HAL
archives-ouvertes.fr

Prototyping virtual instructors from human-human corpora

Luciana Benotti, Alexandre Denis

▶ To cite this version:

Luciana Benotti, Alexandre Denis. Prototyping virtual instructors from human-human corpora. Association for Computational Linguistics: Human Language Technologies, Jun 2011, Portland, United States. 2011. <inria-00636300>

HAL Id: inria-00636300
https://hal.inria.fr/inria-00636300

Submitted on 27 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prototyping virtual instructors from human-human corpora

Luciana Benotti
PLN Group, FAMAF
National University of Córdoba
Córdoba, Argentina
luciana.benotti@gmail.com

Alexandre Denis
TALARIS team, LORIA/CNRS
Lorraine. Campus scientifique, BP 239
Vandoeuvre-lès-Nancy, France
alexandre.denis@loria.fr

Abstract

Virtual instructors can be used in several applications, ranging from trainers in simulated worlds to non player characters for virtual games. In this paper we present a novel algorithm for rapidly prototyping virtual instructors from human-human corpora without manual annotation. Automatically prototyping full-fledged dialogue systems from corpora is far from being a reality nowadays. Our algorithm is restricted in that only the virtual instructor can perform speech acts while the user responses are limited to physical actions in the virtual world. We evaluate a virtual instructor, generated using this algorithm, with human users. We compare our results both with human instructors and rule-based virtual instructors hand-coded for the same task.

1 Introduction

Virtual human characters constitute a promising contribution to many fields, including simulation, training and interactive games (Kenny et al., 2007; Jan et al., 2009). The ability to communicate using natural language is important for believable and effective virtual humans. Such ability has to be good enough to engage the trainee or the gamer in the activity. Nowadays, most conversational systems operate on a dialogue-act level and require extensive annotation efforts in order to be fit for their task (Rieser and Lemon, 2010). Semantic annotation and rule authoring have long been known as bottlenecks for developing conversational systems for new domains.

In this paper, we present novel a algorithm for generating virtual instructors from automatically annotated human-human corpora. Our algorithm, when given a task-based corpus situated in a virtual world, generates an instructor that robustly helps a user achieve a given task in the virtual world of the corpus. There are two main approaches toward automatically producing dialogue utterances. One is the selection approach, in which the task is to pick the appropriate output from a corpus of possible outputs. The other is the generation approach, in which the output is dynamically assembled using some composition procedure, e.g. grammar rules. The selection approach to generation has only been used in conversational systems that are not task-oriented such as negotiating agents (Gandhe and Traum, 2007), question answering characters (Kenny et al., 2007), and virtual patients (Leuski et al., 2006). Our algorithm can be seen as a novel way of doing robust generation by selection and interaction management for task-oriented systems.

In the next section we introduce the corpora used in this paper. Section 3 presents the two phases of our algorithm, namely automatic annotation and dialogue management through selection. In Section 4 we present a fragment of an interaction with a virtual instructor generated using the corpus and the algorithm introduced in the previous sections. We evaluate the virtual instructor in interactions with human subjects using objective as well as subjective metrics. We present the results of the evaluation in Section 5. We compare our results with both human and rule-based virtual instructors hand-coded for the same task. Finally, Section 6 concludes the paper proposing an improved virtual instructor designed as a result of our error analysis.

Proceedings of the ACL-HLT 2011 System Demonstrations, pages 62–67,
Portland, Oregon, USA, 21 June 2011. ©2011 Association for Computational Linguistics

2 The GIVE corpus

The Challenge on Generating Instructions in Virtual Environments (GIVE; Koller et al. (2010)) is a shared task in which Natural Language Generation systems must generate real-time instructions that guide a user in a virtual world. In this paper, we use the GIVE-2 Corpus (Gargett et al., 2010), a corpus of human instruction giving in virtual environments. We use the English part of the corpus which consists of 63 American English written discourses in which one subject guided another in a treasure hunting task in 3 different 3D worlds.

The task setup involved pairs of human partners, each of whom played one of two different roles. The "direction follower" (DF) moved about in the virtual world with the goal of completing a treasure hunting task, but had no knowledge of the map of the world or the specific behavior of objects within that world (such as, which buttons to press to open doors). The other partner acted as the "direction giver" (DG), who was given complete knowledge of the world and had to give instructions to the DF to guide him/her to accomplish the task.

The GIVE-2 corpus is a multimodal corpus which consists of all the instructions uttered by the DG, and all the object manipulations done by the DF with the corresponding timestamp. Furthermore, the DF's position and orientation is logged every 200 milliseconds, making it possible to extract information about his/her movements.

