The Knowledge Engineering Review, Vol. 28:1, 59–73. © Cambridge University Press, 2012
doi:10.1017/S0269888912000343
First published online 28 November 2012

A survey on metrics for the evaluation of user simulations

OLIVIER PIETQUIN[1] and HELEN HASTIE[2]

[1]*SUPELEC – IMS-MaLIS Research Group, UMI 2958 (GeorgiaTech – CNRS), 2 rue Edouard Belin, 57070 Metz, France;*
e-mail: olivier.pietquin@supelec.fr;
[2]*School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK;*
e-mail: h.hastie@hw.ac.uk

Abstract

User simulation is an important research area in the field of spoken dialogue systems (SDSs) because collecting and annotating real human–machine interactions is often expensive and time-consuming. However, such data are generally required for designing, training and assessing dialogue systems. User simulations are especially needed when using machine learning methods for optimizing dialogue management strategies such as Reinforcement Learning, where the amount of data necessary for training is larger than existing corpora. The quality of the user simulation is therefore of crucial importance because it dramatically influences the results in terms of SDS performance analysis and the learnt strategy. 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. In this paper, we give a survey of User Simulations Metrics in the literature, propose some extensions and discuss these metrics in terms of a list of desired features.

1 Introduction

From the mid 90s, user simulation has become an important area of research in the field of spoken dialogue systems (SDSs) because collecting and annotating real human–machine interactions is often expensive and time-consuming (Eckert *et al.*, 1997; Zukerman & Albrecht, 2001; Cuayahuitl *et al.*, 2005; Georgila *et al.*, 2005; Pietquin, 2006; Schatzmann *et al.*, 2007b; Janarthanam & Lemon, 2009b; Pietquin *et al.*, 2009). Such data are generally required for designing, training and assessing dialogue systems (Levin *et al.*, 2000; Scheffler & Young, 2001; López-Cózar *et al.*, 2003; Pietquin & Dutoit, 2006; Schatzmann *et al.*, 2007a). Especially, when using machine learning methods for optimizing dialogue management strategies such as Reinforcement Learning (RL) (Sutton & Barto, 1998; Frampton & Lemon, 2010), the amount of data necessary for training is larger than existing corpora. In particular, it is important for methods such as RL to be able to explore the entire possible state space, including all the user responses that the system might have to deal with but that do not occur in the training data. Indeed, exploring the whole dialogue state space and strategy space requires a number of interactions that increases exponentially with the number of states. Furthermore, 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 (sometimes referred to as user modelling)[1] 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 little

[1] Notice that this naming is generally misleading since user modelling is more about inferring user's mental state than about producing consistent behaviours, which is the real aim of user simulation.

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 (Schatzmann *et al.*, 2005b). 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 (Schatzmann *et al.*, 2005a; Georgila *et al.*, 2006).

Previous publications include Frampton and Lemon (2010) who give a summary of RL methodology for dialogue management and Schatzmann *et al.* (2006) who give an overview of statistical user simulation techniques. This paper complements these publications by providing a survey of user simulation metrics. In this paper, we will first define a list of desired features of a good user simulation metric. Second, state-of-the-art metrics described in the literature are presented in Section 3. A new metric based on Inverse RL (IRL) is discussed in Section 4 and finally, a discussion of all metrics is provided in Section 5.

2 Desired features

Although several theoretical and experimental comparisons of user simulation metrics can be found in the literature (Zukerman & Albrecht, 2001; Schatzmann *et al.*, 2005a, 2006), none of these papers provide a list of desired features for a good user simulation metric. In this section, such a list is provided and will be used to judge the metrics described in the rest of the paper.

In order to do so, it is necessary to provide a clear idea of the purpose of a user simulation. User simulation is required to expand data sets used for training RL-based dialogue managers (Levin *et al.*, 1997; Singh *et al.*, 1999; Scheffler & Young, 2001; Pietquin, 2004; Williams *et al.*, 2005) and natural language generation systems (Janarthanam & Lemon, 2009a, 2009b, 2009c). This provides at least two requirements for the user simulation evaluation metric: it should assess how well the simulation fits the original data statistics (*consistency*) and it should result in efficient strategies when used for training RL-based systems (*quality of learnt strategy*). 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.

