Building and Exploiting a Corpus of Dialog Interactions between French Speaking Virtual and Human Agents

Lina Maria Rojas Barahona, Alejandra Lorenzo, Claire Gardent

Building and Exploiting a Corpus of Dialog Interactions between French Speaking Virtual and Human Agents

Lina M. Rojas-Barahona[+], Alejandra Lorenzo [!]and Claire Gardent[*]

[+]INRIA Nancy, [+]![!]Lorraine University/LORIA Nancy, [*] CNRS/LORIA Nancy
{lina.rojas,alejandra.lorenzo,claire.gardent}@loria.fr

Abstract

We describe the acquisition of a dialog corpus for French based on multi-task human-machine interactions in a serious game setting. We present a tool for data collection that is configurable for multiple games; describe the data collected using this tool and the annotation schema used to annotate it; and report on the results obtained when training a classifier on the annotated data to associate each player turn with a dialog move usable by a rule based dialog manager. The collected data consists of approximately 1250 dialogs, 10454 utterances and 168509 words and will be made freely available to academic and nonprofit research.
Keywords: Corpus, Dialogue, Annotation.

1. Introduction

In recent years, there has been increasing interest in developing corpus-based dialog systems for virtual agents in simulation, games and serious games (Orkin and Roy, 2010; Gandhe et al., 2011; Leuski and Traum, 2010). The dialogs conducted by such systems differ from those found in existing available dialog corpora such as Trains (Heeman and Allen, 1994), Maptask (Carletta et al., 1996) or Verbmobil in several ways:

- they involve multiple participants, sometimes communicating in a variety of modalities (natural language but also facial expression, gestures, etc.)
- the dialog is driven by the logic of the game and consists, not of a single short dialog, but of a series of subdialogs situated in different contexts.
- dialogs are situated and each subdialog may involve a distinct set of participants in a distinct part of the world

While there has been much work on building corpora to train and evaluate English speaking virtual agents (Orkin and Roy, 2010; Gandhe et al., 2011; Leuski and Traum, 2010), there has been much less emphasis on other languages. There is in particular no such corpus for French. In this paper, we present the data collection and the annotation process realised to develop French speaking virtual characters for the "Mission Plastechnology" (MP) serious game, a game where several virtual characters interact with the user to guide her through a plastechnology plant and help her both play the game and learn about plastechnology.

The paper is structured as follows. We first describe the Human-Machine dialogs conducted within the MP serious game (Section 2.). We then go on to present a tool for data collection that is configurable for multiple games (Section 3.). In Section 4., we report on the data collection conducted with this tool both for the MP game and for different games designed by groups of students from a graphic design school thereby illustrating the portability of the data collection tool. Section 5. describes the annotation schema used to label the dialog turns with dialog acts. Section 6. presents the results obtained when training a classifier on this data and testing its use for predicting the dialog act performed by a user utterance. Section 7. concludes with pointers for further research.

2. Dialogs in the MP Serious Game

The MP game is designed to promote careers in the plastic industry. It is a multi-player quest where the players (3 teenagers) seek to build a video game joystick in order to free their uncle trapped in the game. To build this joystick, the players must explore the plastic factory and achieve 17 mandatory goals (find the plans, get the appropriate mould, retrieve some plastic from the storing shed, etc). In addition, they can increase their score by achieving optional goals which, when reached, provide them with extra information about the plastic industry (and therefore increases their knowledge of this industry). In total, the players can achieve up to 28 goals by conducting 12 separate subdialogs in various parts of the virtual world. That is, MP dialogs are long dialogs involving multiple players in various settings.

Table 1 summarises the characteristics of the subdialogs conducted within the MP game highlighting three distinguishing features of game dialogs. First, the dialog participants vary whereby both the game agent and the player can change. Thus in the MP game, the player alternatively plays any of the three children involved in the quest while the game agent is successively, Lucas, M. Jasper, Samir, Melissa, an operator, Serge and Sophia. Second, game dialogs are task-driven whereby each subdialog is related to a step in the game and each dialog turn aims to achieve a game goal and improve the player score. Third, the context in which each subdialog takes place varies as the player moves around the world.

