Similarity Scoring for Dialogue Behaviour Comparison

Stefan Ultes and **Wolfgang Maier**
Mercedes-Benz Research & Development
Sindelfingen, Germany
{stefan.ultes,wolfgang.mw.maier}@daimler.com

Abstract

The differences in decision making between behavioural models of voice interfaces are hard to capture using existing measures for the absolute performance of such models. For instance, two models may have a similar task success rate, but very different ways of getting there. In this paper, we propose a general methodology to compute the similarity of two dialogue behaviour models and investigate different ways of computing scores on both the semantic and the textual level. Complementing absolute measures of performance, we test our scores on three different tasks and show the practical usability of the measures.

1 Introduction and Related Work

Interacting with technical systems through voice is prevalent in our every day lives and in the focus of industry and research alike. For evaluating the behaviour of voice interfaces, interaction-based or corpus-based methods have been employed, both aiming at producing absolute measures like dialogue success. While this is clearly an important aspect of dialogue behaviour evaluation, it remains superficial and does not touch on the actual difference of two behaviour models.

The goal of this paper is to propose a method to quantify the similarity of two behaviour models—the learned or hand-crafted dialogue system decision—by means of a similarity score. The core idea is to use well-defined dialogue contexts—moments within a dialogue where the system needs to make a decision of how to respond—and compare the resulting system response of each behaviour model. We propose different similarity measures and demonstrate their usefulness in different scenarios.

Being able to compare behaviour models on a deeper level opens the door to a deeper understanding of the learned behaviour. It aims to answer questions like:

1. When does the behaviour, i.e., the resulting response in a given context, of a reinforcement learning behaviour model converge?

2. Which effect do modifications of the learning parameters or learning set-up have, e.g, different random seeds (minor) or reward models (significant), on the resulting learned behaviour models? Do these modified behaviour models still result in exhibiting the same behaviour? What difference in behaviour causes the differences in absolute measures? Are there sub-sets of dialogue contexts that are fundamental for these differences?

3. How different are single responses of different behaviour models for the same given dialogue context?

These questions are of high relevance in cases where not only the average absolute performance is of interest but also the actual learned behaviour. On an application level, the answers to those question can help to decide which behaviour model to apply for a concrete live application, as they can support decision such as when to stop learning, or reveal the properties of different random seeds. From a more scientific point of view, the questions contribute to the overall problem of what we can learn about the interaction characteristics from the learned models.

The core task of a voice interface, also called spoken dialogue systems (SDS), is the decision of how to respond to a given user input and a dialogue context. This task is either modelled explicitly or implicitly. An explicit behaviour model usually comprises a distinct dialogue system module called dialogue policy taking in a dialogue state—a combined and dense representation of the current user

Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 37 - 48
July 1-3, 2020 ©2020 Association for Computational Linguistics

input interpretation and the dialogue context—and producing an abstract system response. In a subsequent step, this abstract system response is then transferred into text by a natural language generator. An implicit behaviour model uses a neural network to learn a text response directly based on text input thus combining user input interpretation, dialogue context integration, and dialogue response selection in one model.

Absolute measures to evaluate the performance of these behaviour models through the interaction with real or simulated users are, for example, task success or dialogue length (Gašić and Young, 2014; Lemon and Pietquin, 2007; Daubigney et al., 2012; Levin and Pieraccini, 1997; Young et al., 2013; Su et al., 2016; Ultes et al., 2015; Wen et al., 2017). Other measures are user satisfaction (Walker et al., 1997; Chu-Carroll and Nickerson, 2000; Dzikovska et al., 2011; Ultes et al., 2015; Wen et al., 2016; Ultes et al., 2017a) or quality of interaction (Möller et al., 2008; Schmitt and Ultes, 2015). All are often acquired through interaction-based studies[1].

Others have employed corpus-based evaluation by comparing textual system responses with transcriptions of actual interaction as absolute evaluation criterion where the response in the corpus is treated as ground truth (Serban et al., 2016; Sordoni et al., 2015; Li et al., 2016a; Lowe et al., 2015). Text comparison metrics like BLEU (Papineni et al., 2002) have been adopted from machine translation to evaluate how well the system response matches the one in the database, e.g., (Li et al., 2016b; Sordoni et al., 2015). This way of evaluating has been criticised widely as a system response that is different from the one in the data base can still be a valid system response simply leading to a different subsequent dialogue. Furthermore, the BLEU score evaluation hardly correlates with human judgements (Liu et al., 2016; Novikova et al., 2017).

Dismissing text similarity measures as not useful for dialogue evaluation, however, is overhasty and shortsighted. While those measures may not help with absolute comparison of policies, they may be valuable to compare two policies with each other. In other words, they can help to reveal the similarity between two models without explicitly judging their performance.

In this work, we propose a framework to compute the similarity of two dialogue behaviour models. This comprises the following contributions:

- a set-up to compare behaviour models on the level of single decisions
- similarity scores to compare behaviour models on the level of single decisions
- applications of similarity scoring offering a deeper understanding of the learned behaviour

The remainder of this paper is structured as follows: In Section 2, we introduce the general approach for quantifying the similarity of behaviour models. We then investigate the usability of several different ways of computing a similarity score in Sec. 3, considering scores on the semantic and the textual level. In Sec. 4 and 5, we describe our experimental setup, test our scores on three different tasks, and show their correlation confirming their practical usability.

2 Scoring Framework

To compare two dialogue behaviour models, this paper explores the usage of similarity measures instead of relying on absolute performance measures. The main idea is—in addition to knowing about the absolute performance—to learn about how similar or different two behaviour models are. For this, a defined set of dialogue contexts is applied to each behaviour model to generate corresponding system responses. These responses are then compared to learn about the overall similarity of the models. The general approach illustrated in Figure 1 is as simple as effective:

1. Define a set of dialogue contexts C.
2. Evaluate each behaviour model m in a deterministic way and collect the resulting system responses a_c^m for each context $c \in C$.
3. Calculate similarity scores $\sigma(a_c^m, a_c^{m'})$ for each system response pair $(a_c^m, a_c^{m'})$, e.g., by using one of the measures proposed in Section 3.