User simulation can also be used to assess the quality of an SDS, regardless of the method used to design its management strategy (e.g. machine learning, rule-based or hand-crafted policies; Eckert *et al.*, 1997; Walker *et al.*, 1997b; Scheffler & Young, 2001; López-Cózar *et al.*, 2003). A good user simulation metric should, therefore, predict how well it can be used to predict the performance of an SDS (which may be different from the one used to collect data) when interacting with real users (*performance prediction*).

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

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 *optimization criterion* to use statistical methods applied to parameter search for user simulation.

There are many 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 be able to:

- measure statistical consistency of generated dialogue acts with data (*consistency*);
- measure the ability to generate consistent sequences of dialogue acts (*consistent sequences*);
- assess the quality of learnt strategies when the user simulation is used to train a machine-learning-dialogue management system (*quality of learnt strategy*);

https://doi.org/10.1017/S0269888912000343

- predict the performance of an SDS with real users (*performance prediction*);
- measure the generalization capabilities of the method (*generalization*);
- compute a scalar value to rank and optimize user simulation (*ranking and optimization criteria*);
- evaluate user simulation independently from the task and the SDS (*task independence*);
- automatically compute an assessment measure from objective information (*automatic computation*).

Ideally, a good metric should also correlate well with human evaluation metrics but this is hard to predict. Of course, some of these criteria are more important than others and some weighting should be taken into account when designing a metric. Yet, 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.

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

In this section, the state of the art in user simulation evaluation is provided, reflecting the most frequently used evaluation methods in the literature of the last decade. There are many ways to cluster these methods. In Zukerman and Albrecht (2001) and Schatzmann *et al.* (2006), the authors distinguish two categories[2]: direct methods that assess the user simulation by testing the quality of its predictions (e.g. precision and recall) and indirect methods that evaluate the performance of strategies learned from the different models (e.g. utility). We will take a different approach, splitting methods into local methods, which measure turn-level statistics (e.g. frequency of dialogue act types), and global methods, which measure dialogue-level statistics (e.g. task completion, perplexity).

3.1 Turn-level metrics

A large number of early metrics are based on turn-level context and measure local consistency of generated data and data from real users (Schatzmann *et al.*, 2005a). Some turn-level metrics are useful to analyze the dialogues in terms of dialogue style or user initiative and cooperativeness. They can take the form of distributions or of a set of scalar measures. They all share one major drawback of failing to measure the consistency of *sequences* of dialogue acts. *F*-measure and the Kullback–Leibler (KL) divergence provide a single scalar measure; however, they cannot be used to assess the generalization capabilities of a model (see Sections 3.1.3 and 3.1.2). We will discuss each of these turn-level metrics in detail below.

3.1.1 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 (Pietquin, 2004; Schatzmann *et al.*, 2005a). 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.

Schatzmann *et al.* (2005a) 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. misunderstandings;
- the proportion of slot values provided when requested, which is a measure of the user cooperativeness.

[2] Notice that Zukerman and Albrecht (2001) is more about user modelling than user simulation but the distinction is similar to Schatzmann *et al.* (2006).

https://doi.org/10.1017/S0269888912000343

When comparing the above-mentioned metrics with respect to the desired features listed in Section 2, one can see that these metrics allow for comparison of similarities with actual data but have several shortcomings. They do not provide a single scalar measure for ranking models and it is difficult to use them to predict performance of an SDS when used with real users. Finally, generalization is also difficult to assess.

3.1.2 Precision, Recall, (Expected) Accuracy

Precision and Recall are common measures in machine learning and information retrieval and 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) (Zukerman & Albrecht, 2001). Similar metrics, adapted from user modelling literature (Schatzmann *et al.*, 2005a), are widely used in user simulation and even outside the realm of SDSs. Precision and Recall are here 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. However, the classical balanced F-measure (van Rijsbergen, 1979) can be used to combine both these measures and obtain a single scalar:

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

Other related metrics are Accuracy and Expected Accuracy as first introduced in Zukerman and Albrecht (2001) and adapted by Georgila *et al.* (2006):

- 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 the major drawbacks of these metrics is that they do not measure the generalization capabilities of the user simulation. In fact, these metrics actually penalize attempts to generalize since when the model generates unseen dialogues, their scores are lower.

3.1.3 Kullback–Leibler divergence and dissimilarity

In Section 3.1.1, 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 are available but a common choice is the KL divergence (Kullback & Leiber, 1951). The KL divergence between two distributions P and Q is defined by

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

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 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 generalization capabilities of a user model with this metric since it penalizes dialogue strategies, which are different from the real data. In Section 3.3, we discuss new applications of the KL divergence that seek to resolve these last two issues.

