Comparing Pre-Trained Embeddings and Domain-Independent Features for Regression-Based Evaluation of Task-Oriented Dialogue Systems

Kallirroi Georgila
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Los Angeles, CA 90094-2536, USA
kgeorgila@ict.usc.edu

Abstract

We use Gaussian Process Regression to predict different types of ratings provided by users after interacting with various task-oriented dialogue systems. We compare the performance of domain-independent dialogue features (e.g., duration, number of filled slots, number of confirmed slots, word error rate) with pre-trained dialogue embeddings. These pre-trained dialogue embeddings are computed by averaging over sentence embeddings in a dialogue. Sentence embeddings are created using various models based on sentence transformers (appearing on the Hugging Face Massive Text Embedding Benchmark leaderboard) or by averaging over BERT word embeddings (varying the BERT layers used). We also compare pre-trained embeddings extracted from human transcriptions with pre-trained embeddings extracted from speech recognition outputs, to determine the robustness of these models to errors. Our results show that overall, for most types of user satisfaction ratings and advanced/recent (or sometimes less advanced/recent) pre-trained embedding models, using only pre-trained embeddings outperforms using only domain-independent features. However, this pattern varies depending on the type of rating and the embedding model used. Also, pre-trained embeddings are found to be robust to speech recognition errors, more advanced/recent embedding models do not always perform better than less advanced/recent ones, and larger models do not necessarily outperform smaller ones. The best prediction performance is achieved by combining pre-trained embeddings with domain-independent features.

1 Introduction

The quality of a human-machine dialogue interaction can be influenced by various factors, such as the domain/genre of dialogue, the dialogue system capabilities, and the user expertise and expectations. This makes it very difficult to define what a successful dialogue should look like, and evaluate system performance and predict user satisfaction. Thus, despite many years of research, dialogue evaluation still remains an unsolved problem.

In this paper, our focus is on task-oriented dialogue, and specifically on predicting user satisfaction after their interaction with the dialogue system. We use the Communicator corpus (Walker et al., 2001a, 2002) containing the logs of user interactions with 8 spoken dialogue systems. The user's task is to book a flight and in some cases also make hotel or car-rental arrangements. Each dialogue log is accompanied by user ratings after their interaction with the system. An example dialogue excerpt is shown in Figure 1 in the Appendix.

The original Communicator corpus contains system and user utterances (both human transcriptions and speech recognition outputs), timing information, and speech act and task annotations for the system's side of the conversation. An extended version of this corpus was developed by Georgila et al. (2005b, 2009) via automatic annotation. Georgila et al. (2005b, 2009) added speech act and task annotations for the user's side of the conversation, and dialogue context annotations, e.g., filled slots, filled slots values, grounded slots, speech acts history.

In this paper, we use Gaussian Process Regression for predicting user satisfaction ratings, because in our recent work (Georgila, 2022) it was shown to perform better than other regression methods, for this task and corpus. In our previous work (Georgila, 2022), we considered only domain-independent features (e.g., duration, number of filled slots, number of confirmed slots, word error rate). These features were domain-independent because they were just based on counts, and no lexical, semantic, or specific to the task information was used. Here, in addition to these domain-independent features, we also use pre-trained dialogue embeddings extracted from system and user utterances.

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 610–623
September 18–20, 2024.

Our pre-trained dialogue embeddings are computed by averaging over sentence embeddings for each dialogue. Sentence embeddings are created using various models based on sentence transformers (Reimers and Gurevych, 2019) (appearing on the Hugging Face Massive Text Embedding Benchmark leaderboard), or by averaging over BERT word embeddings (Wieting et al., 2016; Coates and Bollegala, 2018) (varying the BERT layers used). By definition, these embeddings are domain-dependent because they encode lexical and semantic information about the domain. Also, we compare pre-trained embeddings extracted from human transcriptions versus pre-trained embeddings extracted from automatic speech recognition (ASR) outputs, to determine the robustness of these models to errors, which is an understudied research question (Mousavi et al., 2024). We investigate what level of performance can be achieved just by relying on the words of the system and user utterances from which we compute pre-trained dialogue embeddings, whether using only embeddings outperforms using only domain-independent features, and whether combining embeddings and domain-independent features can result in performance gains. We also examine the impact on performance of different feature combinations.