Aside from finding suitable similarity measures, one of the key challenges is to find good dialogue contexts that may be used as basis for comparison. Here, a dialogue context is a sub-dialogue either represented by a set of system utterances and user utterances (which is necessary, e.g., for end-to-end dialogue generators) or directly by the dense

[1]For a good overview over absolute metrics including a taxonomy, please refer to Hastie (2012).

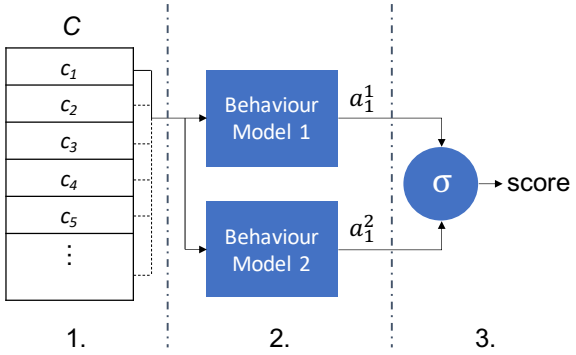

Figure 1: The three steps of the scoring framework.

representation of a dialogue state following the Markovian idea (often only available in modular dialogue systems). The proposed framework relies on a well-defined set of dialogue contexts, avoiding the evaluation of unrealistic situations which would directly influence the similarity scores.

In this work, the focus lies on modular dialogue systems where dialogue states are available to represent the dialogue context. Thus, there are two natural options of finding a set of dialogue contexts: collecting dialogues with corresponding dialogue states from actual real dialogues noted as C_{col} or generating a set of dialogue states noted as C_{gen}.

3 Similarity Scores

A similarity score is computed for the comparison of two behavioural models π and π'. Depending on the nature of the behavioural model, for each context $c_i \in C$, each may produce an abstract system response actions a_i, and an text response p_i. Each abstract system action $a_i = act_i(s_1^i = v_1^i, \ldots, s_j^i = v_j^i)$ consists of a dialogue act act_i, representing the communicative function like *inform* or *request*, and a set S_i of j slot-value-pairs $S_i = \{(s_1^i, v_1^i), \ldots, (s_j^i, v_j^i)\}$ representing the concepts and their respective values[2]. To compute each similarity score, $|C|$ action/text response pairs are compared using the following similarity score measures.

Total Match Rate The total match rate (TMR) is based on a binary score that regards two actions a, a' as equal only if they completely match, i.e., $\delta_{a,a'} = 1$ iff $a = a'$, else 0. The TMR is then defined by

$$TMR = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{a_i,a_i'} \,. \tag{1}$$

Dialogue Act Match Rate The dialogue act match rate (DMR) is based on a binary score comparing the actions a, a' where both match if the corresponding dialogue acts are the same: $\delta_{act,act'} = 1$ iff $act = act'$, else 0. The DMR is defined by

$$DMR = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{act_i,act_i'} \,. \tag{2}$$

Concept Error Rate The concept error rate is a measure usually used for evaluating natural language understanding systems that translate text input to a semantic representation. The concept error rate then is computed by comparing the resulting semantic representation with a ground truth. Instead of comparing a semantic representation a with a ground truth, it can also be used to compare it to another representation a' produced by a different behaviour model.

Similar to the word error rate, it is based on the Levensthein distance of two dialogue actions having one as hypothesis h and one as reference r:

$$dist(h, r) = \#ins + \#del + \#sub \,. \tag{3}$$

$\#ins$, $\#del$, and $\#sub$ are the number of insertions, deletions and substitutions, respectively, when computing the Levensthein distance of the concepts of the sets S_1 and S_2 where each slot s_j^i and each value v_j^i are treated as individual items.

The concept error CE is then defined by

$$CE(h, r) = \frac{|r| - dist(h, r)}{|r|} \tag{4}$$

normalising the error by the length of r. Clearly, this is an asymmetric quantity. To make it symmetric, it is calculated using a and a' both as hypothesis and reference:

$$\tilde{CE}(a, a') = \frac{CE(a, a') + CE(a', a)}{2} \,. \tag{5}$$

The concept error rate is then calculated with

$$CER = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{act,act'} \cdot \tilde{CE}(a, a') \,. \tag{6}$$

[2]For the abstract system action a = inform(name='Golden House', area=centre), act = inform and S = {(name,'Golden House'),(area,centre)}.

Concept Match Rate As an alternative to the asymmetric CER, we propose the symmetric concept match rate. Instead of basing it on an error comparing a hypothesis with a reference, it counts concepts γ that are present in both dialogue actions where $\tilde{m}(a, a', \gamma)$ defines if a match occurred:

$$\tilde{m}(a, a', \gamma) = \begin{cases} 1, \text{if } \gamma \in S \text{ and } \gamma \in S' \\ 0, \text{otherwise .} \end{cases} \quad (7)$$

The concept match CM takes into account the dialogue acts and the unified set of concepts $\tilde{S} = S_1 \cup S_2$ of both dialogue actions treating slots s and values v in $\tilde{S}$ as individual γ:

$$\tilde{CM}(a, a') = \delta_{act,act'} + \sum_{\in \tilde{S}} \tilde{m}(a, a', \gamma) \quad (8)$$

A concept match of two dialogue actions a and a' is thus defined by

$$CM(a, a') = \frac{\tilde{CM}(a, a')}{1 + |\tilde{S}|} \quad (9)$$

and the concept match rate by

$$CMR = \frac{1}{|C|} \sum_{i=1}^{|C|} CM(a, a') \ . \quad (10)$$

Cosine Similarity and angular similarity The Universal Sentence Encoder (USE) (Cer et al., 2018) is a generic sentence encoder which employs two measures for the computation of the distances between encoded sentences, namely cosine similarity and angular similarity:

$$\textit{cosine-sim} = \mathbf{USE}(p) \cdot \mathbf{USE}(p') \quad (11)$$

$$\textit{angular-sim} = 1 - \frac{\arccos(\textit{cosine-sim})}{\pi} \quad (12)$$