3.2 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 2.

3.2.1 Task completion

A task-driven dialogue system assists a user in achieving a goal that is usually not known by the system before the interaction starts. The degree of achievement of this goal is referred to as *task completion*. Commonly used to measure inter-annotation agreement, the κ coefficient can also be used to measure task completion. The κ coefficient (Carletta, 1996) is obtained from a confusion matrix M summarizing 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 $\kappa = 1$ when the system performs perfect interpretation ($P(A) = 1$) and $\kappa = 0$ when the only correct interpretations were obtained by chance ($P(A) = P(E)$). Other task completion measures can be defined such as in Pietquin (2004).

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 Scheffler and Young (2001), Schatzmann *et al.* (2005a) and Pietquin (2004), the task completion or the task completion rate (ratio of successful dialogues) is used to compare dialogues. In Schatzmann *et al.* (2005a), 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.

Once again, capturing task completion involves 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.

3.2.2 Perplexity and log-likelihood

Perplexity is a measure that comes from information theory and was first proposed as a user simulation metric in (Georgila *et al.*, 2006). 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:

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

https://doi.org/10.1017/S0269888912000343

where $p_m(x_i)$ is the probability of x_i given the model, and x_i is a sample from a test data set 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 $\{a_0, a_1, \ldots, a_n\}$ so $p_m(x) = p_m(a_0, a_1, \ldots, a_n)$ is the probability of a sequence of dialogue acts given the user model.

A similar metric is the log-likelihood $\mathcal{L}(x)$ of a data set $x = \{x_i\}_{i=1,\ldots,N}$ 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 measure how well a model is able to predict data in a held-out test set and therefore can be used to measure generalization. Perplexity and log-likelihood are scalar numbers that can be used to rank user simulations, however, the perplexity figure can be difficult to interpret as it ranges from 1 to infinity. These metrics can be viewed as global or dialogue-level metrics as they are based on the probability of sequences of dialogue acts. However, it is somewhat difficult to assign relative importance of dialogue-level features such as dialogue length and task completion.

3.2.3 Hidden Markov Model 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 Cuayahuitl *et al.* (2005), the authors propose to train a HMM on a corpus of real human–machine dialogues (the hidden state is the dialogue state, whereas 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 the 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 Cuayahuitl *et al.* (2005), the authors choose to use the dissimilarity introduced in Section 3.1.3 based on the KL 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. One can, therefore, assume that the measure captures more than what is in the data and that unseen situations can be taken into account (though this has not been experimentally demonstrated by Cuayahuitl *et al.*, (2005)). This metric could be used regardless of the user model, as the artificially generated data can be produced by any model and not only by HMM-based models. It provides a single scalar, however, it does not directly provide any information about the quality of the interaction between real users and a new dialogue system.

3.2.4 Cramér-von Mises divergence

In many applications, a user model for simulation can be used as a predictor of the performance of an SDS. In Williams (2008), 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 $S_0 = x^0_{(1)}, x^0_{(2)}, \ldots, x^0_{(N_0)}$. Similarly, a user simulation U_1 will yield a set of scores $S_1 = x^1_{(1)}, x^1_{(2)}, \ldots, x^1_{(N_1)}$'.
4. 'A user simulation U_1 may be evaluated by computing a real-valued divergence $D(S_0||S_1)$'.

https://doi.org/10.1017/S0269888912000343

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 KL divergence described in Section 3.1.3 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 normalized Cramér-von Mises divergence (Cramer, 1928; Anderson, 1962) is less demanding in this respect because it is based on the *empirical distribution function (EDF)* that does not make any assumptions 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 EDF of the data $S_j = (x^j_{(1)}, x^j_{(N_j)})$ and α is a normalizing 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 2, as 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 generalization capabilities of the model are assessed but that 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.

3.2.5 Bilingual Evaluation Understudy and Discourse-Bilingual Evaluation Understudy

The BLEU (Bilingual Evaluation Understudy) score (Papineni *et al*., 2002) 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. Papineni *et al.* (2002) 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 (BP). It is 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 BP is defined as

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

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

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

Using the BP, BLEU penalizes abnormally short utterances because short-generated utterances have higher N-gram precision.