3 The unsupervised conversational model

Our algorithm consists of two phases: an annotation phase and a selection phase. The *annotation phase* is performed only once and consists of automatically associating the DG instruction to the DF reaction. The *selection phase* is performed every time the virtual instructor generates an instruction and consists of picking out from the annotated corpus the most appropriate instruction at a given point.

3.1 The automatic annotation

The basic idea of the annotation is straightforward: associate each *utterance* with its corresponding *reaction*. We assume that a reaction captures the semantics of its associated instruction. Defining reaction involves two subtle issues, namely *boundary* determination and *discretization*. We discuss these issues in turn and then give a formal definition of reaction.

We define the *boundaries* of a reaction as follows. A reaction r_k to an instruction u_k begins right after the instruction u_k is uttered and ends right before the next instruction u_{k+1} is uttered. In the following example, instruction 1 corresponds to the reaction $\langle 2, 3, 4\rangle$, instruction 5 corresponds to $\langle 6\rangle$, and instruction 7 to $\langle 8\rangle$.

DG(1): hit the red you see in the far room
DF(2): [enters the far room]
DF(3): [pushes the red button]
DF(4): [turns right]
DG(5): hit far side green
DF(6): [moves next to the wrong green]
DG(7): no
DF(8): [moves to the right green and pushes it]

As the example shows, our definition of boundaries is not always semantically correct. For instance, it can be argued that it includes too much because 4 is not strictly part of the semantics of 1. Furthermore, misinterpreted instructions (as 5) and corrections (e.g., 7) result in clearly inappropriate instruction-reaction associations. Since we want to avoid any manual annotation, we decided to use this naive definition of boundaries anyway. We discuss in Section 5 the impact that inappropriate associations have on the performance of a virtual instructor.

The second issue that we address here is *discretization* of the reaction. It is well known that there is not a unique way to discretize an action into subactions. For example, we could decompose action 2 into 'enter the room' or into 'get close to the door and pass the door'. Our algorithm is not dependent on a particular discretization. However, the same discretization mechanism used for annotation has to be used during selection, for the dialogue manager to work properly. For selection (i.e., in order to decide what to say next) any virtual instructor needs to have a *planner* and a *planning domain representation*, i.e., a specification of how the virtual world works and a way to represent the state of the virtual world. Therefore, we decided to use them in order to discretize the reaction.

Now we are ready to define *reaction* formally. Let S_k be the state of the virtual world when uttering in-

struction u_k, S_{k+1} be the state of the world when uttering the next utterance u_{k+1} and D be the planning domain representation. The *reaction* to u_k is defined as the sequence of actions returned by the planner with S_k as initial state, S_{k+1} as goal state and D as planning domain.

The annotation of the corpus then consists of automatically associating each utterance to its (discretized) reaction.

3.2 Selecting what to say next

In this section we describe how the selection phase is performed every time the virtual instructor generates an instruction.

The instruction selection algorithm consists in finding in the corpus the set of candidate utterances C for the current task plan P; P being the sequence of actions returned by the same planner and planning domain used for discretization. We define $C = \{U \in \text{Corpus} \mid U.\textit{Reaction}$ is a prefix of $P\}$. In other words, an utterance U belongs to C if the first actions of the current plan P exactly match the reaction associated to the utterance. All the utterances that pass this test are considered paraphrases and hence suitable in the current context.

While P does not change, the virtual instructor iterates through the set C, verbalizing a different utterance at fixed time intervals (e.g., every 3 seconds). In other words, the virtual instructor offers alternative paraphrases of the intended instruction. When P changes as a result of the actions of the DF, C is recalculated.

It is important to notice that the discretization used for annotation and selection directly impacts the behavior of the virtual instructor. It is crucial then to find an appropriate granularity of the discretization. If the granularity is too coarse, many instructions in the corpus will have an empty associated reaction. For instance, in the absence of the representation of the user orientation in the planning domain (as is the case for the virtual instructor we evaluate in Section 5), instructions like "turn left" and "turn right" will have empty reactions making them indistinguishable during selection. However, if the granularity is too fine the user may get into situations that do not occur in the corpus, causing the selection algorithm to return an empty set of candidate utterances. It is the responsibility of the virtual instructor developer to find a granularity sufficient to capture the diversity of the instructions he wants to distinguish during selection.