BLEU The BLEU score (Papineni et al., 2002) is a measure used for the evaluation of machine translation systems. It is based on an n-gram precision φ_n, computed as the number of common n-grams between reference p and candidate phrase p' (and vice versa) divided by the number of n-grams of the candidate phrase. The score of a corpus is the geometric mean of modified precision scores multiplied with a brevity penalty υ:

$$BLEU = \upsilon \cdot \exp(\sum_n w_n \log \varphi_n) \ , \quad (13)$$

where υ is 1 if $|p| > |p'|$ and $e^{\frac{1-|p'|}{|p|}}$ otherwise. $BLEU$ is computed for multiple values of $n \leq 4$ and geometrically averaged (called BLUE-4). The final score is made symmetric in accordance with Eq. 5.

Table 1: Absolute results of the simulated experiments for R_{TS} and R_{IQ} after different number of training dialogues showing task success rate (TSR), average interaction quality (AIQ), and average dialogue length (ADL) in number of turns. Each value is computed after 100 evaluation dialogues averaged over three trials.

# Training Dialogues	*TSR*		*AIQ*		*ADL*	
	R_{TS}	R_{IQ}	R_{TS}	R_{IQ}	R_{TS}	R_{IQ}
1,000	0.98	0.99	3.78	3.85	4.46	4.44
5,000	0.99	0.98	3.78	3.81	4.41	4.51
10,000	1.00	0.98	3.81	3.80	4.32	4.47
15,000	1.00	0.99	3.79	3.81	4.36	4.43
20,000	1.00	0.97	3.86	3.73	4.15	4.62
25,000	1.00	0.98	3.77	3.85	4.37	4.30
30,000	1.00	0.96	3.71	3.87	4.49	4.41
35,000	0.99	0.96	3.73	3.84	4.42	4.46
40,000	1.00	0.94	3.77	3.77	4.35	4.75

BERTscore The BERTscore (Zhang* et al., 2020) is an automatic evaluation metric used for text generation that has shown a high correlation with human ratings. Given a function β which returns the BERT embedding (Devlin et al., 2018) for a given token, recall and precision along with the F1-score are computed for a reference p and a candidate p' as

$$R_{BERT} = \frac{1}{|p|} \sum_{p_i \in p} \max_{p'_j \in p'} \beta(p_i)^\top \beta(p'_j) \ , \quad (14)$$

$$P_{BERT} = \frac{1}{|p'|} \sum_{p'_j \in p'} \max_{p_i \in p} \beta(p_i)^\top \beta(p'_j) \ , \quad (15)$$

$$F_{BERT} = 2 \frac{R_{BERT} \cdot P_{BERT}}{R_{BERT} + P_{BERT}} \ . \quad (16)$$

F_{BERT} has been selected as a symmetric similarity score that also represents a reasonable balance between R_{BERT} and P_{BERT}.

Examples scores are shown in Appendix A.

4 Application Scenarios of Similarity Score Evaluation

We present three different scenarios addressing the following questions: When does the behaviour of a reinforcement learning policy converge? Which effect do modifications of the random seeds have on the resulting learned policies? Which effect do modifications of the reward models have on the resulting learned policies?

4.1 Evaluation Setup

To answer these question, we apply the following evaluation setup.

Table 2: Similarity measures for testing convergence of each trial (random seed) for R_{TS} empmloying task success and R_{IQ} employing interaction quality for C_{real}.

	Trial	# Training Dialogues	TMR	DMR	CER	CMR	ang sim	cos sim	BLEU-4	BERTscore
Task Success	0	10,000	0.978	0.978	0.978	0.980	0.863	0.868	0.470	0.905
		20,000	0.989	0.989	0.989	0.989	0.863	0.877	0.481	0.910
		30,000	0.984	0.995	0.991	0.990	0.874	0.888	0.518	0.916
		40,000	1.000	1.000	1.000	1.000	0.871	0.891	0.493	0.918
	1	10,000	0.945	0.978	0.962	0.953	0.855	0.843	0.507	0.907
		20,000	0.962	0.978	0.970	0.966	0.860	0.858	0.529	0.917
		30,000	0.995	0.995	0.995	0.995	0.874	0.885	0.517	0.922
		40,000	0.978	0.989	0.986	0.985	0.875	0.885	0.532	0.925
	2	10,000	0.885	0.940	0.917	0.907	0.837	0.815	0.462	0.893
		20,000	0.995	0.995	0.995	0.995	0.872	0.876	0.519	0.913
		30,000	0.984	0.989	0.988	0.988	0.868	0.880	0.485	0.911
		40,000	0.978	0.984	0.981	0.982	0.876	0.885	0.525	0.918
Interaction Quality	0	10,000	0.944	0.944	0.944	0.950	0.860	0.856	0.484	0.901
		20,000	0.972	0.972	0.972	0.972	0.859	0.863	0.441	0.897
		30,000	0.994	0.994	0.994	0.997	0.859	0.870	0.422	0.897
		40,000	0.978	0.983	0.980	0.984	0.867	0.880	0.461	0.901
	1	10,000	0.972	0.972	0.972	0.972	0.875	0.884	0.551	0.928
		20,000	0.966	0.978	0.974	0.973	0.837	0.827	0.463	0.899
		30,000	0.994	0.994	0.994	0.995	0.845	0.833	0.474	0.903
		40,000	0.994	0.994	0.994	0.996	0.849	0.846	0.491	0.907
	2	10,000	0.961	0.961	0.961	0.965	0.848	0.830	0.498	0.901
		20,000	0.978	0.978	0.978	0.978	0.837	0.820	0.458	0.895
		30,000	0.983	0.983	0.983	0.984	0.848	0.841	0.502	0.903
		40,000	0.989	0.989	0.989	0.990	0.846	0.837	0.495	0.902