Jung *et al.* (2009) 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 realizations 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, Jung *et al.* (2009) 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 BP but the N-grams considered here are not sequences of words but sequences of intentions (both user and system 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 score is known to be highly correlated with human judgement Papineni *et al.* (2002), Doddington (2002) and Jung *et al.* (2009) argue that D-BLEU also follows the same tendencies as human judgement. In some cases, BLEU has been reported to fail to predict translation improvements and naturalness. Too few studies are reported about the D-BLEU score for dialogue to allow one to draw the same conclusions, however, the BLEU and D-BLEU are quite similar in their definitions. Once again, this metric also fails to measure the generalization capabilities of the user simulation.

3.2.6 Simulated User Pragmatic Error Rate

SUPER (Simulated User Pragmatic Error Rate; Rieser & Lemon, 2006a; Rieser, 2008) 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* (defined below). 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 generalize 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 $(|P_0(action) - P_1(action)| < \epsilon)$: $V = (+1)$
Tolerated variation:
if $(\epsilon < |P_0(action) - P_1(action)| < \delta)$: $V = (0)$
Penalized variation:
if $(\delta \leqslant |P_0(action) - P_1(action)|)$: $V = (-|P_0(action) - P_1(action)|)$

where P_0 is the unigram probability of observing an action in the real data and P_1 is the probability of a user simulation predicting a action.

https://doi.org/10.1017/S0269888912000343

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 measurement used to assess speech recognition systems.

The SUPER score addresses many of the desired features described in Section 2, however, it is not a direct measure of the ability of the user model to predict the performances of an SDS when used with real users.

3.2.7 Human evaluation

Ai and Litman (2008) 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 method 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 Ai and Litman (2008) is based on 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 supervized 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 PARADISE (Walker *et al.*, 1997b), which evaluates SDS strategies by predicting user satisfaction from real interactions.

The major drawback of the human evaluation method is that it requires human judges to score the dialogues. It is very 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 generalize to other domains.

3.2.8 Absolute performance of learnt strategy

User simulations are frequently used for training optimization algorithms for SDS management strategies, such as RL (Sutton & Barto, 1998). Ai and Litman (2009) propose to measure the performance of the SDS when trained against different user models. The performance is measured as an expected cumulative reward (Williams & Young, 2007) when applying a learnt strategy. The performance of the SDS with real users 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 discussed in Section 3.1.1 and raises the same issues.

3.2.9 Strategy evaluation on real dialogue data

Schatzmann *et al.* (2005b) discuss the influence of the user model on the system dialogue strategy learnt by means of RL (Sutton & Barto, 1998). Although most of the assessment methods are based on user simulation, the paper proposes one method based on direct comparison with real data. The principle of the method is rather simple. Let us suppose that a database of real

https://doi.org/10.1017/S0269888912000343

dialogues $\{d_i\}$ is available, a user model is trained on this data set. This user model is then used to train an RL algorithm that searches for the optimal dialogue strategy π^* to interact with this model. After learning, the computed strategy π^* is compared with the strategy π_{d_i} followed in each of the dialogues d_i in the database. To do so, for each dialogue d_i a similarity measure $Sim\{\pi^*, \pi_{d_i}\}$ is computed. A quality measure R_i is also computed for each dialogue d_i. This quality measure can be the same as the reward function used for training the reinforcement algorithm for instance. The basic idea is thus that if the user model represents the real user population well, then there should be a positive correlation between $Sim\{\pi^*, \pi_{d_i}\}$ and R_i. Indeed, if the strategy followed by the system during dialogue d_i is close to π^* then R_i should be high. If not, it means that the learnt strategy is not optimal for real users and so that the user model that served for learning the strategy π^* did not behave similarly to the actual users. The actual metric that can be extracted is thus the correlation coefficient between $Sim\{\pi^*, \pi_{d_i}\}$ and R_i.

This metric is very interesting since it directly uses existing data rather than needing new data collection or human evaluation. Besides the need of training an RL algorithm for each user model, the main drawback of this method is the need of a similarity measure between strategies. Indeed, defining such a measure could probably be a topic of research by itself. Depending on the definition of the *Sim* measures, the proposed metrics may exhibit interesting features such as the consistency of sequences of dialogue acts for instance.

3.3 N-gram Kullback–Leibler divergence

As discussed in Section 3.1.3, 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 3.2.3, the KL 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 (Cuayahuitl *et al.*, 2005; Cuayahuitl, 2009).