3. Wizard-of-OZ tool for Data Collection

To collect Human-Game dialog data, we developed a Wizard-of-OZ (WOZ) interface using the MITRE Dialog Toolkit Midiki (Burke et al., 2003). Midiki, is a portable

Id	VC	Player	Mandatory Goals	Location
1	Lucas	Ben	Find the address of the enterprise.	Uncle's place.
2	M.Jasper	Lucas	The manufacturing first step	Enterprise reception
3	Samir	Julie	Find the plans of the joystick	Designing Office
4	Samir	Julie	Find out what to do next	Designing Office
5	Melissa	Lucas	Manufacturing process ...	Plant
6	Melissa	Lucas	Find the right machine	Plant
7	Melissa	Lucas	Find out what to do next	Plant
8	Operator	Julie	Knowing about the material space ...	Material Space
9	Serge	Ben	Perform quality tests	Laboratory Tests
10	Serge	Ben	Find out what to do next	Laboratory Tests
11	Sophia	Julie	Find the electronic components.	Finishing
12	Sophia	Lucas	Finishing process	Finishing

Table 1: Description of the subdialogs in the MP Game.

toolkit for building dialogue managers in Java. It implements the information-state model of dialogue (Traum and Larsson, 2003) where in essence, the information state models the progression of dialog while update rules formalise the way that information state is changed as the dialog progresses.

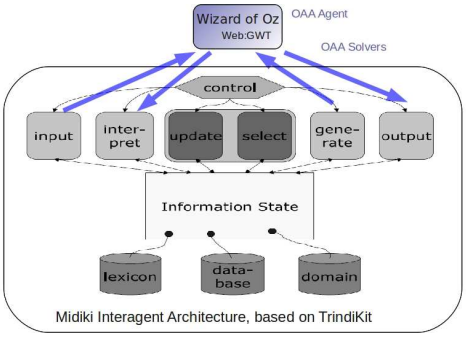

Figure 1: Interactions between the wizard and the Midiki rule-based dialog manager: in the *the semi-automatic wizard*, the player's input is interpreted by the wizard, the result of the interpretation is sent to the dialog system, the information state is updated and a response is selected. This response is then passed on to the wizard either to use it or to issue a different response.

We first extended Midiki to support a multi-agent architecture and the configuration from a relational database. We then used this extended Midiki (i) to develop a rule based dialog system for the MP game and (ii) to implement two Wizard-of-OZ interfaces for data collection: the free- and the semi-automatic WOZ interface.

The *free WOZ interface* aims to simulate mixed-initiative dialogs by allowing the wizard to chat with the player as she moves around the game while simultaneously storing all interactions in a database. A virtual dialog manager ensures that the wizard respects the game logic, starting the appropriate subdialogs at the appropriate place in the virtual world. In this setup, the interactions between the wizard and the player simulate a direct Human-Human dialog in the context of the MP game.

In contrast, *the semi-automatic wizard* favours system-driven dialogs by connecting the Wizard not only with the player and the game but also with the rule-based dialog manager (Figure 1). This dialog manager supports the Wizard by automatically interpreting the player's input and selecting a possible response. As the Wizard interacts with a player, she can then either accept the response suggested by the rule-based dialog manager (if this response is appropriate) or enter a different response (whenever the response suggested is incorrect or inappropriate).

Figure 2 shows the architecture of the WOZ interface: the dialog manager (either Midiki or a virtual DM), the MP game, the Wizard of Oz interface and an automatic speech recognition module (ASR) [1] communicate together within the Open Agent Architecture (OAA) (Cheyer and Martin, 2001). The WOZ interface is implemented as a web service and all interactions are logged into a relational database.

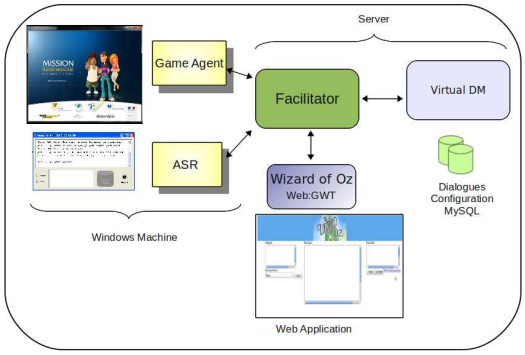

Figure 2: General Architecture for the Wizard of OZ experiments: modules are implemented as agents within the Open Agent Architecture.

