CLASSiC

D3.5 Metrics for the evaluation of user simulations

Olivier Pietquin, Helen Hastie

Distribution: Public

The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	216594
Project acronym	CLASSiC
Project full title	Computational Learning in Adaptive Systems for Spoken Conversation
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2008 / 36 Months

Security	Public
Contractual date of delivery	M24 = February 2010
Actual date of delivery	March 2010
Deliverable number	3.5
Deliverable title	D3.5 Metrics for the evaluation of user simulations
Type	Report
Status & version	Final 1.0
Number of pages	19 (excluding front matter)
Contributing WP	3
WP/Task responsible	SUPELEC
Other contributors	HWU
Author(s)	Olivier Pietquin, Helen Hastie
EC Project Officer	Philippe Gelin
Keywords	Metrics

Contents

Executive summary

This document explains the work done for Task 3.5 and refers to simulations built in Tasks 3.2, 3.3 and 3.4. This deliverable (D3.5) is a report on metrics used for evaluating user models in the framework of spoken dialogue simulation. User simulation has become an important trend of research in the field of spoken dialogue systems because collecting and annotating real interactions with users is often expensive and time consuming. Yet, such data are required for designing and assessing efficient dialogue systems. The general goal of user simulation is thus to produce as many as necessary natural, varied and consistent interactions from as few data as possible. Metrics are required in order to assess the quality of these simulated dialogues. In this report, we list the desired features of such metrics, we discuss the state-of-the-art in this field and propose new extensions to the described metrics.

Publications arising from this work are :

- Srini Janarthanam and Oliver Lemon, A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies, *Proceedings of SIGDial 2009*, London, September 2009.

- Olivier Pietquin, Stéphane Rossignol, and Michel Ianotto. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1rst International Workshop on Spoken Dialogue Systems Technology*, page 4 pages, Irsee (Germany), December 2009.

- Olivier Pietquin, Natural Language and Dialogue Processing. In *Multi-modal signal processing: methods and techniques to build multimodal interactive systems*, Jean-Philippe Thiran, Hervé Bourlard, Ferran Marqués, ISBN: 0123748259, Elsevier Science & Technology Books, 4:61-90, 2009.

A special session at the InterSpeech 2009 conference (Brighton, UK) has also been organized to promote the CLASSiC project and work done in spoken dialogue simulation and evaluation. A special issue of the ACM Transactions on Speech and Language Technologies on Machine Learning for Robust and Adaptive Spoken Dialogue Systems has also been launched recently.

Chapter 1

Introduction

From the mid 90's user simulation has become an important trend of research in the field of spoken dialogue systems (SDS) [9, 45, 10, 7, 25, 34, 13, 27], because collecting and annotating real human-machine interactions is often expensive and time consuming. Yet, such data are generally required for designing, training and assessing dialogue systems [18, 36, 19, 26, 33]. Especially when using machine learning methods for optimising dialogue management strategies such as Reinforcement Learning (RL) [38], the amount of data necessary for training is larger than existing corpora. Indeed, exploring the whole dialogue state space and strategy space requires a number of interactions that increases exponentially with the number of states while even simple dialogue systems have continuous state spaces (because of the inclusion of speech recognition and understanding confidence levels into the state description). User simulation is, therefore, necessary to expand data sets. The general goal of a user simulation is thus to produce as many as necessary natural, varied and consistent interactions from as few data as possible. The quality of the user simulation is, therefore, of crucial importance because it dramatically influences the results in terms of SDS performance analysis and learnt strategy [32]. Assessment of the quality of simulated dialogues and user simulation methods is an open issue and, although assessment metrics are required, there is no commonly adopted metric [31, 11]. In this deliverable, we will first define a list of desired features of a good user simulation metric. Secondly, state-of-the-art of metrics described in the literature are presented in Chapter 2. Finally, some new directions are proposed in Chapter 3.

1.1 Desired features

Although several theoretical and experimental comparisons of user simulation metrics can be found in the literature [45, 31, 35], none of these papers provides a list of desired features for a good user simulation metric. In the following, such a list is provided and will be used to judge the metrics described in the rest of the text.

To do so, it is necessary to provide a clear idea of the purpose of a user simulation. In the context of the CLASSiC project, user simulation is required to expand data sets used for training RL-based dialogue managers [17, 37, 36, 23, 43] and natural language generation systems [13, 12, 14]. This provides at least two requirements for the user simulation evaluation metric: it should assess how well the simulation fits to the original data statistics (*consistency*) and it should result in efficient strategies when used for training RL-based systems (*quality of learnt strategy*).

User simulation can also be used to assess the quality of a spoken dialogue system, whatever the method used to design its management strategy (e.g. machine learning, rule-based or hand-crafted policies) [9, 41, 36, 19]. A good user simulation metric should, therefore, predict how well it can be used to predict the performance of a spoken dialogue system (which maybe different from the one used to collect data) when interacting with real users (*performance prediction*).

Another goal of user simulation is to expand existing datasets. It is, therefore, important to measure the capability of the user simulation to generate unseen dialogues (*generalisation*).