In a similar way to the HMM approach discussed in Section 3.2.3, the N-gram KL divergence (Janarthanam & Lemon, 2009b) is computed between the distributions captured by advanced N-grams trained on a human–machine interaction corpus. Notice that Georgila *et al.* (2006) already introduced the combination of N-grams with other metrics such as prediction and recall. These combined metrics have properties similar to the SUPER one. The advanced N-gram model is a realistic model of each corpus since it takes into account context variables and is deemed sufficiently smoothed by the author to support variability in the generated sentences. This way, unseen dialogues are not penalized if they are ‘too far’ from the distribution captured by the advanced N-gram thus solving the *generalization* problem. It could be argued that this application of the KL divergence takes it from a turn-level metric to a dialogue-level one.

4 Future directions

In this section, we propose a new metric for ranking and optimization of user simulations using IRL. RL (Sutton & Barto, 1998) is now a state-of-the-art method for optimizing dialogue management systems (Levin *et al.*, 1997; Singh *et al.*, 1999; Scheffler and Young, 2001; Pietquin, 2004; Williams *et al.*, 2005; Frampton and Lemon, 2010). It is based on the Markov Decision Processes 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 a *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 perform. The reward function is often defined as the expected user satisfaction, which has to be maximized (Singh *et al.*, 1999). RL 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 RL problem has a dichotomy: the IRL problem (Russell, 1998). By observing an expert agent (mentor) performing optimally, IRL aims at discovering the reward function serving as the optimization 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 optimizing an artificial agent (Ng & Russell, 2000).

The idea of using IRL for dialogue management optimization has been proposed in Paek and Pieraccini (2008). In their paper, the authors propose to use IRL on data collected from human–human interactions to learn the policy followed by the human operator. Similarly, Rieser and Lemon (2008) and Janarthanam and Lemon (2009d) use Wizard-of-Oz techniques to gather data and train a policy that mimics the human wizard. The assumption here is that the human operator or wizard's behaviour is optimal; however, several criticisms can be made about this approach. First, 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. Second, even if the human operator can freely interact with the users, optimality from the user satisfaction point of view is not guaranteed. Third, when interacting with a human, users adopt different behaviours than when interacting with machines (Walker *et al.*, 1997a).

In this paper, 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 optimize 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 (Chandramohan *et al.*, 2011).

There are several advantages to this approach. First, IRL algorithms can be trained on human–machine interaction data, which are easier to automatically annotate than human–human interaction data. Second, 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 optimization criterion to train a user simulation that is independent from the dialogue management system. Indeed, the reward function optimized by the user is related to their satisfaction 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.

A metric obtained from IRL addresses many of the features listed in Section 2. It would provide a single scalar value that is automatically computed and can serve to rank and optimize user models. It could be used to predict performances of real users when interacting with an SDS since this metric is related to user satisfaction. It would not be 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 could be automatically obtained regardless of the task since nothing is task dependent in the IRL paradigm.

Several issues have still to be solved. First, there is inter-user variability that makes the notion of satisfaction user dependent. Thus, it would be 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.

https://doi.org/10.1017/S0269888912000343

Table 1 A comparison of metrics presented in this paper with respect to the list of desired features presented in Section 2

Metric	Consistency	Quality of learnt strategy	Performance prediction (In-direct/ Direct)	Generalisation	Ranking and optimization criteria	Consistent sequences	Task independence	Automatic computation
				Turn level				
DA statistics	Yes	No	No (D)	No	No	No	No	Yes
Precision Recall	Yes	No	No (D)	No	Yes (*F*-score)	No	Yes	Yes
KL	Yes	No	No (D)	No	Yes (but un-bounded)	No	Yes	Yes
				Dialogue level				
Task completion	Yes	No	Yes (I)	No	Yes (but not in isolation)	No	No	Obj (Yes) Subj (No)
Perplexity	Yes	No	No (D)	No	Yes	Yes	Yes	Yes
HMMs	Yes	No	No (D)	Yes	Yes	Yes	Yes	Yes
Cramér von Mises	Yes	No	Yes (I)	Yes	Yes	No	Yes	Yes
BLEU, D-BLEU	Yes	No	No (D)	No	Yes	Yes	Yes	Yes
SUPER	Yes	No	No (D)	Yes	Yes	Yes	Yes	Yes
Human evaluation	Yes	No	Yes (I)	No	Yes	No	No	No
Absolute performance of learnt strategy	Yes–depends on strategy performance metrics	Yes	Yes (I)	Yes	Yes	No	No	Yes—but require real user tests
Strategy evaluation on real data	Yes–depends on *Sim* measure	Yes	Yes (I)	No	Yes	No	No	Yes
N-gram KL divergence	Yes	No	No (D)	No	Yes (but un-bounded)	Yes	Yes	Yes