To support data collection for different game scenarios, we also developed a Dialogue Configuration Tool (Figure 3). This tools permits defining for each new game the information that is relevant for the dialog system namely, which characters are present in the game; which goals are being pursued at each step in the game; and which subdialogs are being conducted in which order during the game, between which characters and to achieve which goal.

[1] Although the Wizard Framework supported both speech and written input, we did not record speech in our first experiments. All data is therefore written data.

	Subjects	Dialogs	Uttces	Tokens	Player U.	Player Tokens	Player Token Types
Semi-Aut.	40	591	4874	77854	1321	12901	1427
Free	50	658	5580	90655	2288	18712	1542
Total	90	1249	10454	168509	3609	31613	2969

Figure 4: Data collected

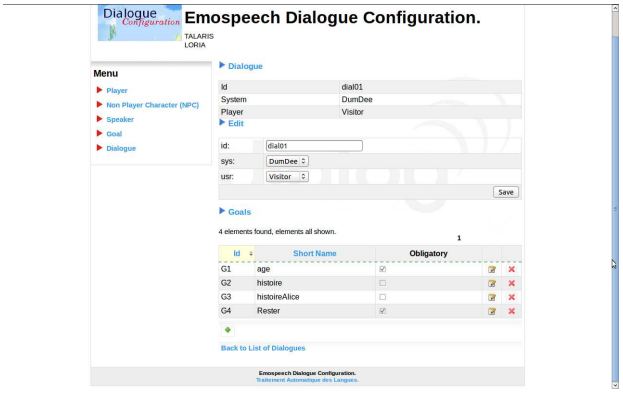

Figure 3: A screenshot of the Dialog Configuration Tool showing a configuration for a serious game leading a visitor through an exhibition on Alice in Wonderland. The first subdialog scenario is specified as involving Dumdee as the system, the player as the exhibition visitor and four goals namely, declaring one's age (age), choosing a story to listen to (histoire), deciding whether or not to listen to Alice's story and deciding whether to stay or not (rester).

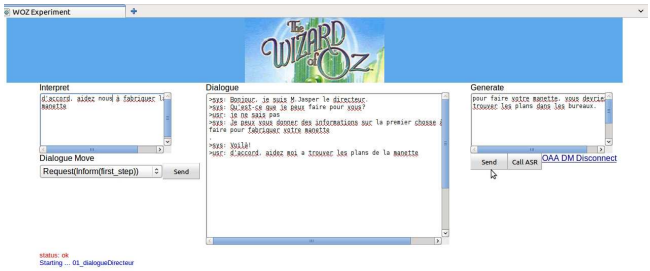

Figure 5: Wizard of OZ Graphical User Interface.The GUI has three parts, the interpretation, the dialog and the generation. The wizard receives the input sentence at the left-side, she can see the whole dialog in the center and she edits the generated utterance at the right side. She can introduce also the dialog move associated to the input and output sentences.

4. Data Collection

Data collection for the MP game proceeded in two steps. First, a native French speaker played the wizard using the semi-automatic WOZ with 40 subjects. Next, three groups of students from the Language and Communication Erasmus Mundus Master in Nancy collected dialogs using the free WOZ. The results are shown in Table 4.

Dialog length varies between 78 and 142 turns with an average length of 106 turns per dialog. Expert players completed the game in around 50 minutes in average while novice players took between 1 and 1.5 hour.

After configuration of the WOZ tool using the Dialog Configuration Tool mentioned in the preceding section, the free-WOZ was also used by Master students from Rennes University for collecting dialogs in a game simulating the visit of an exhibition on Alice in Wonderland. In this way, they collected 25 dialogs using Lewis Carroll's surrealist language (see Figure 7).

5. Annotation Scheme

To support supervised learning, we annotated the collected data with dialog moves.

5.1. Annotation Procedure

The Wizard-of-Oz tool for data collection presented in section 3. allows the wizard to annotate both, the player and the game agent turns (Figure 5). Since not all the data was pre-annotated this way and because the wizard can make errors, all annotations where manually reviewed in a second step. We evaluated the inter-annotator agreement with two students, who separately annotated 5% of the corpus and we obtained an inter-annotator agreement of 0.80, meaning that the annotation scheme has good reliability.