Ideally, the metric should allow *ranking* of different user simulation methods. Practically, it should, therefore, be a scalar metric or such a scalar number should be computable from the metric. As a side-effect, a scalar metric could be used as an *optimisation criterion* to use statistical methods applied to parameter search for user simulation.

An efficient user simulation should not only reproduce the statistical distribution of dialogue acts measured in the data but should also reproduce complete dialogue structures. The optimal metric should, therefore, measure the ability of the user simulation to generate *consistent sequences* of dialogue acts.

There exists a lot of application domains where spoken dialogue can be useful. The metric should, therefore, be task independent and should apply to any domain (*task-independence*). The metric should also be independent of the dialogue management system used. Even if the task is similar, the SDS can be different and the user simulation evaluation metric should not be affected.

Finally, the metric should of course be *automatically computed* from objective measures and should not require any external human intervention.

To summarise, an evaluation metric for user simulation should allow to:

- measure statistical consistency of generated dialogue acts with data;

- assess the quality of learnt strategies when the user simulation is used to train a machine-learning dialogue management system;

- predict the performance of a spoken dialogue system with real users;

- measure the generalisation capabilities of the method;

- compute a scalar value to rank and optimise user simulation;

- measure the ability to generate consistent sequences of dialogue acts;

- evaluate user simulation independently from the task and the SDS;

- automatically compute an assessment measure from objective information.

These criteria will be used as a framework for describing state-of-the-art metrics, and will help identify where these metrics are lacking, providing new avenues of research for designing optimal metrics.

Chapter 2

State-of-the-art metrics for evaluating user simulations

In this chapter, the state-of-the-art in user simulation evaluation is provided, reflecting the most used evaluation methods in the literature of the last decade. There are many ways to cluster these methods. In [45, 35] the authors distinguish two categories: direct methods which assess the user simulation by testing the quality of its predictions and indirect methods which attempt to measure the quality of a user model by evaluating the effect of the model on the dialogue system performance. We will take a different approach, splitting methods into *local methods* which measure turn-level statistics and *global methods* which measure dialogue-based statistics.

2.1 Turn-level metrics

A large number of early metrics are based on a turn-level context. They measure local consistency of generated data and originally collected data with real users. They can take the form of distributions or of a set of scalar measures. They all share the major drawback of failing to measure the *consistency* of sequences of dialogue acts. Yet they also provide some useful information. All the metrics described in this section are *direct* measures.

2.1.1 Dialogue act statistics

As a human-machine dialogue can be considered as a sequence of dialogue acts uttered in turn by the human user and the dialogue manager, it is natural to compare statistics related to dialogue acts used in real and simulated dialogues. In a goal-driven dialogue, the dialogue acts can be open questions, closed questions, implicit or explicit confirmations but also greetings and dialogue closures. The first set of metrics that compares real and simulated dialogues is the measure of the relative frequency of each of the dialogue acts [23, 31]. This provides a histogram of dialogue act frequencies for each data set (real and simulated). It allows for comparison of dialogue styles, for example, are there more or less confirmations or open questions in one of the data sets. Such metrics were used to assess the Bayesian-network-based simulated user developed in the CLASSiC project Task 3.2 and described in deliverable D3.2.

In [31], the authors propose other statistics related to dialogue acts such as:

- the ratio of user and system acts, which is a measure of the user participation;

- the ratio of goal-directed actions vs. grounding actions vs. dialogue formalities vs. misunderstand-ings;

- the proportion of slot values provided when requested, which is a measure of the user cooperative-ness.

When comparing the above-mentioned metrics with respect to the desired features listed in Section 1.1, one can see that these metrics allow for comparison of similarities with actual data but have several short-comings. They do not allow the computation of a single scalar measure for ranking methods. It is also difficult to use them to predict performance of a SDS when used with real users. Finally, generalisation is also difficult to assess.

2.1.2 Precision, Recall, (Expected) Accuracy

Precision and Recall are common measures in machine learning and information retrieval and are used to evaluate the user model developed within the Task 3.4 (to be reported in deliverable D3.4). Precision and recall measure how well a model predicts observed values. A user model can be considered as a predictor of a dialogue act given some context (which can be more or less rich). These metrics are widely used in user simulation and even outside the realm of spoken dialogue systems [45]. Precision and Recall are defined as:

$$\text{Precision: } P = 100 \times \frac{\text{Correctly predicted actions}}{\text{All actions in simulated response}};$$

$$\text{Recall: } R = 100 \times \frac{\text{Correctly predicted actions}}{\text{All actions in real response}}.$$

These two measures are complementary and cannot be used individually to rank user simulation methods. Yet, the classical balanced F-measure [39] can be used to combine both these measures and obtain a single scalar:

$$F = \frac{2PR}{P+R}.$$

Another related metric is the accuracy as defined in [45]:

- Accuracy: "percentage of times the event that actually occurred was predicted with the highest probability"

- Expected accuracy: "Average of the probabilities with which the event that actually occurred was predicted"

One of their major drawbacks is that they do not measure well the generalisation capabilities of the model. In fact, these metrics actually penalise attempts to generalise since when the model generates unseen dialogues, their scores are lower.