4.1.1 Policy Training

For the evaluation, two policies are trained to reflect two different set-ups. One set-up uses the conventional task success as main reward component as heavily used within the literature (Gašić and Young, 2014; Vandyke et al., 2015; Su et al., 2016, e.g.) and the other set-up uses the interaction quality (IQ) (Schmitt and Ultes, 2015) representing user satisfaction as described by Ultes et al. (Ultes et al., 2017a; Ultes, 2019). IQ is defined on a five-point scale from five (satisfied) down to one (extremely unsatisfied). To derive a reward from this value,

$$R_{IQ} = -T + (iq - 1) \cdot 5 \tag{17}$$

is used where R_{IQ} describes the final reward. It is applied to the final turn of the dialogue of length T with a final IQ value of iq. Thus, a per-turn penalty of -1 is added to the dialogue outcome. This results in a reward range of 19 down to $-T$ which is consistent with related work in which binary task success (TS) was used to define the reward as:

$$R_{TS} = -T + \mathbb{1}_{TS} \cdot 20 \ , \tag{18}$$

where $\mathbb{1}_{TS} = 1$ only if the dialogue was successful, $\mathbb{1}_{TS} = 0$ otherwise.

For each set-up, three policies with different random seeds were trained in a simulation environment using the PyDial Statistical Spoken Dialogue System Toolkit (Ultes et al., 2017b) with an agenda-based user simulator (Schatzmann and Young, 2009). For each trial, a GP-SARSA (Gašić and Young, 2014) policy model was trained—a learning algorithm known for its high sample-efficiency—with dialogues in the Cambridge restaurants domain about finding restaurants in Cambridge, UK. The domain comprises three slots used as search constraints (area, price range, food type). For belief state tracking—updating the probability distribution over all possible dialogue states in each turn—the focus belief tracker is used (Henderson et al., 2014). Prompts were generated using the SC-LSTM (Wen et al., 2015) natural language generator implementation of PyDial.

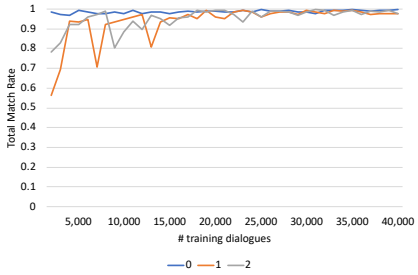

Figure 2: Convergence of each trial (random seed) for R_{TS} evaluated with the total match rate and on C_{col}^{TS}. The curves for R_{IQ} as well as C_{gen} set are similar.

To ensure consistency, the standardised Environment 1 from Casanueva et al. (2017) is used. The interaction quality is estimated using a BiLSTM with self-attention as described by Ultes (2019).

For each trial of the task success and the interaction quality set-ups, a policy was trained with 40,000 dialogues and evaluated after each 1,000 training dialogues with 100 evaluation dialogues. The absolute performance of each set-up in terms of task success rate (TSR), average interaction quality (AIQ) as estimated at the end of each dialogue, and the average dialogue length (ADL) is shown in Table 1 averaged over all three trials.

4.1.2 Collected and Generated Context Sets

For computing the similarity scores described in Section 3, two types of dialogue context sets are used: collected dialogue contexts C_{col} and generated dialogue contexts C_{gen}.

The contexts of C_{col} are collected from the evaluation cycles of the 40,000 training batch of R_{TS} and R_{IQ}. From each trial, 10 evaluation dialogues are taken to constitute C_{col}^{TS} and C_{col}^{IQ}. This results

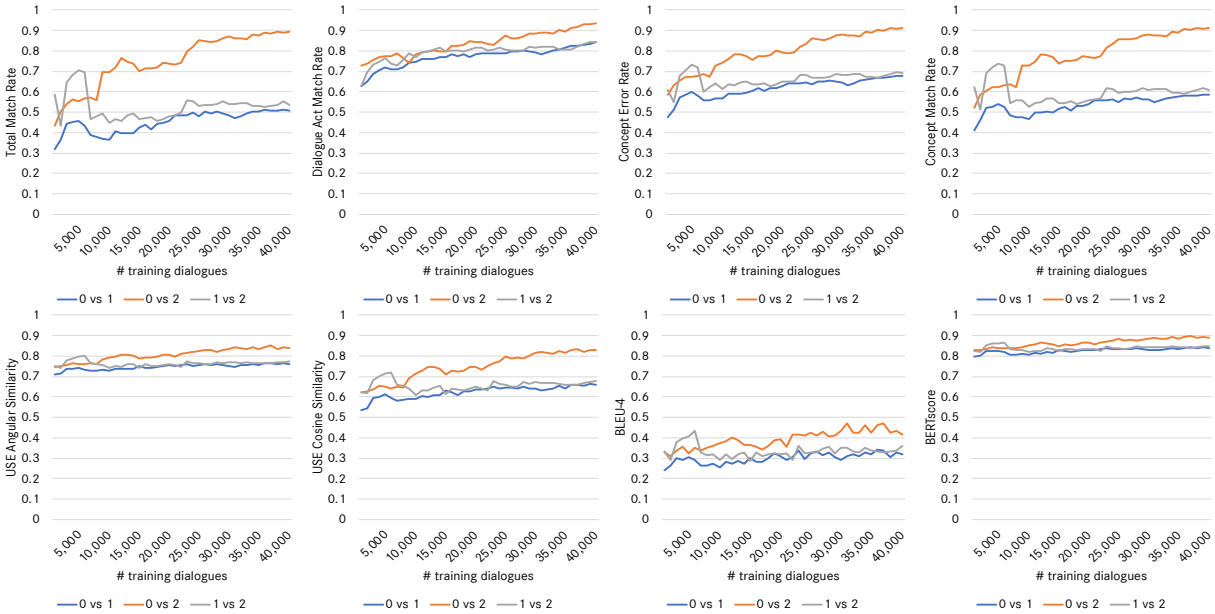

Figure 3: All similarity scores for C_{col}^{TS} for comparing the different trials / random seeds with each other (trial 0 vs. trial 1, trial 0 vs. trial 2, trial 1 vs. trial 2) of the R_{TS} policies evaluated after each training cycle of 1,000 dialogues.

in a total of 30 dialogues each with 183 dialogue contexts in C_{col}^{TS} and 178 collected dialogue contexts in C_{col}^{IQ}.