4 A virtual instructor for a virtual world

We implemented an English virtual instructor for one of the worlds used in the corpus collection we presented in Section 2. The English fragment of the corpus that we used has 21 interactions and a total of 1136 instructions. Games consisted on average of 54.2 instructions from the human DG, and took about 543 seconds on average for the human DF to complete the task.

On Figures 1 to 4 we show an excerpt of an interaction between the system and a real user that we collected during the evaluation. The figures show a 2D map from top view and the 3D in-game view. In Figure 1, the user, represented by a blue character, has just entered the upper left room. He has to push the button close to the chair. The first candidate utterance selected is "red closest to the chair in front of you". Notice that the referring expression uniquely identifies the target object using the spatial proximity of the target to the chair. This referring expression is generated without any reasoning on the target distractors, just by considering the current state of the task plan and the user position.

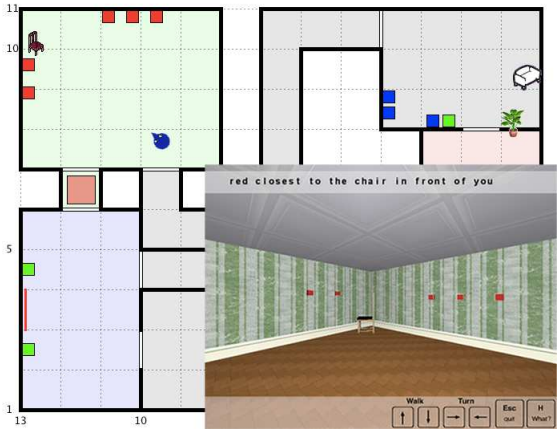

Figure 1: "red closest to the chair in front of you"

After receiving the instruction the user gets closer to the button as shown in Figure 2. As a result of the new user position, a new task plan exists, the set of candidate utterances is recalculated and the system selects a new utterance, namely "the closet one".

The generation of the ellipsis of the button or the

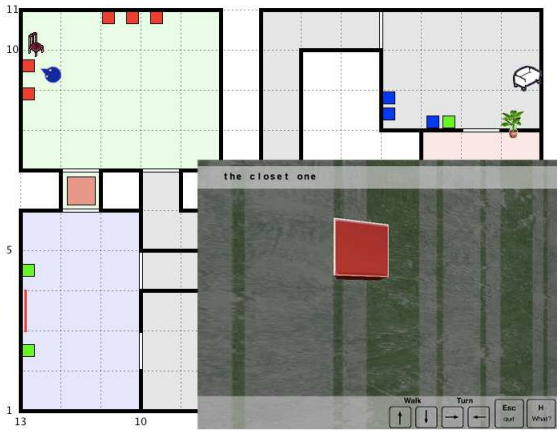

Figure 2: “the closet one”

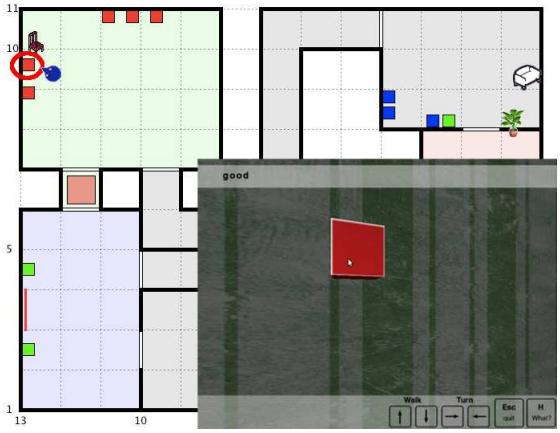

Figure 3: “good”

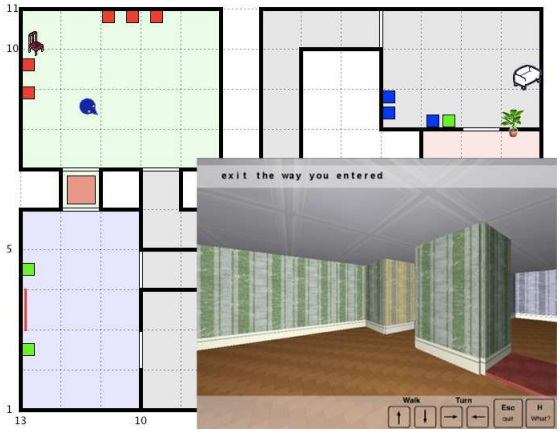

Figure 4: “exit the way you entered”

chair is a direct consequence of the utterances normally said in the corpus at this stage of the task plan (that is, when the user is about to manipulate this object). From the point of view of referring expression algorithms, the referring expression may not be optimal because it is over-specified (a pronoun would be preferred as in “click it”), Furthermore, the instruction contains a spelling error (‘closet’ instead of ‘closest’). In spite of this non optimality, the instruction led our user to execute the intended reaction, namely pushing the button.