2.1.3   Kullback-Leibler (KL) divergence and dissimilarity

In Section 2.1.1, metrics based on frequencies of dialogue acts have been defined. Histograms of frequencies are obtained for both the simulated data and the human-machine data. One way to obtain a single scalar value from these histograms is to compute a statistical distance between the distributions they represent. Several statistical distances could be imagined but a common choice is the Kullback-Leibler (KL) divergence [16]. This measure is frequently used in the literature and is being adopted in the CLASSiC Task 3.4 (to be described in deliverable D3.4) for evaluating simulated users trained for the TownInfo domain. It has also been used as a comparison metric to assess the user models developed within Task 3.3 for the TownInfo and Self-Help domains. This is explained in deliverable D3.3 and discussed futher in Section 3.1.

The KL divergence between two distributions P and Q is defined by:

$$D_{KL}(P||Q) = \sum_{i=1}^{M} p_i log(\frac{p_i}{q_i}),$$

where p_i (resp. q_i) is the frequency of dialogue act a_i in the histogram of distribution P (resp. Q). Actually, the KL divergence is not a distance since it is not symmetric $(D_{KL}(P||Q) \neq D_{KL}(Q||P))$. To remedy this defect, the dissimilarity metric $DS(P||Q)$ is introduced:

$$DS(P||Q) = \frac{D_{KL}(P||Q) + D_{KL}(Q||P)}{2}$$

The KL divergence does have some drawbacks. It is an unbounded metric which is difficult to use for ranking. In addition, there is an unbalanced penalty between the estimation of the mean and the variance of the distributions. To be specific, it gives more importance to the similarity of the means of these two distributions than to the variances. Therefore, two distributions having the same means but very different variances will appear to be closer to each other than two distributions having slightly different means but similar variances. This is particularly prevalent for our spoken dialogue applications. KL divergence also requires a correct estimation of densities P and Q while traditionally only counts are available from data. It is also difficult to assess the generalisation capabilities of a user model with this metric since it penalises dialogue styles which are different from the real data. In Section 3.1, we discuss new applications of the KL divergence that seeks to resolve these last two issues.

2.1.4   Conclusion

Turn-level metrics are direct measures and, therefore, share the same shortcoming of being unable to assess the quality of generated sequences of dialogue acts. Some of them are useful to analyse the dialogues in terms of dialogue style or user initiative and cooperativeness. Yet, it is difficult to extract a scalar value from these metrics although the balanced F-measure and the Kullback-Leibler divergence are such scalar metrics. Nevertheless, these two metrics cannot be used to assess the generalisation capabilities of a model. These drawbacks are addressed in Chapter 3.

2.2   Dialogue-level metrics

In this section, metrics are presented that use higher-level information. They are based on complete dialogue properties instead of local turn information. Most of the metrics discussed in this section have been developed more recently than the turn-level metrics and attempt to achieve the goals listed in Section 1.1.

2.2.1   Task completion

A task-driven dialogue system assists a user to achieve a goal which is usually not known by the system before the interaction starts. The degree of achievement of this goal is referred to as *task completion*. One way to measure the task completion is the κ coefficient. The κ coefficient [4] is obtained from a confusion matrix M summarising how well the transfer of information performed between the user and the system. *M* is a square matrix of dimension *n* (number of pieces of information that have to be transmitted from the user to the system) where each element m_{ij} is the number of dialogues in which the value *i* was interpreted while value *j* was meant. The κ coefficient is then defined as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of correct interpretations (sum of the diagonal elements of M (m_{ii}) on the total number of dialogues) and $P(E)$ is the proportion of correct interpretations occurring by chance. One can see that κ = 1 when the system performs perfect interpretation ($P(A) = 1$) and κ = 0 when the only correct interpretations were obtained by chance ($P(A) = P(E)$). Other task completion measures can be defined like in [23].

To assess the similarity between artificially generated dialogues and human-machine dialogues, it is legitimate to measure the similarity in terms of task completion. In [36, 31, 24], the task completion or the task completion rate (ratio of successful dialogues) is used to compare dialogues. In [31], the authors also propose to use the completion time, that is the number of turns (or the actual dialogue duration) required to achieve a satisfying task completion. Task completion related metrics have been used to assess the user simulation methods developed within the CLASSiC Tasks 3.2 and 3.4 which are described respectively in deliverables D3.2 and D3.4.

Once again, considering the task completion implies several different metrics (task completion, rate, duration) and each of them provides different information. It is difficult to chose one of them for ranking user models. They cannot be used independently since two corpora sharing a same task completion rate may not contain similar dialogues. In addition, the completion time does not guarantee dialogue similarity. Each of these measures is an indirect and global measure.