To generate dialogue contexts C_{gen}, the most relevant parts of a dialogue state are considered. For the Cambridge Restaurants domain, these are the three main search constraints *area*, *pricerange*, *foodtype* as well as the *method* of how to look for information. In the belief state used by PyDial, the joint probability of the dialogue state $P(s)$ is divided based on independence assumptions so that each slot probability is modelled separately. Hence, dialogue contexts are generated with probabilities for each slot in 0.1 steps, e.g., for a value of slot *area*[3], dialogue contexts with a probability of $0.0, 0.1, 0.2, \ldots, 1.0$, respectively, are created. With four slots and taking into account all possible slot and probability combinations, this results in a total of 1,296 generated dialogue contexts C_{gen} used for both R_{TS} and R_{IQ}.

4.2 Experiments and Results

The experimental results of applying above setup are described in the following.

4.2.1 Computing Similarity Scores to Test for Policy Convergence

The first scenario addresses the question if and when each single policy converges in its behaviour. Thus, a similarity score is computed comparing each policy before and after each training iteration, i.e., the additional training of 1,000 dialogues[4]. If the policy converges, the similarity score should be close to 1.0 for all similarity measures. The resulting similarity scores for C_{col}^{TS} and C_{col}^{IQ} for each reward and each trial are shown in Table 2. For convergence testing, the total match rate is used as the main criterion as two behaviour models are the same if they result in the exact same action for each dialogue context. The resulting learning curve for R_{TS} is shown in Figure 2 which is similar to the curve of R_{IQ}. Results for C_{gen} are omitted as they are almost identical to C_{col}. Notably, even though the differences are very small, all policies might still change after 40,000 training dialogues.

4.2.2 Computing Similarity Scores to Test for Seed Convergence

The second scenario addresses the question if and when policies trained with different random seeds.For this, each policy trained with R_{TS} and each policy trained with R_{IQ} are compared with the other policies trained with the same reward at each training iteration. As there are three trials / random seeds for each set-up, this results in three

[3]The actual value to pick is not important due to the way the dialogue state is used by the GP-SARSA algorithm.

[4]A policy after 2,000 training dialogues is compared with the same policy after 1,000 training dialogues, then again the policy after 3,000 training dialogues with the policy after 2,000 training dialogues, and so on.

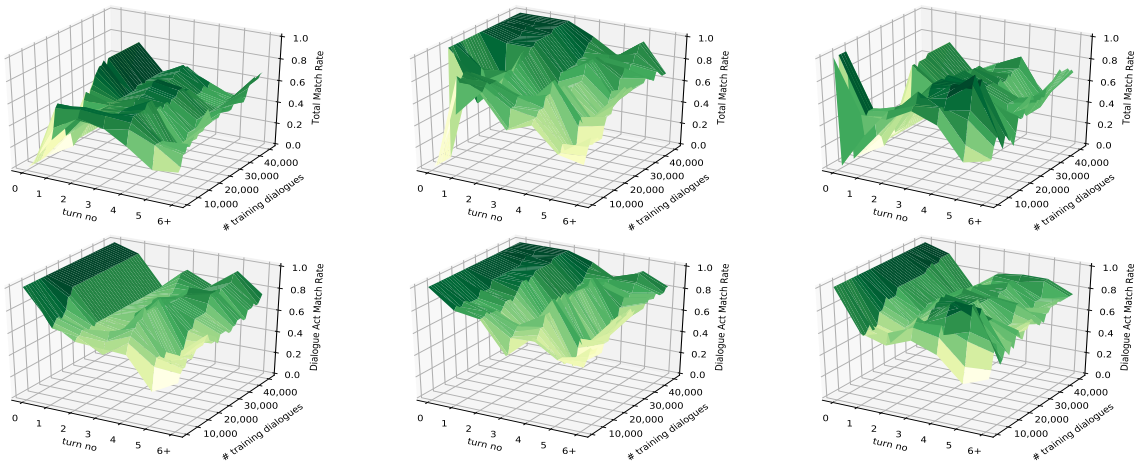

Figure 4: Turn-based results for total match rate (top row) and dialogue act match rate (bottom row) for all three trials using the TS-based reward.

comparisons for both R_{TS} and R_{IQ}[5]. If the trials converge to the same policy, the similarity score is close to 1.0 for all similarity measures. The resulting similarity scores for the respective C_{col}^{TS}/C_{col}^{IQ} and C_{gen} for each reward are shown in Table 4 with a visualisation for R_{TS} on C_{col}^{TS} for all metrics and training iterations in Figure 3.

Evidently, neither the policies of R_{TS} nor the policies of R_{IQ} converge to the same behaviour. Instead, they only reach a maximal TMR of 0.896 for R_{TS} and 0.68 for R_{IQ} for only one pair in each case using C_{col}. Even though the policies do not converge to the identical behaviour, the convergence in terms of DMR is much better and all policy models tend to learn the same basic behaviour—the respective dialogue acts—independent of the random seed that is used.

Comparing the scores of C_{col}^{TS}/C_{col}^{IQ} with C_{gen} shows that for the latter, the scores are much lower but the overall tendencies of the similarity scores are the same. This shows that the basis that is used is important when looking at absolute scores but not relevant when only the tendency is of interest. One explanation for this difference in absolute scores is that C_{col} may contain more dialogue contexts that are very similar to each other where the policies rather agree. C_{gen} contains each dialogue context only once. Additionally, C_{gen} may also contain dialogue contexts that have not been visited during training or evaluation and thus it is harder for the policy model to learn consistent behaviour.