DAs = dialogue acts; KL = Kullback–Leibler; HMMs = Hidden Markov Models; BLEU = Bilingual Evaluation Understudy; D-BLEU = Discourse-BLEU; SUPER = Simulated User Pragmatic Error Rate.
(D) is a Direct and (I) is a Indirect measure.

5 Discussion and conclusions

Table 1 shows a comparison of metrics presented in this paper with respect to the list of desired features presented in Section 2. In this paper, we have described metrics in terms of turn level and dialogue level. An alternative approach is to categorize metrics into direct or indirect methods of measuring quality of user simulation: direct methods assess the user simulation by testing the quality of its predictions and indirect methods attempt to measure the quality of a user model by evaluating the effect of the model on the dialogue system performance. Direct metrics include Perplexity, HMM similarity, D-BLEU, SUPER and (N-gram) KL. Indirect metrics include Task Completion, Cramér-von Mises divergence, human evaluation, quality of learnt strategy and strategy evaluation on real data.

As illustrated in Table 1, not one of the existing metrics possesses all the desired features; however, 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 generalize to unseen situations. The SUPER score is able to measure generalization capabilities, although it cannot be used to predict performance of a dialogue system as such. The N-gram KL metric has all the advantages of the KL metric but is also able to capture generalization. Only D-BLEU really focuses on the naturalness of generated dialogue but it also fails in predicting performance of an SDS when interacting with real users. It also shares the disadvantages of the BLEU metric, which is widely used in machine translation. All the described methods provide metrics that can be automatically computed from the log files. Finally, a scalar value is provided by all the metrics, however, some are easier to use for ranking than others as they provide a bounded variable, for example, Cramér-von Mises divergence, which is between 0 and 1. The newly proposed IRL metric fulfills all of the above-mentioned desired features with one exception that it does not explicitly measure consistency of the sequences of dialogue acts.

In summary, this paper contains several contributions. First, a list of desired features for user simulation evaluation metrics is provided in Section 2. This list serves as comparison criteria for state-of-the-art metrics that can be found in the literature and summarised in Table 1. A comprehensive list of state-of-the-art metrics for assessing user simulation are listed in Section 3. 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, advantages and disadvantages are listed and it is evident that no one single metric fulfills all of these desired features. Finally in Section 4, we present a promising, new metric based on IRL, which comes close to fulfilling all the desired features of a user simulation quality metric.

Acknowledgements

This research has received funding from the European Community's Seventh Framework Programme (FP72007-2013) under grant agreement no. 216594 (CLASSIC project: www. classic-project.org). The authors thank the referees, as well as Oliver Lemon and Paul Crook for their help and comments.

References

Ai, H. & Litman, D. 2008. Assessing dialog system user simulation evaluation measures using human judges. In *Proceedings of the 46th Meeting of the Association for Computational Linguistics*, Columbus, OH, USA, 622–629.

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

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

https://doi.org/10.1017/S0269888912000343

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

Chandramohan, S., Geist, M., Lefèvre, F. & Pietquin, O. 2011. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Proceedings of Interspeech 2011*, Florence, Italy.

Cramer, H. 1928. On the composition of elementary errors. Second paper: statistical applications. *Skandinavisk Aktuarietidskrift* **11**, 171–180.

Cuayahuitl, H., Renals, S., Lemon, O. & Shimodaira, H. 2005. Human–computer dialogue simulation using hidden Markov models. In *Proceedings of ASRU*, 290–295. Cancun, Mexico

Cuayahuitl, H. 2009. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. PhD thesis, University of Edinburgh, UK.

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

Eckert, W., Levin, E. & Pieraccini, R. 1997. User modeling for spoken dialogue system evaluation. In *Proceedings of ASRU'97*. Santa Barbara, USA.

Frampton, M. & Lemon, O. 2010. Recent research advances in reinforcement learning in spoken dialogue systems. *The Knowledge Engineering Review* **24**(4), 375–408.

Georgila, K., Henderson, J. & Lemon, O. 2005. Learning user simulations for information state update dialogue systems. In *Proceedings of Interspeech 2005*. Lisboa, Portugal.