Right after the user clicks on the button (Figure 3), the system selects an utterance corresponding to the new task plan. The player position stayed the same so the only change in the plan is that the button no longer needs to be pushed. In this task state, DGs usually give acknowledgements and this then what our selection algorithm selects: “good”.

After receiving the acknowledgement, the user turns around and walks forward, and the next action in the plan is to leave the room (Figure 4). The system selects the utterance “exit the way you entered” which refers to the previous interaction. Again, the system keeps no representation of the past actions of the user, but such utterances are the ones that are found at this stage of the task plan.

5 Evaluation and error analysis

In this section we present the results of the evaluation we carried out on the virtual instructor presented in Section 4 which was generated using the dialogue model algorithm introduced in Section 3.

We collected data from 13 subjects. The participants were mostly graduate students; 7 female and 6 male. They were not English native speakers but rated their English skills as near-native or very good.

The evaluation contains both objective measures which we discuss in Section 5.1 and subjective measures which we discuss in Section 5.2.

5.1 Objective metrics

The objective metrics we extracted from the logs of interaction are summarized in Table 1. The table compares our results with both human instructors and the three rule-based virtual instructors that were top rated in the GIVE-2 Challenge. Their results correspond to those published in (Koller et al., 2010) which were collected not in a laboratory but connecting the systems to users over the Internet. These hand-coded systems are called NA, NM and Saar. We refer to our system as OUR.

	Human	NA	Saar	NM	OUR
Task success	100%	47%	40%	30%	70%
Canceled	0%	24%	n/a	35%	7%
Lost	0%	29%	n/a	35%	23%
Time (sec)	543	344	467	435	692
Mouse actions	12	17	17	18	14
Utterances	53	224	244	244	194

Table 1: Results for the *objective* metrics

In the table we show the percentage of games that users completed successfully with the different instructors. Unsuccessful games can be either canceled or lost. To ensure comparability, time until task completion, number of instructions received by users, and mouse actions are only counted on successfully completed games.

In terms of task success, our system performs better than all hand-coded systems. We duly notice that, for the GIVE Challenge in particular (and probably for human evaluations in general) the success rates in the laboratory tend to be higher than the success rate online (this is also the case for completion times) (Koller et al., 2009).

In any case, our results are preliminary given the amount of subjects that we tested (13 versus around 290 for GIVE-2), but they are indeed encouraging. In particular, our system helped users to identify better the objects that they needed to manipulate in the virtual world, as shown by the low number of mouse actions required to complete the task (a high number indicates that the user must have manipulated wrong objects). This correlates with the subjective evaluation of referring expression quality (see next section).

We performed a detailed analysis of the instructions uttered by our system that were unsuccessful, that is, all the instructions that did not cause the intended reaction as annotated in the corpus. From the 2081 instructions uttered in the 13 interactions, 1304 (63%) of them were successful and 777 (37%) were unsuccessful.

Given the limitations of the annotation discussed in Section 3.1 (wrong annotation of correction utterances and no representation of user orientation) we classified the unsuccessful utterances using lexical cues into 1) correction ('no','don't','keep', etc.), 2) orientation instruction ('left', 'straight', 'behind', etc.) and 3) other. We found that 25% of the unsuccessful utterances are of type 1, 40% are type 2, 34% are type 3 (1% corresponds to the default utterance "go" that our system utters when the set of candidate utterances is empty). Frequently, these errors led to contradictions confusing the player and significantly affecting the completion time of the task as shown in Table 1. In Section 6 we propose an improved virtual instructor designed as a result of this error analysis.

5.2 Subjective metrics

The subjective measures were obtained from responses to the GIVE-2 questionnaire that was presented to users after each game. It asked users to rate different statements about the system using a continuous slider. The slider position was translated to a number between -100 and 100. As done in GIVE-2, for negative statements, we report the reversed scores, so that in Tables 2 and 3 greater numbers are always better. In this section we compare our results with the systems NA, Saar and NM as we did in Section 5.1, we cannot compare against human instructors because these subjective metrics were not collected in (Gargett et al., 2010).