2.2.2   Perplexity and log-likelihood

Perplexity is a measure that comes from information theory. It is generally used to compare probabilistic predictive models. In natural language processing, it is widely used to compare language models. Perplexity of a model is defined as follows:

$$PP = 2^{\sum_{i=0}^{N} \frac{1}{N} \log_2 p_m(x_i)},$$

where $p_m(x_i)$ is the probability of x_i given the model, and x_i is a sample from a test dataset containing N samples. If the model is good, it will tend to give high probabilities to the test samples since it is supposed to predict them and, therefore, have a low perplexity. In the case of a user model, the data to be predicted are sequences of dialogue acts $\{a_0, a_1, ..., a_n\}$ so $p_m(x) = p_m(a_0, a_1, ..., a_n)$ is the probability of a sequence of dialogue acts given the user model.

A similar metric, which is the log-likelihood of the data given the model, has been used to evaluate the agenda-based user model designed during Task 3.4. This is the log-likelihood $\mathcal{L}(x)$ of a data set $x = \{x_i\}_{i=1,...,N}$ given a model m is defined by $\mathcal{L}(x) = \log p(x|m) = \log p_m(x)$. If the data samples x_i are assumed to be independent (a common assumption), $\mathcal{L}(x)$ can be written as:

$$\mathcal{L}(x) = \log \prod_{i=1}^{N} p_m(x_i) = \sum_{i=1}^{N} \log p_m(x_i)$$

The higher the log-likelihood is, the higher the consistency between the data and the model.

Perplexity and log-likelihood are scalar numbers that can be used to rank user models. They are also global or dialogue-level metrics as they are based on the probability of sequences of dialogue acts. Yet, as explained above, the perplexity and log-likelihood measure how well a model is able to predict data. Therefore, it is very difficult to use them to measure the generalisation capabilities of a model and the ability to generate unseen sequences that are still reasonable in the context of a dialogue.

2.2.3   HMM similarity

One particular statistical model that can be used to predict sequences of dialogue acts (or dialogue states) is the Hidden Markov Model (HMM). In [7] the authors propose to train a HMM on a corpus of real human-machine dialogues (the hidden state is the dialogue state while the observations are dialogue acts) and to use the HMM as a generative model to produce artificially generated data. To assess the quality of the model, the authors propose to generate a corpus of artificial data using the trained HMM and then to train a new HMM on these generated data. The metric proposed to assess the model is then a distance between the HMM trained on real data and the one trained on artificially generated data. Computing a distance between HMMs is not an easy problem. In [7], the authors choose to use the dissimilarity introduced in Section 2.1.3 based on the Kullback-Leibler divergence of distributions encoded by the two HMMs.

This evaluation method does not directly measure the dissimilarity between corpora but instead it measures the dissimilarity of the models by computing a distance between distributions encoded by the models (which is possible because HMMs are well known). One can, therefore, assume that the measure captures more than what is in the data and that unseen situations can be taken into account. Yet, this has not been experimentally demonstrated by the authors of [7]. Therefore, it could be used whatever the user model as the artificially generated data can be produced by any model, not only HMM-based models. It provides a single scalar, yet it does not directly provide any information about the quality of the interaction between real users and a new dialogue system, i.e. it does not adapt to new dialogue management strategies.

2.2.4   Cramér-von Mises divergence

In many applications, a user model for simulation can be viewed as a predictor of the performance of a spoken dialogue systems. In [42], the author describes a metric based on this point of view and built upon

the following statements:

1. "For a given dialogue system D and a given user population U_0, the goal of a user simulation U_1 is to accurately predict the performance of D when it is used by U_0."

2. "The performance of a dialogue system D in a particular dialogue $d_{(i)}$ can be expressed as a single real-valued score $x_{(i)}$, computed by a scoring function $Q(d_{(i)}) = x_{(i)}$."

3. "A given user population U_0 will yield a set of scores $S_0 = x^0_{(1)}, x^0_{(2)}, ..., x^0_{(N_0)}$. Similarly, a user simulation U_1 will yield a set of scores $S_1 = x^1_{(1)}, x^1_{(2)}, ..., x^1_{(N_1)}$."

4. "A user simulation U_1 may be evaluated by computing a real-valued divergence $D(S_0||S_1)$."

The proposed metric is thus a divergence measure that expresses how well the distribution of scores obtained by the real users S_0 is reproduced by the user simulation S_1. This divergence could have been the Kullback-Leibler divergence described in Section 2.1.3 but the author argues that this metric is accurate only if actual distributions are known or well approximated. This is rarely the case, he argues, in the field of dialogue systems where data are tricky to obtain and usually low amounts of data are available. The normalised Cramér-von Mises divergence [5, 3] is less demanding with this respect because it is based on the *empirical distribution function* which does not make any assumption about the distribution of data. It provides a real number ranging from 0 to 1 and is computed as follows:

$$D_{CvM}(F_0||F_1) = \alpha\sqrt{\sum_{i=1}^{N_0}(F_0(x^0_{(i)}) - F_1(x^0_{(i)}))^2},$$

where F_j is the empirical distribution function (EDF) of the data $S_j = (x^j_{(1)}, x^j_{(N_j)})$ and α is a normalising constant given by $\alpha = \sqrt{\frac{12N_0}{4N_0^2-1}}$. By definition, the EDF is:

$$F_j(x) = \frac{1}{N_j}\sum_{i=1}^{N_j}\begin{cases} 1 & \text{if } x^j_{(i)} < x \\ \frac{1}{2} & \text{if } x^j_{(i)} = x \\ 0 & \text{if } x^j_{(i)} > x \end{cases}$$

This metric addresses many of the desired features listed in Section 1.1 since it provides a bounded scalar value usable for ranking, it is a global feature that also predicts the performance of a dialogue system. Since it is based on scores, it is not sensitive to unseen situations given that they result in similar scores. This does not mean that the generalisation capabilities of the model are assessed but at least, they do not result in a reduction of the metric value. One drawback of this metric is that it does not measure the degree of similarity of sequences of dialogue acts and, therefore, dialogues may not be realistic even for high values of the metric.