[5]For example, the policy of trial 1 after 3,000 training dialogues is compared with the policy of trial 2 after 3,000 training dialogues, the policy of trial 1 after 4,000 training dialogues is compared with the policy of trial 2 after 4,000 training dialogues, and so on.

Analysing all used similarity scores generally, Figure 3 shows that for all similarity scores expect DMR, the curves are similar in terms of shape but different in terms of scores and differences between trials. Each of the text-based scores angular similarity, BLEU-4 and BERTscore seems to produce values in the same range within one set-up. Thus, the scores are not very suitable for comparison.

For R_{TS} on C_{col}^{TS}, a more detailed analysis has been conducted on the similarities of the dialogue behaviour models with respect to the progression through the dialogue, i.e., what are the similarity scores when only looking at the first turn, the second turn, etc. Figure 4 shows that, for the first system turn, behaviour is learned where both models either always agree or always disagree in terms of TMR but always agree in terms of DMR. Again, the agreement on the communicative function is evident. This is not surprising as in the beginning, the system needs to acquire information from the user with the *request* dialogue act.

4.2.3 Computing Similarity Scores to Compare Policies from Different Reward Models

The final scenario addresses the question of how similar the dialogue behaviour of two models is that are trained with the different rewards R_{TS} and R_{IQ}. As common base, both collected contexts are combined to $C_{col}^{TS+IQ} = C_{col}^{TS} \cup C_{col}^{IQ}$. The results are shown in Table 3 with the TMR and DMR compared to the results of scenario 2 in Figure 5.

The cross-comparison of R_{TS} and R_{IQ} shows that the TMR and DMR are a bit lower than for the comparison of policies within R_{TS} and R_{IQ}, re-

Table 3: All similarity scores for comparing the respective policies trained with R_{TS} and R_{IQ} with each other using C_{gen}^{TS+IQ} after 40,000 training dialogues each.

TS vs IQ	*TMR*	*DMR*	*CER*	*CMR*	*USE (ang)*	*USE (cos)*	*BLEU-4*	*BERTscore*
0 vs 0	0.645	0.825	0.739	0.695	0.783	0.700	0.325	0.845
0 vs 1	0.310	0.789	0.556	0.428	0.725	0.589	0.243	0.808
0 vs 2	0.288	0.756	0.529	0.413	0.723	0.585	0.240	0.801
1 vs 0	0.338	0.867	0.606	0.458	0.735	0.608	0.263	0.826
1 vs 1	0.485	0.831	0.665	0.574	0.778	0.698	0.329	0.843
1 vs 2	0.380	0.798	0.596	0.486	0.757	0.658	0.281	0.825
2 vs 0	0.615	0.803	0.709	0.668	0.776	0.682	0.313	0.840
2 vs 1	0.307	0.759	0.542	0.421	0.722	0.576	0.246	0.803
2 vs 2	0.296	0.748	0.527	0.414	0.722	0.579	0.232	0.799

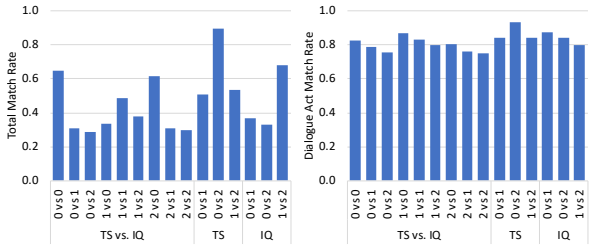

Figure 5: Total match rates and dialogue act match rates for tho cross compare experiments computing the similarity scores for each policy of R_{TS} with each policy of R_{IQ} after 40,000 training dialogues. Along with that, the results of the internal policy similarity scores of R_{TS} and R_{IQ} shown for comparison.

spectively, but generally, the differences are similar. This means that, generally, the differences of policies trained with R_{TS} compared to policies trained with R_{IQ} are not much bigger than just using a different random seed.

5 Correlation of Scores

To analyse how complementary the different scores are, all system behaviour pairs of all experiments have been used to compute correlation and mean squared error for each score pair. The results are shown in Figure 6. An interesting finding is that the CER and CMR have a very high correlation and a very low error. Thus, both seem to capture the same similarities. In contrast to that, the DMR has a very low correlation with other scores and thus does provide additional information. BLEU-4 also does not have a high correlation with other metrics but does also not provide a huge variety, as shown in the example in Figure 3. Comparing semantic-based similarity scores with text-based similarity scores shows that CER and USE-based cosine distance have a quite high correlation and a relatively small mean squared error. Thus, the similarity of two systems that provide semantic output and the similarity of two other systems that only provide text output can be comparably quantified with the CER and the USE-based cosine distance.

The overall total match rate of the samples used for calculating the correlation and mean squared error is 63.3%. Thus, the matches govern the correlation scores. Computing the correlation only on the samples that do not match reveals slightly different numbers that still match the overall impression. The main difference is that the correlation between CER and CMR drops down to 0.466.

6 Conclusion

This work proposes a first step towards a more detailed analysis of dialogue behaviour models by proposing a framework to compute similarity scores. A similarity score is meant to quantify how similar the decisions made by one dialogue behaviour model are compared to a second dialogue behaviour model. Using a fixed set of dialogue contexts, each model is evaluated and the resulting system responses—as semantic representations and/or as text utterances—are captured and used for the similarity score. We proposed eight similarity scores and applied them to three different scenarios.

By doing that, we were able to validate supposed certainties about reinforcement-based policy learning. We could observe that in the used set-ups, all policy models converged towards a fixed behaviour while still showing minor behavioural changes even after a very large number of training iterations.

Modifications of the random seeds, however, already result in a noticeable differences in the converged behaviour in the applied evaluation setup. The quantified differences are even similar in magnitude to a modification of the reward model, i.e., changing a random seed has a similar effect on the learned policy as switching from task success to interaction quality as the principal reward component.