To our knowledge, our work is one of a few studies (if not the first) to compare such a large variety of pre-trained embeddings (including the most recent embedding models by OpenAI) under the same conditions, and the first study to do so for predicting user ratings in task-oriented dialogue. This is also the first work to compare all these different types of pre-trained embeddings with various domain-independent features for user ratings' prediction in task-oriented dialogue. Last, but not least, this is one of a very limited number of studies comparing the performance of pre-trained embeddings on human transcriptions versus ASR outputs, and the first study to do so for user ratings' prediction.

2 Related Work

Despite many years of research, dialogue evaluation still remains an unsolved problem (Hastie, 2012; Deriu et al., 2021; Mehri et al., 2022). For task-oriented dialogue there are subjective evaluation metrics, such as user satisfaction, computed using information from surveys (Hone and Graham, 2000; Paksima et al., 2009), and objective metrics, such as task completion and dialogue length, computed using information from interaction logs.

PARADISE (Walker et al., 2000) is the most well-known framework for automatic evaluation of task-oriented dialogue. The goal of PARADISE is to optimize user satisfaction (or another desired quality) by formulating it as a linear combination of various factors, such as task success and dialogue cost (e.g., dialogue length, ASR errors). Weights calculated via linear regression determine the contribution of each factor. PARADISE can be used to predict user satisfaction at the end of the dialogue, but can also be applied to any point in the dialogue prior to completion. Generally it is useful to be able to evaluate on the fly how the dialogue is unfolding, so that appropriate measures can be taken (e.g., transfer to a human operator), if a dialogue is problematic. Based on this idea, much work has been done on estimating user satisfaction at the system-user exchange level rather than rating the whole dialogue (Engelbrecht et al., 2009; Higashinaka et al., 2010; Ultes and Minker, 2014; Schmitt and Ultes, 2015).

For chatbots and other non-task-oriented dialogue systems it is not clear what success means, and it is common to use subjective evaluations of system responses (e.g., coherence, engagingness) given some context, or use word-overlap similarity metrics (e.g., BLEU, ROUGE) even though such metrics do not correlate well with human judgments of dialogue quality (Liu et al., 2016). Recently, new evaluation metrics have been proposed for open-domain dialogue leveraging pre-trained language models such as BERT and DialoGPT (Mehri and Eskenazi, 2020a,b; Ghazarian et al., 2020), and commonsense knowledge bases (Ghazarian et al., 2023).

In this paper, we focus on predicting user satisfaction ratings for the whole dialogue. We use Gaussian Process Regression (GPR) for predicting user satisfaction ratings, because in our recent work (Georgila, 2022) it was shown to perform better than other regression methods, for this task and corpus. In our previous work (Georgila, 2022), we only used domain-independent features, but here we also use pre-trained dialogue embeddings extracted from system and user utterances.

Linear regression has been used before for dialogue evaluation (Walker et al., 2000, 2001b; Cervone et al., 2018; Georgila et al., 2019, 2020; Georgila, 2022). Also, Support Vector Regression has been used before for dialogue evaluation (Cervone et al., 2018; Georgila, 2022).

We use GPR for our experiments because modern regression methods are a natural evolution of the PARADISE framework. Furthermore, we do not have many data points for data-hungry methods such as neural networks. As we will see in section 3, we only have 500 data points in the training data and 506 data points in the test data.

3 Data and Features

We use the Communicator corpus (Walker et al., 2001a, 2002) because it has been used before for this task, but also because it is one of a few task-oriented dialogue corpora that include user ratings. Other popular corpora, such as MultiWOZ (Budzianowski et al., 2018; Eric et al., 2020), do not include user ratings or ASR outputs.

The original Communicator corpus contains system and user utterances (both human transcriptions and ASR outputs), timing information, and speech act and task annotations for the system's side of the conversation based on the DATE scheme (Walker and Passoneau, 2001). An extended version of this corpus was developed by Georgila et al. (2005b, 2009) via automatic annotation. Based on the ASR outputs, speech act and task annotations for the user's side of the conversation were added, as well as dialogue context annotations, e.g., filled slots, filled slots values. Basically these extended annotations are the kind of information one would get by deploying a dialogue system, but because the original corpus did not include such information, Georgila et al. (2005b, 2009) reconstructed it.

Georgila et al. (2009) verified the validity and reliability of these automatic annotations by evaluating them with respect to the task completion metrics of the original corpus and in comparison to manually annotated data. The utility of these extended annotations has been demonstrated by their use by various researchers for different purposes, such as learning dialogue policies (Henderson et al., 2005; Frampton and Lemon, 2006; Henderson et al., 2008; McLeod et al., 2019) and building simulated users (Schatzmann et al., 2005; Georgila et al., 2005a, 2006).