2.2.5 BLEU and Discourse-BLEU

The BLEU (Bilingual Evaluation Understudy) score [22] is widely used in machine translation. It is a metric that compares two semantically equivalent sentences. Generally it is used to compare an automatically translated sentence to several reference sentences generated in the target language by human experts. The

authors of [22] argue that "the closer a machine translation is to a professional human translation, the better it is". The BLEU score is the geometric mean of the n-gram precisions with a brevity penalty. It is actually computed as follows. Let C be a corpus of human-authored utterances and S be a translation candidate. First, a precision score is computed for each n-gram:

$$ p_n = \frac{\sum_{S \in C} \sum_{ngram \in S} count_{matched}(ngrams)}{\sum_{S \in C} \sum_{ngram \in S} count(ngrams)}, $$

where $count_{matched}(ngrams)$ is the number of ngrams in S that match the n-grams in the corpus. The brevity penalty is defined as:

$$ BP = \begin{cases} 1 \text{ if } g > r \\ e(1 - r/g) \text{ if } g \leq r \end{cases} $$

where g is the length of the generated utterance while r is the length of the reference. Finally, the BLEU score is computed as:

$$ \text{BLEU} = BP^{(\sum_{n=1}^{N} \frac{1}{N} \log p_n)} $$

Using the brevity penalty, BLEU penalises abnormally short utterances because short generated utterances have higher n-gram precision.

In [15], the authors propose to use the same metric to compare a sentence automatically generated by a user simulation and sentences produced by human experts or available in the training data. It is indeed the same task since in both cases the problem is to compare surface realisations of a semantic target. The BLEU measure can, therefore, be used to measure the naturalness of a given utterance.

As described above, the BLEU metric can be used to measure the naturalness of a simulated utterance but what we are interested in is a measure of naturalness of dialogues (that is a sequence of utterances). To achieve this goal, the authors of [15] propose another metric they call Discourse-BLEU (D-BLEU). It is designed to measure the similarity of simulated dialogue and human-machine dialogues. D-BLEU is also the geometric mean of the n-gram precisions with a brevity penalty but the n-grams considered here are not sequences of words but sequences of intentions (user intentions and systems intentions). D-BLEU is, therefore, computed in the same way as BLEU by replacing words by intentions. D-BLEU score ranges from 0 to 1 and gives higher scores to similar dialogues (especially if they have the same length).

The BLEU scores is known to be highly correlated with human judgement [22, 8] and [15] argues that D-BLEU also follows the same tendencies as human judgement. In some cases, BLEU has been reported to fail to predict improvements in the machine translation domain. In some cases, an increased BLEU score does not reflect enhanced naturalness. Too few studies are reported about the D-BLEU score for dialogue to allow one to draw the same conclusions, yet the BLEU and D-BLEU are quite similar in their definitions. Once again, this metric also fails to measure the generalisation capabilities of the user simulation.

2.2.6 SUPER

SUPER (Simulated User Pragmatic Error Rate) [29, 28] is an attempt to combine different metrics so as to take advantage of their respective features. The aim of SUPER is to measure the naturalness and variety of artificially generated dialogues. Naturalness is actually measured as a mix of *completeness* and *consistency*. It is based on the three following statements:

- The simulated user should not produce intentions that real users would not produce in the same context. It should not create insertions (*I*) of intentions. This relates to *consistency*.

- The intentions generated by the simulated user should cover the whole range of intentions generated by real users. It should not create deletions (*D*) of intentions. This relates to *completeness*.

- The user should generalise and produce a sufficient *variety* (*V*) of behaviours and not reproduce exactly the real users' behaviour. A lower bound ε is defined to reflect the desired variation and an upper bound δ is defined to reflect undesired variation.

The variables *I*, *D* and *V* are computed as follows:

Consistency:
   if $(P_0(action) = 0$ and $P_1(action) > 0) : I = (-1)$
Completeness:
   if $(P_0(action) > 0$ and $P_1(action) = 0) : D = (-1)$
Desired variation:
   if $(|P_0(action) - P_1(action)| < \varepsilon) : V = (+1)$
Tolerated variation:
   if $(\varepsilon < |P_0(action - P_1(action)| < \delta) : V = (0)$
Penalised variation:
   if $(\delta <= |P_0(action) - P_1(action)|) : V = (-|P_0(action) - P_1(action)|)$

---

where P_0 is the observed user action probability for one system action and P_1 the probability given by the user model. The SUPER score is then given by:

$$\text{SUPER} = \frac{1}{m} \sum_{k=1}^{m} \frac{V + I + D}{n},$$

where *n* is the number of possible user acts and *m* the number of contexts. Notice that the SUPER score is similar to the Word Error Rate (WER) measurement used to assess speech recognition systems.

