the current version, the speaker roles are assigned based on the dialogue act. All questions and clarification requests are assigned to the STUDENT and other dialogue acts are assigned to the TEACHER.

As an additional post-processing step, to maintain perspective in the dialogue, we change pronouns in the dialogue turns. The turns assigned to the TEACHER character remain unchanged. The turns assigned to the STUDENT character change the perspective: non-possessive pronouns are inverted, e.g. *you* $\rightarrow$ *I*, *we* $\rightarrow$ *us*, *my* $\rightarrow$ *your*.

3 Conclusions and Further Work

In this paper, we described a Text-to-Dialogue generation system that converts text annotated with discourse relations into dialogue. The system is modular, data-driven, and takes advantage of state-of-the-art question generation tools. Our evaluation of the dialogue modeller and verbalizer components described in (Piwek and Stoyanchev, 2011) shows that both accuracy and fluency of generated dialogues are not worse than that of human-written dialogues.

We plan to release the CODA Text-to-Dialogue system as open source code later this year. The system can be used as a starting point for researchers interested in evaluating NLP tools for question generation, dialogue modelling and paraphrasing in a dialogue generation task.

References

L. Carlson, D. Marcu, and M. E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, SIGDIA.

S. Craig, B. Gholson, M. Ventura, A. Graesser, and the Tutoring Research Group. 2000. Overhearing dialogues and monologues in virtual tutoring sessions. *International Journal of Artificial Intelligence in Education*, 11:242–253.

D. duVerle and H. Prendinger. 2009. A novel discourse parser based on support vector machines. In *Procs of ACL-IJCNLP)*, pages 665–673, Singapore, August.

M. Heilman and N. A. Smith. 2010. Good question! statistical ranking for question generation. In *Proc. of NAACL/HLT*, Los Angeles.

J. Lee, F. Dinneen, and J. McKendree. 1998. Supporting student discussions: it isn't just talk. *Education and Information Technologies*, 3:217–229.

W. C. Mann and S. A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

P. Piwek and S. Stoyanchev. 2011. Data-oriented Monologue-to-Dialogue Generation. In *Procs of ACL*.

P. Piwek, H. Hernault, H. Prendinger, and M. Ishizuka. 2007. T2D: Generating Dialogues between Virtual Agents Automatically from Text. In *Procs of IVA07*, LNAI 4722, pages 161–174. Springer Verlag.

H. Prendinger and M. Ishizuka, editors. 2004. *Life-Like Characters: Tools, Affective Functions, and Applications*. Cognitive Technologies Series. Springer, Berlin.

S. Stoyanchev and P. Piwek. 2010. Constructing the CODA corpus. In *Procs of LREC*, Malta.

S. V. Suzuki and S. Yamada. 2004. Persuasion through overheard communication by life-like agents. In *Procs of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Beijing.