The GIVE-2 Challenge questionnaire includes twenty-two subjective metrics. Metrics Q1 to Q13 and Q22 assess the effectiveness and reliability of instructions. For almost all of these metrics we got similar or slightly lower results than those obtained by the three hand-coded systems, except for three metrics which we show in Table 2. We suspect that the low results obtained for Q5 and Q22 relate to the unsuccessful utterances identified and discussed in Section 5.1. The high unexpected result in Q6 is probably correlated with the low number of mouse actions mentioned in Section 5.1.

NA	Saar	NM	OUR
Q5: I was confused about which direction to go in			
29	5	9	-12
Q6: I had no difficulty with identifying the objects the system described for me			
18	20	13	40
Q22: I felt I could trust the system's instructions			
37	21	23	0

Table 2: Results for the *subjective* measures assessing the efficiency and effectiveness of the instructions

Metrics Q14 to Q20 are intended to assess the nat-

uralness of the instructions, as well as the immersion and engagement of the interaction. As Table 3 shows, in spite of the unsuccessful utterances, our system is rated as more natural and more engaging (in general) than the best systems that competed in the GIVE-2 Challenge.

NA	Saar	NM	OUR
Q14: The system's instructions sounded robotic			
-4	5	-1	28
Q15: The system's instructions were repetitive			
-31	-26	-28	-8
Q16: I really wanted to find that trophy			
-11	-7	-8	7
Q17: I lost track of time while solving the task			
-16	-11	-18	16
Q18: I enjoyed solving the task			
-8	-5	-4	4
Q19: Interacting with the system was really annoying			
8	-2	-2	4
Q20: I would recommend this game to a friend			
-30	-25	-24	-28

Table 3: Results for the *subjective* measures assessing the naturalness and engagement of the instructions

6 Conclusions and future work

In this paper we presented a novel algorithm for rapidly prototyping virtual instructors from human-human corpora without manual annotation. Using our algorithm and the GIVE corpus we have generated a virtual instructor[1] for a game-like virtual environment. We obtained encouraging results in the evaluation with human users that we did on the virtual instructor. Our system outperforms rule-based virtual instructors hand-coded for the same task both in terms of objective and subjective metrics. It is important to mention that the GIVE-2 hand-coded systems do not need a corpus but are tightly linked to the GIVE task. Our algorithm requires human-human corpora collected on the target task and environment, but it is independent of the particular instruction giving task. For instance, it could be used for implementing game tutorials, real world navigation systems or task-based language teaching.

In the near future we plan to build a new version of the system that improves based on the error analysis that we did. For instance, we plan to change our discretization mechanism in order to take orientation into account. This is supported by our algorithm although we may need to enlarge the corpus we used so as not to increase the number of situations in which the system does not find anything to say. Finally, if we could identify corrections automatically, as suggested in (Raux and Nakano, 2010), we could get another increase in performance, because we would be able to treat them as corrections and not as instructions as we do now.

In sum, this paper presents a novel way of automatically prototyping task-oriented virtual agents from corpora who are able to effectively and naturally help a user complete a task in a virtual world.

[1]Demo at `cs.famaf.unc.edu.ar/~luciana/give-OUR`

References

Sudeep Gandhe and David Traum. 2007. Creating spoken dialogue characters from corpora without annotations. In *Proceedings of Interspeech*, Belgium.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 corpus of giving instructions in virtual environments. In *Proc. of the LREC*, Malta.

Dusan Jan, Antonio Roque, Anton Leuski, Jacki Morie, and David Traum. 2009. A virtual tour guide for virtual worlds. In *Proc. of IVA*, pages 372–378, The Netherlands. Springer-Verlag.

Patrick Kenny, Thomas D. Parsons, Jonathan Gratch, Anton Leuski, and Albert A. Rizzo. 2007. Virtual patients for clinical therapist skills training. In *Proc. of IVA*, pages 197–210, France. Springer-Verlag.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Sara Dalzel-Job, Johanna Moore, and Jon Oberlander. 2009. Validating the web-based evaluation of nlg systems. In *Proc. of ACL-IJCNLP*, Singapore.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. Report on the second challenge on generating instructions in virtual environments (GIVE-2). In *Proc. of INLG*, Dublin.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proc. of SIGDIAL*, pages 18–27, Australia. ACL.

Antoine Raux and Mikio Nakano. 2010. The dynamics of action corrections in situated interaction. In *Proc. of SIGDIAL*, pages 165–174, Japan. ACL.

Verena Rieser and Oliver Lemon. 2010. Learning human multimodal dialogue strategies. *Natural Language Engineering*, 16:3–23.