The SUPER score addresses many of the desired features described in Section 1.1, yet it is not a direct measure of the ability of the user model to predict the performances of a spoken dialogue system when used with real users.

2.2.7   Human evaluation

In [1], the authors propose to use human judges to evaluate automatically generated corpora. In this approach, human judges serve as a gold standard for user simulation assessment. This choice is based on several arguments. First, it provides a way to evaluate how hard it is to distinguish between simulated and real dialogues. If a human judge performs a bad classification, the machine is likely not to perform better. Second, a new metric could be developed by using human judgement as a gold standard. This new metric should predict this judgement using objective measures. Finally, comparing human judgement with automatically computed scores can help in validating the quality of the metric.

The study reported in [1] is based on very subjective questions asked to the human judges observing dialogues between a student and a tutor. It subsequently uses the scores provided by human judges to

train different metrics with supervised learning methods (stepwise multiple linear regression and ranking models). The study concludes that the latter method is able to mimic correctly human judgements and could be used to evaluate new simulated dialogues.

This method is very close to the PARADISE [41] metric that evaluates SDS strategies by predicting user satisfaction from real interactions. Of course, the major drawback of this method is that it requires human judges to score the dialogues. It is time consuming and it is always difficult to know how many human judges should be involved (e.g. what is the protocol for reaching a meaningful inter-annotator agreement?). The metrics are also trained for a specific application and it is very difficult to tell how such a metric could generalise to other domains.

2.2.8   Quality of learnt strategy

As user simulations are frequently used for training an optimisation algorithm (such as a RL algorithm) for SDS management strategies, [2] proposes to measure the performance of the SDS when trained against different user models. The performance is measured as an expected cumulative reward [44] when applying a learnt strategy. The performance of the SDS is used as an assessment measure for the user simulation. Also, the transition probabilities $P(s_{i+1}|s_i, a_i)$ (where s_i is the dialogue state and a_i the system's dialogue act at time step i) are measured in both the real data and the simulated data and compared state by state.

This method suffers from a bootstrapping problem. It requires testing the dialogue system on real users after training to obtain the quality measurement while it is precisely the goal of the user model to predict the performance of the trained system when used with real users. Concerning the comparison of the transition probabilities, it is very similar to the methods exposed in Section 2.1.1 and raises the same issues. Moreover, the study reports conclusions that can seem confusing, for example, a completely random user model is reported to perform quite well.

2.2.9   Conclusion

This section surveyed the large majority of dialog-level metrics used to assess user simulation. Dialog-level metrics can be direct (perplexity, HMM similarity, D-BLEU, SUPER) or indirect (task completion, Cramér-von Mises divergence, human evaluation, quality of learnt strategy). None of these metrics possess all the features listed in Section 1.1. Yet, the Cramér-von Mises divergence is one metric presenting most of the desired features together with the SUPER score. The Cramér von Mises divergence is able to predict the performance of a dialogue system with real users while it is not able to judge if the user simulation can generalise to unseen situations. The SUPER score is able to measure generalisation capabilities although it cannot be used to predict performance of a dialogue system as such. Only D-BLEU really focuses on the naturalness of generated dialogue but it also fails in predicting performances of a SDS when interacting with real users. It also shares the disadvantages of the BLEU metrics widely used in machine translation. All the described methods provide a scalar value and could be used for ranking and optimising user simulation.

Chapter 3

Proposed extensions

In this section, we proposed new applications for the KL divergence and present a new, metric for ranking and optimisation of user simulations.

3.1 N-gram Kullback-Leibler divergence

As discussed in 2.1.3, KL divergence is a direct measure and captures the similarity between two distributions. However, it does not capture similarities between sequences of dialogue acts but only between frequency distributions of dialogue acts. It is, therefore, hard to tell whether dialogues are similar as they could simply use the same dialogue acts but not in the same order.

As discussed in Section 2.2.3, the Kullback-Leibler divergence has been computed by comparing the distribution captured by two HMMs, one being trained on the original data (containing real human-machine interactions) and the other being trained on the artificially generated data [7, 6].

As part of the CLASSiC Task 3.3 (reported in deliverable D3.3 and [13]), KL divergence has been applied in a new unique way for SelfHelp and TownInfo Natural Language Generation. In a similar way to the HMM approach discussed in Section 2.2.3, the Kullback-Leibler divergence is computed between the distributions captured by different advanced n-grams trained on the human-machine interaction corpus. We argue that the advanced n-gram model is a realistic model of each corpus since it takes into account all context variables and it is sufficiently smoothed to support variability in the generated sentences. This way, unseen dialogues are not penalised if they are "too far" from the distribution captured by the advanced n-gram thus solving the *generalisation* problem. It could be argued that this application of the KL divergence takes it from a turn-level metric to a dialogue-level one.