5.2. Annotation Schema

Many annotation schemes have been proposed for dialog. Precisely which schema to choose largely depends on how the annotations are to be used. As pointed out in (Traum and Larsson, 2003):

> dialog moves are meant to serve as an abstraction between the large number of different possible messages that can be issued and the types of update to be made on the basis of performed utterances [..] Dialogue moves are often conceived of as speech act in the sense of (Searle, 1969) but this is not a necessity, dialogue moves could be any mediating input e.g., logical forms or event word-lattices augmented with likelihoods.

In other words, the choice of the annotation schema is not necessarily dictated by speech act theory alone but might also consider more practical issues namely, how well it will support interpretation and/or dialog management (i.e., deciding how to respond to a player's utterance).

To enhance learning, the annotation schema designed for the MP game combines core communicative acts (Bunt et al., 2010) with domain specific information. The domain specific information specifies the goals being pursued/discussed/achieved etc. while the communicative act can be viewed as specifying how the current information state is updated by the speaker's utterance.

Dialog Acts	Dialog Moves	Gloss	Speaker
Welcome greeting	greet	Welcome greeting	P,S
Farewell greeting	quit	Farewell greeting	P, S
Address Request	ask(Goal)	Request to pursue Goal	S
Address Request	help	Request for help	P
Confirm	yes	Confirms previous query	P
Disconfirm	no	Disconfirms previous query	P
Provide Information	inform(Goal)	Provides information about how to achieve Goal	S
Provide Information	Goal	Provides information about the goal Goal	P
Positive Feedback	ack	Acknowledges understanding of preceding turn	S
Propositional Question	ask(do(more(X)))	Asks whether other topics should be discussed	S
Set Question	ask(topic(X))	Asks which other topics should be discussed	S
Out of Context	other	Out of context turn	P,S
Misunderstanding	reqRep	Request for rephrasing	S

Table 2: The MP Annotation Scheme. P and S stands for Player and System respectively.

The communicative acts used in the MP annotation schema are the following[2]:

- Welcome (greet) and farewell (quit) greetings
- Accept Request (yes), Decline Request (no), Address Request (ask)
- Positive Feedback (ack): acknowledges understanding
- Set Question (ask(topic(x)), Propositional Question (ask(P)): Set questions are question where the speaker wants to identify a set of elements having a given property; Propositional questions are questions where the speaker wants to know whether a statement is true of false
- Provide Information (inform(do(x)), Confirm (yes), Disconfirm (no): Confirm and disconfirm are functionally dependent on a previous Propositional Question dialog act.
- Out of domain: denotes dialog acts that are unrelated to the game.

As mentioned above, these communicative acts are specialised using domain knowledge namely knowledge about the game goals. For instance, a request can be specialised as a request for help on how to achieve a particular goal. Similarly, a "Provide Information" act can be specialised with a label indicating which game goal the dialog act provides information about. The following dialog illustrates this by showing the annotation of one of the subdialog in the MP game.

dialog: 01_dialogDirecteur-Tue Jun 14 11:04:23 2011
>**M.Jasper:**Bonjour, je suis M.Jasper le directeur.
|| ***greet***
Hello, I am the director, Mr. Jasper.
>**M.Jasper:**Qu'est-ce que je peux faire pour vous?
|| ***ask(task(X))***
What can I do for you?
>**Lucas:**je dois sauver mon oncle
|| ***first_step***
I must rescue my uncle
>**M.Jasper:** Pour faire votre manette, il vous faut des plans. Allez voir dans le bureau d'études, ils devraient y être.
|| ***inform(do(first_step))***
To build the joystick you will need the plans. You will find them in the Designing Office.
>**M.Jasper:**J'aurais aimé continuer à répondre à vos questions mais je dois reprendre mon travail! Bonne Chance!
|| ***quit***
I have to go back to work! Good Luck!

The first turn is labeled with a general communicative act (Welcome greeting). The second is a set question specialised to indicate that the question pertains to the player's goal (***ask(task(X))*** is used to label system turns asking the player which goal she is trying to achieve). Next the ***first_step*** label indicates that the player's current goal is ***first_step*** i.e., the first goal that needs to be achieved in the game. ***inform(do(first_step))*** is a "Provide information" dialog act which indicates that the system provides information about the ***first_step*** goal. Finally, ***quit*** labels a farewell greeting.