In the Appendix, Figure 1 shows an example dialogue excerpt including speech act and task annotations, and Figure 2 depicts an example dialogue state.

These extended dialogue context annotations are divided into two broad categories: logs of the current status of the slots ('FilledSlotsStatus', 'FilledSlotsValuesStatus', 'GroundedSlotsStatus'), and logs containing information about how the status of the slots has changed over time through the dialogue ('FilledSlotsHist', 'FilledSlotsValuesHist', 'GroundedSlotsHist'). The former inform us about the current status of the slots, and may only contain one instance per slot. The latter provide information about the order in which slots have been filled or confirmed, and may contain several instances of the same slot. The annotations also include the history of speech acts and tasks.

For our experiments we use the 2001 collection, which consists of 1,683 dialogues between human users and 8 dialogue systems. These systems vary in their dialogue policies, e.g., some of them request multiple pieces of information at the same time, others request explicit confirmation, others request implicit confirmation, etc. Overall there are 78,718 turns (39,419 system turns and 39,299 user turns). Similarly to Georgila (2022), for our experiments we only used dialogues for which all user ratings were available: ATT (157 dialogues), BBN (137 dialogues), CMU (69 dialogues), COLORADO (157 dialogues), IBM (77 dialogues), LUCENT (140 dialogues), MIT (166 dialogues), and SRI (103 dialogues). The first half of the dialogues from each system is used for training (500 dialogues in total) and the rest for testing (506 dialogues in total).

So our task is to predict the following user satisfaction ratings on a Likert scale (1-5, higher is better): ease of the tasks the user had to accomplish ('Task-Ease'), whether it was easy or not to understand the system ('System-Comprehend-Ease'), the user's expertise ('User-Expertise'), whether the system behaved as expected ('System-Behaved-As-Expected'), and if the user would use the system again in the future ('System-Future-Use'). We use the same domain-independent features as Georgila (2022), with the addition of the number of times the user requested a 'start-over'. Our 17 domain-independent features are divided into 4 categories:

- **duration-related features (9):** overall duration, duration of the system talking part, duration of the user talking part, overall average duration per utterance, average duration per system utterance, average duration per user utterance, number of overall speech acts, number of system speech acts, number of user speech acts;
- **slots-related features (6):** number of filled

slots, number of filled slots without any 'null' values, number of grounded slots, number of filled slots in the dialogue history, number of filled slots without any 'null' values in the dialogue history, number of grounded slots in the dialogue history (all at the end of the dialogue) – we distinguish between slots filled with normal versus 'null' values as an extra piece of information;

- **word error rate (WER) (1)**: calculated as the edit distance between the ASR output and the transcription of the user utterance (this information was included in the original Communicator corpus);
- **start-over feature (1)**: number of 'start-over' requests by the user extracted from the human transcription or the ASR output.

All these features are automatically extracted from the data. Feature values are replaced with z-scores by subtracting from each feature value the mean for that feature and then dividing by the standard deviation for that feature. For each feature, the mean and standard deviation are calculated on the training data.

We use 4 variations of these feature combinations: 'orig-man' (original corpus with features from manual annotations such as human transcriptions of speech plus fully automatic annotations), 'orig-auto' (original corpus with fully automatic annotations), 'ext-man' (extended corpus with features from manual annotations plus fully automatic annotations), and 'ext-auto' (extended corpus with fully automatic annotations). So 'ext-man' is a super set of 'orig-man', and 'ext-auto' is a super set of 'orig-auto', because the extended corpus contains all the annotations of the original corpus plus new annotations (note that, as mentioned above, these new annotations are automatically generated). Also, 'orig-man' and 'ext-man' include both manual and automatic annotations, whereas 'orig-auto' and 'ext-auto' include only automatic annotations.

For duration, the number of user speech acts is only used in 'ext-man' and 'ext-auto', because (as discussed above) the original corpus did not include annotations of the user's side of the conversation. Likewise, slots-related features are only part of the extended corpus ('ext-man' and 'ext-auto'). Information about WER is only part of the manual annotations because it can be computed only when human transcriptions are available.

	orig -man	**orig -auto**	**ext -man**	**ext- auto**
duration	x	x	x	x
slots			x	x
WER	x		x	
start-over	x	x	x	x

Table 1: Categories of feature combinations; x means that a feature category is included.

For clarity, Table 1 shows exactly which features are used in each category.