3.2 IRL-based metrics

Reinforcement Learning [38] is now a state-of-the-art method for optimising dialogue management systems [17, 37, 36, 23, 43]. It is based on the Markov Decision Processes (MDP) paradigm where a system is described in terms of states and actions. The goal of RL is to learn the best action to perform in each state according to a criterion called *reward function*. In the context of dialogue management, states are given by the dialogue context and actions are the different dialogue acts the dialogue manager can per-

form. The reward function is often defined as the expected user satisfaction which has to be maximized [37]. Reinforcement learning is, therefore, used to learn which dialogue act should be transmitted to the user given the dialogue context so as to maximize their satisfaction.

The reinforcement learning problem has a dichotomy: the inverse reinforcement learning (IRL) problem [30]. By observing an expert agent (mentor) performing optimally, IRL aims at discovering the reward function serving as optimisation criterion to the expert. Once this reward function is learnt, an artificial agent can be trained upon this function so as to mimic the expert's behaviour. The main problem of IRL is that there exists an infinite number of reward functions explaining the expert's behaviour including trivial solutions and constraints have to be added to obtain a solution usable for optimising an artificial agent [20].

The idea of using IRL for dialogue management optimisation has been proposed in [21]. In this paper, the authors propose to use IRL on data collected from human-human interactions to learn the policy followed by the human operator. The goal is to mimic the human operator's behaviour considering it as optimal. Several criticisms can be made about this approach. Firstly, in most of the real-world applications, human operators are instructed to follow decision-tree-based scenarios. Imitation of the human operator would result in learning this decision-tree and nothing can ensure that this is optimal. Secondly, even if the human operator can freely interact with the users, optimality from the user satisfaction point of view is not guaranteed. Thirdly, when interacting with a human, users adopt different behaviours than when interacting with machines [40].

In this deliverable, we propose to use IRL in a different manner. We argue that if the human operator may not be optimally acting to maximize the users' satisfaction, the users are unconsciously trying to optimise their satisfaction when interacting with a machine. IRL could, therefore, be used to learn the internal (non-observable) reward function that users naturally try to maximize.

There are several advantages to this approach. Firstly, IRL algorithms can be trained on human-machine interaction data which is easier to automatically annotate than human-human interaction data. Secondly, the learnt reward function can serve as a metric for user simulations since a user simulation that performs badly according to this function is probably not reproducing real users' behaviour. Finally, this reward function can serve as an optimisation criterion to train a user simulation that is independent from the dialogue management system. Indeed, the reward function optimised by the user is related to their satis-faction and not to the actual performance of the system. Therefore, if the SDS policy is modified, the user simulation should change its behaviour so as to continue maximizing the reward function as a real user would change their behaviour so as to continue maximizing their satisfaction.

This last feature is important for training RL-based dialogue management systems since RL involves a trial-and-error process aiming at incrementally improving the interaction policy. The policy thus changes frequently and current user models that are trained from data collected with a fixed interaction policy, cannot adapt their behaviour according to modifications of the SDS.

The metric obtained from IRL addresses many of the features listed in Section 1.1. It provides a single scalar value that is automatically computed and can serve to rank and optimise user models. It can be used to predict performances of real users when interacting with a SDS since this metric is related to user satisfaction. It is not sensitive to unseen dialogues that are generated by simulation since if they reach good performance, the dialogues themselves are not important but can be judged as realistic from the user's point of view. It can be automatically obtained whatever the task since nothing is task-dependent in the IRL paradigm.

Several issues have still to be solved. Firstly, there is inter-user variability which makes the notion of

satisfaction user-dependent. Thus, it is hard to compute a single reward function for a whole set of users. A method for automatically clustering the user population according to the metric while learning this metric is required. Moreover, users may not always be optimal according to their internal reward function (they can make errors) and a tolerance factor has to be included in the learning algorithm.

Chapter 4

Conclusions

This deliverable contains several contributions. Firstly, a list of desired features for user simulation evaluation metrics is provided in Section 1.1. This list serves as comparison criteria for state-of-the-art metrics that can be found in the literature. A comprehensive list of state-of-the-art metrics for assessing user simulation are listed in Chapter 2. Instead of using the standard direct/indirect classification, metrics are described according to their level of analysis, specifically turn-level or dialogue-level analysis. For each of these metrics, pros and cons are listed and it is evident that no one single metric does fulfill all of the desired features. A collection of metrics from this section have been applied to user simulations in Tasks 3.2 and 3.4.

Section 3.1, presents work done in Task 3.3 using the Kullback-Leibler divergence for comparing N-grams used for Natural Language Generation. This approach addresses many of the drawbacks of the Kullback-Leibler divergence and shows that it can be useful for ranking user simulations.

Finally in Section 3.2, we present a new, innovative metric based on Inverse Reinforcement Learning (IRL). This last metric raises several scientific issues and as yet has not be implemented and tested on real dialogue data.

Bibliography

[1] H. Ai and D. Litman. Assessing dialog system user simulation evaluation measures using human judges. In *Proceedings of the 46th meeting of the Association for Computational Linguistics*, pages 622–629, Columbus, Ohio (USA), 2008.