Table 2 gives the full list of Dialog Moves used for annotation together with the corresponding dialog acts and a gloss of their meaning. As can be seen the labels used are very specific to the game. The motivations for this are twofold. First, this permits a direct integration with the game in that the goals that need to be achieved in the game are the same goals that are used to specify the dialog moves. In particular, each time an ***inform(do(Goal))*** turn is identified (indicating that the system has provided the player with information about the game goal ***Goal***), the game can be notified that the player knows about the goal ***Goal*** and is currently pursuing it[3].

Another important motivation for having domain specific annotation labels is that this permits bypassing much of the pragmatic reasoning necessary to associate a dialog turn with a communicative function. For instance, in the dialog above, the turn *je dois sauver mon oncle (I must rescue my uncle)* does not explicitly state that the player (i) is seeking to achieve the game goal "rescueing one's uncle" and (ii)

[2]We use (Bunt et al., 2010)'s terminology and definitions and indicate in brackets the labels actually used to annotate the data

[3]The game is sequential and the player pursues a set of goals at a time.

is asking the game agent for the first step towards achieving that goal. As shown by the wizard answer however (*To build the joystick you will need the plans.*), a human being can infer this information and appropriately respond by instructing the player with the next step required to achieve the final goal.
As shall be shown in the next section, even though the annotation schema is relatively specific (since many of the labels can be instantiated with any of the goals in the game), thereby reducing the amount of data available for each move, learning still performs reasonably well.

6. Interpreting Dialog Moves

We used the annotated data to learn to interpret dialog moves. The dialog moves and the associated turns are then passed on to the Midiki rule based dialog manager which selects an appropriate response based on these and on the current information state. In what follows, we describe the experimental setup and the results obtained by the acquired classifiers.

6.1. Experimental setup

We experimented with both an SVM (Support Vector Machine) and an LR (Logistic Regression) [4] classifier using different sets of features on different data sets with and without TF*IDF (term frequency*Inverse Document Frequency) filtering.

	Whole Dialog		Subdialogs	
	w/o Tf*Idf	w/ Tf*Idf	w/o Tf*Idf	w/ Tf*Idf
LR	79.74	**90.26**	86.41	88.22
SVM	78.79	88.55	76.45	83.99
SVM (P)			78	83.55

Table 3: Global Results for the Logistic Regression (LR), the SVM (SVM) and the SVM Classifier with Penalisation (SVM(P))

Data Set, Categories and Evaluation Mode. Since the game dialogs consist of subdialogs, two options naturally arise: we could either train a single classifier on the whole dataset or train a classifier for each subdialog. We experimented with both options and learned either one or 12 classifiers. In both cases the categories to be learned are restricted to the speaker's dialog moves (greet,quit,inform(Goal), ack, ask(do(more(X))), ask(topic(X)), other in Table 2). Taking into account the game goals, the total number of categories to be learned is 27. As shown in Tables 4 and 5, when learning on subdialogs, the number of categories to be learned is smaller but so is the size of the training set. In all experiments, we use 30 fold cross validation on the training data for evaluation.
Features. As a preprocessing, utterances were unaccented and converted to lower-case, and stop words were removed. For all experiments, we used as features the content words remaining after this preprocessing. In addition, we experimented with various notions of contexts using as features the 0 to 4 previous dialog moves. When training the classifier on the whole data (rather than for each subdialog), subdialog Identifiers were also used.
*Tf*idf Filtering*. We experimented using tf*idf filtering to limit the impact of frequent uninformative words not filtered out in the preprocessing step.
Penalisation. As shown in Table 7, the data is highly skewed. In particular, dialog moves that relate to optional goals (e.g. the job of the virtual agent, the security policies, etc) are often not followed up by the player resulting in data sparseness for the corresponding dialog moves. For the SVM classifier[5], we experimented penalizing those categories with more training instances.

[4]We used MALLET (McCallum, 2002) for the LR classifier with L1 Regularisation.