Out of the eight proposed similarity scores, many seem to capture different aspects of similarity, so it remains to the application to decide which score is more useful. Only text-based scores coming from the language translation field like BLEU and BERTscore seem not to be too useful. One reason for this might be the dependency of the absolute score on the prompt length: quantifying textual

Table 4: All similarity measures for comparing the trials (random seeds) with each other for R_{TS} employing task success and for R_{IQ} employing interaction quality as main reward component. Results for the respective C_{col} are on the left and C_{gen} are on the right.

Comparison	# Training Dialogues	TMR	DMR	CER	CMR	ang sim	cos sim	BLEU-4	BERTscore
Task Success	0 vs. 1	10,000	0.366	0.749	0.568	0.467	0.729	0.592	0.254	0.807
		20,000	0.459	0.781	0.627	0.540	0.755	0.638	0.311	0.832
		30,000	0.486	0.792	0.645	0.561	0.752	0.639	0.290	0.828
		40,000	0.508	0.842	0.678	0.585	0.762	0.657	0.317	0.840
	0 vs. 2	10,000	0.694	0.781	0.744	0.727	0.792	0.715	0.373	0.852
		20,000	0.738	0.842	0.794	0.770	0.805	0.746	0.393	0.865
		30,000	0.869	0.885	0.879	0.880	0.836	0.815	0.433	0.887
		40,000	0.896	0.934	0.915	0.914	0.840	0.828	0.416	0.890
	1 vs. 2	10,000	0.448	0.770	0.613	0.525	0.742	0.611	0.291	0.820
		20,000	0.481	0.814	0.651	0.560	0.759	0.651	0.320	0.835
		30,000	0.541	0.814	0.681	0.610	0.771	0.675	0.351	0.843
		40,000	0.536	0.842	0.692	0.609	0.774	0.679	0.361	0.850
Interaction Quality	0 vs. 1	10,000	0.371	0.792	0.581	0.473	0.736	0.599	0.290	0.826
		20,000	0.337	0.781	0.565	0.450	0.720	0.575	0.237	0.803
		30,000	0.354	0.843	0.602	0.471	0.735	0.612	0.260	0.819
		40,000	0.365	0.876	0.618	0.480	0.741	0.619	0.283	0.826
	0 vs. 2	10,000	0.320	0.764	0.548	0.434	0.713	0.556	0.247	0.801
		20,000	0.343	0.860	0.603	0.462	0.732	0.596	0.265	0.813
		30,000	0.337	0.876	0.607	0.468	0.726	0.596	0.242	0.810
		40,000	0.331	0.843	0.589	0.459	0.736	0.613	0.263	0.820
	1 vs. 2	10,000	0.376	0.803	0.585	0.482	0.759	0.666	0.281	0.830
		20,000	0.663	0.826	0.742	0.724	0.790	0.722	0.374	0.854
		30,000	0.663	0.820	0.739	0.717	0.783	0.705	0.360	0.847
		40,000	0.680	0.798	0.738	0.728	0.792	0.723	0.375	0.852

Comparison	# Training Dialogues	TMR	DMR	CER	CMR	ang sim	cos sim	BLEU-4	BERTscore
Task Success	0 vs. 1	10,000	0.199	0.526	0.363	0.317	0.666	0.454	0.139	0.754
		20,000	0.259	0.542	0.400	0.362	0.677	0.475	0.163	0.762
		30,000	0.357	0.586	0.471	0.436	0.700	0.525	0.209	0.783
		40,000	0.398	0.615	0.507	0.465	0.712	0.548	0.233	0.790
	0 vs. 2	10,000	0.660	0.696	0.678	0.734	0.781	0.708	0.343	0.845
		20,000	0.707	0.732	0.720	0.764	0.792	0.726	0.357	0.854
		30,000	0.752	0.775	0.763	0.800	0.804	0.754	0.369	0.865
		40,000	0.823	0.843	0.832	0.854	0.825	0.795	0.406	0.886
	1 vs. 2	10,000	0.258	0.532	0.395	0.383	0.683	0.488	0.171	0.774
		20,000	0.305	0.577	0.441	0.419	0.693	0.510	0.194	0.782
		30,000	0.371	0.580	0.476	0.463	0.705	0.537	0.217	0.791
		40,000	0.405	0.612	0.508	0.476	0.714	0.552	0.231	0.795
Interaction Quality	0 vs. 1	10,000	0.011	0.601	0.306	0.154	0.635	0.404	0.079	0.751
		20,000	0.066	0.710	0.388	0.205	0.649	0.435	0.104	0.759
		30,000	0.157	0.730	0.443	0.282	0.670	0.475	0.142	0.775
		40,000	0.248	0.747	0.497	0.356	0.692	0.520	0.175	0.790
	0 vs. 2	10,000	0.003	0.856	0.430	0.188	0.653	0.458	0.085	0.754
		20,000	0.002	0.863	0.432	0.190	0.653	0.456	0.077	0.750
		30,000	0.117	0.843	0.480	0.285	0.679	0.512	0.129	0.774
		40,000	0.206	0.813	0.510	0.345	0.695	0.541	0.161	0.784
	1 vs. 2	10,000	0.113	0.584	0.345	0.251	0.687	0.530	0.116	0.763
		20,000	0.361	0.645	0.498	0.467	0.717	0.588	0.183	0.780
		30,000	0.413	0.669	0.526	0.500	0.731	0.610	0.232	0.795
		40,000	0.446	0.681	0.545	0.523	0.739	0.625	0.252	0.806