Georgila, K., Henderson, J. & Lemon, O. 2006. User simulation for spoken dialogue systems: learning and evaluation. In *Proceedings of Interspeech'06*. Pittsburg, USA.

Janarthanam, S. & Lemon, O. 2009a. A data-driven method for adaptive referring expression generation in automated dialogue systems: maximising expected utility. In *Proceedings of PRE-COGSCI 09*. Boston, USA.

Janarthanam, S. & Lemon, O. 2009b. A two-tier user simulation model for reinforcement learning of adaptive referring expression generation policies. In *Proceedings of SIGDIAL*. London, UK.

Janarthanam, S. & Lemon, O. 2009c. Learning adaptive referring expression generation policies for spoken dialogue systems using reinforcement learning. In *Proceedings of SEMDIAL*. Stockholm, Sweden.

Janarthanam, S. & Lemon, O. 2009d. A Wizard-of-Oz environment to study referring expression generation in a situated spoken dialogue task. In *Proceedings of ENLG, 2009*. Athens, Greece.

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

Kullback, S. & Leiber, R. 1951. On information and sufficiency. *Annals of Mathematical Statistics* **22**, 79–86.

Levin, E., Pieraccini, R. & Eckert, W. 1997. Learning dialogue strategies within the Markov decision process framework. In *Proceedings of ASRU'97*. Santa Barbara, USA.

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

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

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

Paek, T. & Pieraccini, R. 2008. Automating spoken dialogue management design using machine learning: an industry perspective. *Speech Communication* **50**, 716–729.

Papineni, K., Roukos, S., Ward, T. & Zhu, W. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318.

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

Pietquin, O., Rossignol, S. & Ianotto, M. 2009. Training Bayesian networks for realistic man–machine spoken dialogue simulation. In *Proceedings of the 1st International Workshop on Spoken Dialogue Systems Technology*, Irsee, Germany, 4.

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

Pietquin, O. 2006. Consistent goal-directed user model for realisitc man–machine task-oriented spoken dialogue simulation. In *Proceedingsof ICME'06*. Toronto, Canada.

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

Rieser, V. & Lemon, O. 2006. Simulations for learning dialogue strategies. In *Proceedings of Interspeech 2006*, Pittsburg, USA.

Rieser, V. & Lemon, O. 2008. Learning effective multimodal dialogue strategies from Wizard-of-Oz data: bootstrapping and evaluation. In *Proceedings of ACL, 2008*. Colombus, Ohio.

Russell, S. 1998. Learning agents for uncertain environments (extended abstract). In *COLT' 98: Proceedings of the 11th Annual Conference on Computational Learning Theory*. ACM, 101–103. Madisson, USA.

Schatzmann, J., Georgila, K. & Young, S. 2005a. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proceedings of SIGdial'05*. Lisbon, Portugal.

Schatzmann, J., Stuttle, M. N., Weilhammer, K. & Young, S. 2005b. Effects of the user model on simulation-based learning of dialogue strategies. In *Proceedings of ASRU'05*. Cancun, Mexico.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H. & Young, S. 2007a. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proceedings of ICASSP'07*. Honolulu, USA.

Schatzmann, J., Thomson, B. & Young, S. 2007b. Statistical user simulation with a hidden agenda. In *Proceedings of SigDial'07*. Anvers, Belgium.

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

Scheffler, K. & Young, S. 2001. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proceedings of NAACL Workshop on Adaptation in Dialogue Systems*. Pittsburgh, PA, USA.

Singh, S., Kearns, M., Litman, D. & Walker, M. 1999. Reinforcement learning for spoken dialogue systems. In *Proceedings of the NIPS'99*. Vancouver, Canada.

Sutton, R. & Barto, A. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

van Rijsbergen, C. J. 1979. *Information Retrieval*, second edn. Butterworths.

Walker, M., Hindle, D., Fromer, J., Fabbrizio, G. D. & Mestel, C. 1997a. 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.

Walker, M., Litman, D., Kamm, C. & Abella, A. 1997b. Paradise: a framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 271–280. Madrid, Spain.

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

Williams, J., Poupart, P. & Young, S. 2005. Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management. In *Proceedings of the SigDial Workshop (SigDial'06)*. Sydney, Australia.

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

Zukerman, I. & Albrecht, D. 2001. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction* **11**, 5–18.