Id	Train	Feat.	Cat.	LR	SVM(P.)	SVM
1	182	193	6	87.27	72.93	72.38
2	223	285	7	83.58	78.47	76.23
3	529	602	12	83.65	78.07	75.61
4	341	452	10	82.35	73.90	71.26
5	301	372	8	85.56	80.13	78.45
6	153	197	6	93.48	79.74	79.08
7	158	197	6	95.74	79.75	79.75
8	538	525	10	82.61	77.82	76.88
9	294	315	9	84.09	82.65	79.93
10	254	420	8	78.95	68.89	67.32
11	355	445	9	87.85	83.66	80.56
12	245	304	7	91.78	80.00	80.00
Average				86.41	78.00	76.45

Table 4: Results of the LR, SVM and SVM with penalisation trained for the interpretation with unigrams and a context of four previous system moves

6.2. Results

Table 3 shows the results for the 6 main configurations: training on the whole dialog or on subdialogs, with and without tf*idf filtering and using LR, SVM or SVM with penalisation. The best results are obtained using the LR classifier on the whole dataset with tf*idf filtering. Penalising improved slightly the accuracy of the SVM when classifying without tf*idf filtering or when having a reduced context (0 or 2 previous moves in Table 6).
*Impact of the tf*idf filtering*. Using each dialog move category as a document, we discarded words with tf*idf values smaller than 0.15. Turns containing only words with a low tf*idf value were eliminated from the training set.
As shown in Table 5, globally, the tf*idf filtering has a positive impact leading to an increase in accuracy ranging from 2.81 to 11.52 points. For the SVM classifier, the tf*idf filtering consistently lead to better results. However, for the LR classifier, Tables 5 shows that the filtering adversely impacts performance on subdialogs 6 and 7. These are short subdialogs where one unique goal is being discussed. We conjecture that for these cases, the tf*idf filtering removed words which helped the classifier distinguish between turns

[5]We use LIBSVM (Chang and Lin, 2001) for the implementation of the SVM classifier. We implemented a radial basis kernel with $\gamma = 0.07$.

Id	Train	Feat.	Cat.	LR	SVM(P.)	SVM
1	170	133	6	88.24	80.0	78.82
2	222	282	7	88.06	81.53	82.88
3	524	498	12	85.35	79.96	84.73
4	338	377	10	85.15	78.70	78.70
5	294	214	8	90.91	85.71	85.71
6	151	111	6	86.67	81.46	81.46
7	148	117	6	90.91	87.84	87.84
8	532	447	10	88.12	84.77	85.15
9	294	239	9	89.77	86.73	86.73
10	252	336	8	81.58	79.76	79.76
11	355	331	9	92.52	88.73	88.73
12	230	186	7	91.3	87.39	87.39
Average				88.22	83.55	83.99

Table 5: Results of the LR, SVM and SVM with penalisation trained on unigrams filtered by tf*idf and a context of four previous system moves.

about the unique goal from other turns.
Finally, it is worth noting that the SVM with penalisation yields worse results with the tf*idf filtering than without, thus suggesting overfitting. We are currently working on exploiting synonyms to improve generalisation.
Impact of contextual features. Having a notion of context is crucial for correctly interpreting dialog moves. As mentioned above, we use the dialog moves of the previous turns to model context. However the further back we look into the previous turns, the more features there will be to train on. In other words, depending on the number of previous turns considered, the data to learn from will be more or less sparse. We experimented with 3 setups: a null context, the dialog moves of the two previous turns and the dialog moves of the four previous moves. Table 6 shows the results.

	Whole Dialog			Subdialogs		
	0	2	4	0	2	4
LR	88.43	90.26	90.26	84.43	87.59	88.22
SVM	84.36	86.76	88.55	78.04	82.06	83.99
SVM(P)				79.32	83.12	83.55

Table 6: The impact of context on accuracy. 0,2 and 4 indicates that the context is captured by having as features the dialog moves of 0, 2 and 4 previous turns respectively

Impact of dialog moves. Figure 6 shows the variations in accuracy per dialog move. This ranges from 48% to 99%. with most of the moves having an accuracy above 80%. Unsurprisingly, the moves with lowest accuracy are also the moves with fewest training data (cf. Table 7). The data is split randomly for the 30-fold evaluation with the risk of having insufficient data for optional goals. As an example, we can see in Figure 6 that the LR classifier reports the lowest f1-measure for the optional goal "security_policies". Even if it is not the label with less instances, the bad results in this category are due to the data partitioning (only few instances belonging to this category remain in the 70% of

Mv	Train	Mv	Train	Mv	Train
greet	159	yes	353	no	298
ack	413	help	161	other	77
quit	234	fstep	89	fd plans	114
job	92	studies	20	staff	15
security	24	nstep	236	staff ent	16
stud ent	14	manuf	102	mould	51
engine	6	mat space	50	irrep	53
qual	87	fail	27	comp	87
deco	26	finish	74	cond	31

Table 7: Size (number of turns) of the training data per dialog move for the whole game classifiers reported in Table 3. We have imbalanced datasets with a total 2909 instances.