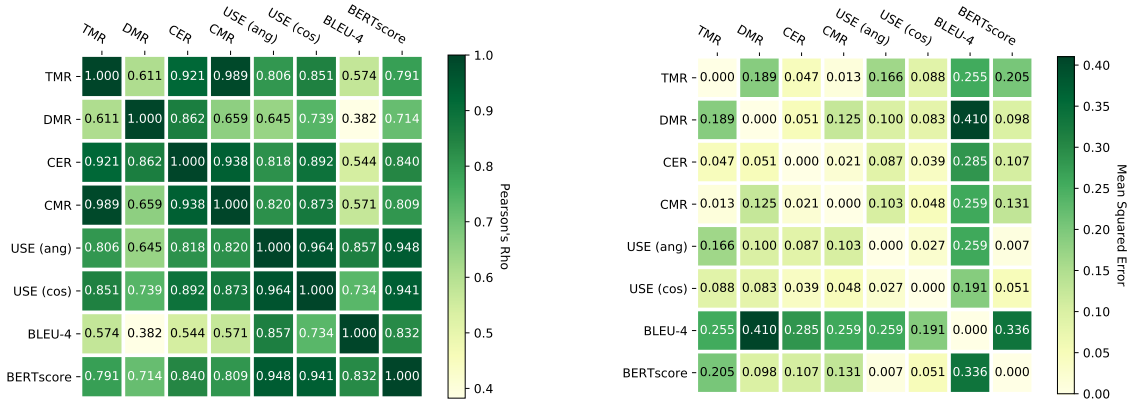

Figure 6: Correlation coefficients (left) and mean squared error (right) when comparing all similarity scores for all experiments.

difference in this way does not necessarily capture the relevant semantic differences.

Moreover, the set of dialogue contexts has a high impact on the absolute score for all similarity metrics but not on the trend when comparing two dialog behaviour models with each other.

For future work, the analysis must be more fine-grained, e.g., by sub-dividing the set of dialogue contexts into meaningful sub-sets. Furthermore, the proposed evaluation method is also suitable for directly looking at the actual behaviour of models by identifying crucial dialogue contexts and comparing the actual system reaction.

References

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. 2017. A benchmarking environment for reinforcement learning based task oriented dialogue management. In *Deep Reinforcement Learning Symposium, 31st Conference on Neural Information Processing Systems (NIPS)*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jennifer Chu-Carroll and Jill Suzanne Nickerson. 2000. Evaluating automatic dialogue strategy adaptation for a spoken dialogue system. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, page 202–209, USA. Association for Computational Linguistics.

Lucie Daubigney, Matthieu Geist, and Olivier Pietquin. 2012. Off-policy Learning in Large-scale POMDP-based Dialogue Systems. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 4989–4992, Kyoto (Japan). IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhauser, and Gwendolyn Campbell. 2011. Exploring user satisfaction in a tutorial dialogue system. In *Proceedings of the SIGDIAL 2011 Conference*, pages 162–172. Association for Computational Linguistics.

Milica Gašić and Steve J. Young. 2014. Gaussian processes for POMDP-based dialogue manager optimization. *IEEEACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Helen Hastie. 2012. Metrics and evaluation of spoken dialogue systems. In Oliver Lemon and Olivier Pietquin, editors, *Data-Driven Methods for Adaptive Spoken Dialogue Systems*, pages 131–150. Springer New York.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, volume 263.

Oliver Lemon and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688.

Esther Levin and Roberto Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogue strategies. In *Eurospeech*, volume 97, pages 1883–1886.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Sebastian Möller, Klaus-Peter Engelbrecht, and Robert Schleicher. 2008. Predicting the quality and usability of spoken dialogue services. *Speech Communication*, 50(8-9):730–744.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jost Schatzmann and Steve J. Young. 2009. The hidden agenda user simulation model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(4):733–747.

Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: Assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication*, 74:12–36.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings*

of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Pei-Hao Su, M. Gašić, N. Mrkšić, L. Rojas-Barahona, Stefan Ultes, D. Vandyke, T. H. Wen, and S. Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441. Association for Computational Linguistics.

Stefan Ultes. 2019. Improving interaction quality estimation with BiLSTMs and the impact on dialogue policy learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 11–20, Stockholm, Sweden. Association for Computational Linguistics.

Stefan Ultes, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, Lina Rojas-Barahona, Pei-Hao Su, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017a. Domain-independent user satisfaction reward estimation for dialogue policy learning. In *Interspeech*, pages 1721–1725. ISCA.

Stefan Ultes, Matthias Kraus, Alexander Schmitt, and Wolfgang Minker. 2015. Quality-adaptive spoken dialogue initiative selection and implications on reward modelling. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 374–383. ACL.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017b. PyDial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. Association for Computational Linguistics.

David Vandyke, Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialogue success classifiers for policy training. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 763–770. IEEE.

Marilyn Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: a framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics (EACL)*, pages 271–280, Morristown, NJ, USA. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego, California. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, pages 438–449. ACL.

Steve J. Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Example Similarity Scores

	Action a / Prompt p	Action a' / Prompt p'	*TM*	*DM*	*$\tilde{CE}$*	*CM*	*USE (ang)*	*USE (cos)*	*BLEU-4*	*BERTscore*
0	*request(pricerange)* what price range are you interested in	*request(food)* what kind of food are you looking for	0	1	0.5	0.2	0.68	0.54	0.07	0.82
1	*request(food)* what type of food are you looking for	*request(pricerange)* what price range are you interested in	0	1	0.5	0.2	0.67	0.52	0.07	0.83
2	*inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")* la mimosa is a expensive mediterranean restaurant in the centre area	*inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")* la mimosa is a expensive mediterranean restaurant in the centre area	1	1	1	1	1.00	1.00	1.00	1.00
3	*inform(food="mediterranean", pricerange="expensive", name="shiraz restaurant", area="centre")* shiraz restaurant is in the centre area and is expensively priced and serves mediterranean food	*inform(food="mediterranean", pricerange="expensive", name="shiraz restaurant", area="centre")* shiraz restaurant is in the centre area and serves mediterranean food and is expensively priced	1	1	1	1	0.95	0.99	0.80	0.98
4	*inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")* la mimosa is in the centre area that is expensively priced and serves mediterranean food	*request(pricerange)* what price range would you like	0	0	0	0.1	0.53	0.09	0.00	0.61