[2] H. Ai and D. Litman. Setting up user action probabilities in user simulations for dialog system development. In *Proceedings 47th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, Singapore, 2009.

[3] T. Anderson. On the distribution of the two-sample Cramér-von Mises criterion. *Annals of Mathematical Statistics*, 33(3):1148–1159, 1962.

[4] J. Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[5] H. Cramer. On the composition of elementary errors. second paper: Statistical applications. *Skandinavisk Aktuarietidskrift*, 11:171–180, 1928.

[6] H. CuayAhuitl. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. PhD thesis, University of Edinburgh, UK, 2009.

[7] H. CuayAhuitl, S. Renals, O. Lemon, and H. Shimodaira. Human-computer dialogue simulation using hidden markov models. In *In Proc. of ASRU*, pages 290–295, 2005.

[8] G. Doddington. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 128–132, San Diego, CA, 2002.

[9] W. Eckert, E. Levin, and R. Pieraccini. User modeling for spoken dialogue system evaluation. In *Proc. ASRU'97*, December 1997.

[10] K. Georgila, J. Henderson, and O. Lemon. Learning user simulations for information state update dialogue systems. In *Proceedings of Interspeech 2005*, 2005.

[11] K. Georgila, J. Henderson, and O. Lemon. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. Interspeech'06*, September 2006.

[12] S. Janarthanam and O. Lemon. A Data-driven method for Adaptive Referring Expression Generation in Automated Dialogue Systems: Maximising Expected Utility. In *Proceedings of PRE-COGSCI 09*, 2009.

[13] S. Janarthanam and O. Lemon. A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies. In *Proceedings of SIGDIAL*, 2009.

[14] S. Janarthanam and O. Lemon. Learning Adaptive Referring Expression Generation Policies for Spoken Dialogue Systems using Reinforcement Learning. In *Proceedings of SEMDIAL*, 2009.

[15] S. Jung, C. Lee, K. Kim, M. Jeong, and G. G. Lee. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech & Language*, 23(4):479 – 509, 2009.

[16] S. Kullback and R. Leiber. On information and sufficiency. *Annals of Mathematical Statistics*, 22, 1951.

[17] E. Levin, R. Pieraccini, and W. Eckert. Learning dialogue strategies within the markov decision process framework. In *Proc. ASRU'97*, December 1997.

[18] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, 2000.

[19] R. López-Cózar, A. de la Torre, J. Segura, and A. Rubio. Assesment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3):387–407, May 2003.

[20] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

[21] T. Paek and R. Pieraccini. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communication*, 50:716–729, 2008.

[22] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311318, 2002.

[23] O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. PhD thesis, Faculté Polytechnique de Mons (FPMs, Belgium), 2004.

[24] O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. Presses Universitaires de Louvain, 2004.

[25] O. Pietquin. Consistent goal-directed user model for realisitc man-machine task-oriented spoken dialogue simulation. In *Proc. ICME'06*, July 2006.

[26] O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589–599, March 2006.

[27] O. Pietquin, S. Rossignol, and M. Ianotto. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1rst International Workshop on Spoken Dialogue Systems Technology*, page 4 pages, Irsee (Germany), December 2009.

[28] V. Rieser. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*. PhD thesis, Saarland University, Dpt of Computational Linguistics, July 2008.

[29] V. Rieser and O. Lemon. Simulations for learning dialogue strategies. In *Proceedings of Interspeech 2006*, Pittsburg (USA), 2006.

[30] S. Russell. Learning agents for uncertain environments (extended abstract). In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103, New York, NY, USA, 1998. ACM.

[31] J. Schatzmann, K. Georgila, and S. Young. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. SIGdial'05*, September 2005.

[32] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young. Effects of the user model on simulation-based learning of dialogue strategies. In *Proceedings of ASRU'05*, December 2005.

[33] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *In Proceedings of ICASSP 07*, 2007.

[34] J. Schatzmann, B. Thomson, and S. Young. Statistical user simulation with a hidden agenda. In *In Proceedings of SigDial 07*, 2007.

[35] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2):97–126, June 2006.

[36] K. Scheffler and S. Young. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, 2001.

[37] S. Singh, M. Kearns, D. Litman, and M. Walker. Reinforcement learning for spoken dialogue systems. In *Proc. NIPS'99*, 1999.

[38] R. Sutton and A. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 1998.

[39] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.

[40] M. Walker, D. Hindle, J. Fromer, G. D. Fabbrizio, and C. Mestel. Evaluating competing agent strategies for a voice email agent. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech'97)*, Rhodes (Greece), 1997.

[41] M. Walker, D. Litman, C. Kamm, and A. Abella. Paradise: A framework for evaluating spoken dialogue agents. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 271–280, 1997.

[42] J. Williams. Evaluating user simulations with the Cramer-von Mises divergence. *Speech Communication*, 50:829–846, 2008.

[43] J. Williams, P. Poupart, and S. Young. Partially observable markov decision processes with continuous observations for dialogue management. In *Proceedings of the SigDial Workshop (SigDial'06)*, 2005.

[44] J. D. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.

[45] I. Zuckerman and D. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11, 2001.