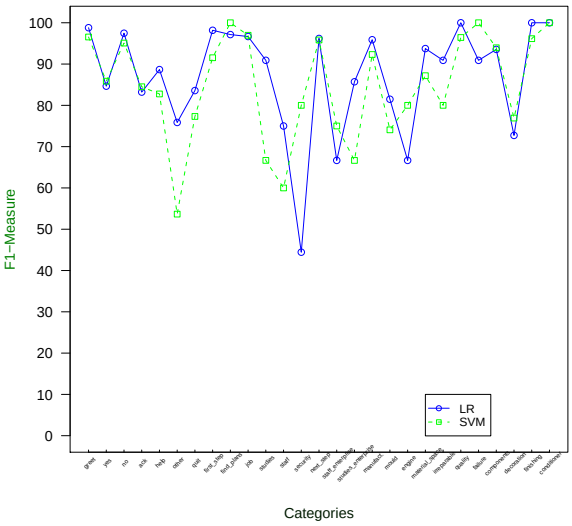

Figure 6: F1-measure per dialog move for the best global accuracy for the entire game (LR=90.26 and SVM=88.55 in Table 3).

the training data).
Interaction with the Midiki Rule-Based System. As already mentioned, we integrated the classifiers with a Midiki rule-based dialog manager which chooses how to respond to a player's utterance based on (i) the dialog move associated by the classifier for this utterance and (ii) the current information state in the rule-based dialog manager. In this way, we can take into account the current game context to correct misclassification errors. For instance, when using the classifier trained on the entire game dialog corpus, a turn can be misclassified whereby a dialog move is predicted by the classifier which cannot be interpreted in the context of the current subdialog. In some cases, this can easily be corrected by taking into account the information present in the information state. For instance, if (i) the classifier analyses the player's turn as a YES turn and (ii) the previous system utterance is a welcome greeting, the Midiki rule-based system will ignore the classifier's prediction and offer a response appropriate to the current context such as for instance, offering help on how to achieve the next pending

goal.

An informal first evaluation suggests that, although the classifier trained on the whole data yields a better accuracy overall, in practice, this hybrid dialog management system yields more coherent dialogs when combined with the classifiers trained on subdialogs. We are currently carrying out a more in-depth evaluation of this point.

7. Conclusion

We presented tools, data and results obtained in a project whose final aim is to develop French speaking virtual agents for serious games.

The tools include two wizard-of-Oz interfaces supporting free chat and guided interactions respectively; a Midiki based open agent architecture that supports asynchronous communication between a game agent, a speech recognition module, sentence interpreters, rule-based or statistical dialog managers and a relational database for storing interactions. The wizard-of-Oz interfaces can furthermore be customised for new games as illustrated by an experiment with a group of graphics students.

The data collected was manually annotated with an annotation schema integrating general communicative acts with domain specific information about the game (mainly game goals). And the resulting annotated data was used for training classifiers associating with each player's move a dialog move that is then used to determine the game's response.

We are currently exploring two main directions for further evaluating and improving the dialog engine.

First, we are investigating different ways of overcoming data sparseness and of providing the classifier with more training data: We are examining how automatically extracted synonyms can be used to handle words that have not been seen in the training data; how different similarity metrics can be used to handle mistyped words; and how paraphrase acquisition techniques can be used to increase the size of the training data. One particular aim here is to develop a general methodology for quickly acquiring enough training data to develop a dialog engine for learners of French as a second language.

Second, we are conducting a detailed comparative evaluation of how the different learning configurations described in this paper impact not the accuracy of the system but the interaction with a player. That is, we seek to complement the intrinsic evaluation provided in this paper with a task-based, extrinsic evaluation.

8. Acknowledgments

The research presented in this paper was partially supported by the Eurostar EmoSpeech project and by the European Fund for Regional Development within the framework of the INTERREG IV A Allegro Project. The authors would like to thank the students of the master of computational linguistics at Lorraine University, for their evaluation of the annotation schema, Professor Erwan Mahe (University of Rennes), for organizing the scriptwriter workshop and the students for testing the portability of our tools in the design of other scenarii.

9. References

Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. 2010. Towards an iso standard for dialogue act annotation. In *Proceedings of LREC 2010, the Seventh International Conference on Language Resources and Evaluation*, Malta.

C. Burke, C. Doran, A. Gertner, A. Gregorowicz, L. Harper, J. Korb, and D. Loehr. 2003. Dialogue complexity with portability?: research directions for the information state approach. In *Proceedings of the HLT-NAACL 2003 workshop on Research directions in dialogue processing - Volume 7*.

J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson. 1996. HCRC dialogue structure coding manual. TR-82, HCRC. Technical report, HCRC Edinburgh.

Chih C. Chang and Chih J. Lin, 2001. *LIBSVM: a library for support vector machines*.

Adam Cheyer and David Martin. 2001. The open agent architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):143–148.

S. Gandhe, A. Taylor, J. Gerten, and D. R. Traum. 2011. Rapid development of advanced question-answering characters by non-experts. In *Proceedings of the SIGDIAL 2011 Conference, The 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 347–349, Portland, Oregon, USA, 06/2011. The Association for Computer Linguistics.

P. A. Heeman and J. Allen. 1994. The TRAINS 93 dialogues. TRAINS technical note 94-2. Technical report, Department of Computer Science, University of Rochester.

A. Leuski and D. R.. Traum. 2010. Practical language processing for virtual humans. In *Twenty-Second Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-10)*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Jeff Orkin and Deb Roy. 2010. Semi-automated dialogue act classification for situated social agents in games. In *Proceedings of the Agents for Games and Simulations Workshop at the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Toronto, Canada.

John Searle. 1969. *Speech Acts*. Cambridge University Press.

David R. Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In Jan Kuppevelt, Ronnie W. Smith, and Nancy Ide, editors, *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*, pages 325–353. Springer Netherlands. 10.1007/978-94-010-0019-2_15.

dialog: dial01–Mon Jan 16 16:25:50 CET 2012
>**DumDee:**Bonjour ! Nous sommes Tweedledum et Tweedledee !
Si vous nous croyez vivants vous devez nous parler, c'est plus correct !
Les gens polis, quand ils rendent visite, disent à leur hôte :
Votre santé est bonne, bonne, bonne Votre santé est bonne et
puis vous serrent la main. Voilà la politesse !
>**DumDee:**Quel âge as-tu/toi?
>**Visitor:**J'ai 10 ans.
>**DumDee:**Vous êtes vieux, Père François, vous perdez vos dents
Vous aurez avant peu cent dix ans. Malgré vos années vous marchez sur la tête.
À votre âge croyez-vous que c'est bien malin ? À votre âge croyez-vous que c'est bien ?
>**Visitor:**Je ne sais pas.
>**DumDee:**Vous n'allez pas vous en aller déjà ! Oui, vous venez tout juste d'arriver !
Tu préfères l'histoire du pêcheur de tutus ou celle de la grenouille en vermicelles ?
>**Visitor:**je préfère l'histoire du pêcheur de tutus.
>**DumDee:**C'est logique ! Alors approche ton oreille ! 1 2 3 4... Ah, tu es curieux ?
Alice aussi était curieuse, n'est-ce pas ? Oui oui, curieuse !
Tu te souviens ce qui lui est arrivé ?
>**Visitor:**Non
>**DumDee:**Un lapin blanc en retard, en retard ! Qu'elle a poursuivi. Et il est parti.
Essaye de le trouver vite. Tu vas être en retard ! Mais ne pars pas tout de suite !
Si tu restes un peu, tu verras une belle bagarre ! Ou des acrobaties ! Tu veux ?
>**Visitor:**Non, je veux ma maman.
>**DumDee:**Oh, quand j'étais tout gamin j'avais un peu peur d'abîmer ma cervelle.
Mais je n'ai de cervelle pas plus gros qu'un petit pois.
Qui grelotte, qui grelotte dans ma tête. Ouh ! Repos ! Dodo ! Changeons de conversation !
Il est temps que tu retrouves nos compagnons. A plus tard !

Figure 7: Example of a French dialog collected by the scriptwriter students of the University of Rennes, who were asked to design an exposition about Alice's adventures in wonderland.