We also compute pre-trained dialogue embeddings by averaging over sentence embeddings for each dialogue. Sentence embeddings are created using various models based on sentence transformers (appearing on the Hugging Face MTEB leaderboard), or by averaging over BERT word embeddings (varying the BERT layers used). We do not calculate z-scores for the embeddings.

We use the following types of embeddings from Hugging Face and OpenAI, and in parentheses we can see the sizes of the vectors they produce:

- 'glove-6B-300d' (300) (Pennington et al., 2014),
- 'all-distilroberta-v1' (768),
- 'all-mpnet-base-v2' (768),
- 'all-MiniLM-L6' (384),
- 'all-MiniLM-L12' (384),
- 'e5-small-v2' (384),
- 'e5-base-v2' (768),
- 'e5-large-v2' (1024) (Wang et al., 2024),
- 'gte-small' (384),
- 'gte-base' (768),
- 'gte-large' (1024),
- 'bge-small-en-v1.5' (384),
- 'bge-base-en-v1.5' (768),
- 'bge-large-en-v1.5' (1024),
- OpenAI's 'text-embedding-3-small' (1536),
- OpenAI's 'text-embedding-3-large' (3072).

The latest models of OpenAI have a new feature that allows selecting the size of the generated vector. According to OpenAI, this compressed vector retains its concept-representing properties. For 'text-embedding-3-small' we experimented with 3 vector sizes (50, 256, 1536) and for 'text-embedding-3-large' with 3 vector sizes (50, 512, 3072).

Because we only have 500 data points in the training data and 506 data points in the test data, and large vector sizes, we also applied Principal Component Analysis (PCA) for dimensionality reduction, with "whitening" to ensure that the resulting features are less correlated with each other. Huang et al. (2021) and Su et al. (2021) have found that "whitening" can enhance the isotropy of sentence embeddings, with the additional advantage of reducing their dimensionality.

We generated results with different numbers of PCA components, and we show results with a value of 50 which performed well for all models (better e.g., than 75 or 100). Of course, when we generated vectors of size 50 from OpenAI, we did not apply PCA. Note that we apply PCA only to the embedding vectors (the domain-independent features are not affected by PCA).

4 Experiments and Results

In our previous work (Georgila, 2022), we compared several state-of-the-art regression methods, and showed that GPR with an exponential kernel or a rational quadratic kernel performed the best. Thus, here we use GPR with an exponential kernel. Also, by performing more experiments, we verified again that GPR outperforms other regression methods, and that using an exponential kernel produces competitive results for different types of embeddings. For all GPR experiments we vary the length scale, and we report results for length scale equal to 1 (higher length scale values indicate smoother learned functions). Varying the length scale did not produce significant differences. GPR is considered as the state-of-the-art for regression, and has been used before in the NLP community for machine translation quality estimation (Cohn and Specia, 2013) and emotion prediction (Beck et al., 2014). For all our experiments we use the GPy library[1], and GPR is applied after PCA.

To evaluate our models, for each of the 5 ratings, we calculate the Root Mean Square Error (RMSE). RMSE measures the average error between the model predictions and the ground truth (the ratings in the test data). Its value varies from 0 to 4, given that user ratings are on a scale from 1 to 5. Lower RMSE values are better.

4.1 Using Only Pre-Trained Embeddings

Table 2 shows results in terms of RMSE when using only our embedding models (not including domain-independent features), based on human transcriptions ('man') and ASR outputs ('auto').

For BERT, we experimented with various layer combinations, and we report the best results. We found that it helps to use the first layer (L1) together with the last layers (L10, L11, L12). Other researchers have also looked into the impact of different BERT layers, reporting that sometimes it is better not to use the last layer, as it is largely fine-tuned to the specific task (Li et al., 2020; Huang et al., 2021; Su et al., 2021). Although differences were small, the best layer combination was L1-10-11 which means that the vectors of layers L1, L10, and L11 were averaged. Layer L1 alone also produced competitive results. We hypothesize that layer L1 is important for our task because it encodes lexical information rather than semantic meaning, and for dialogue evaluation some words such as "start-over" or "no" can be quite predictive.

For each BERT layer, we also compared averaging of word embeddings versus using the output of the [CLS] token, and averaging performed better. Thus, here we only present results with averaging (see Table 5 in the Appendix for a comparison between averaging and using the output of the [CLS] token). Reimers and Gurevych (2019) showed that averaging of BERT word embeddings or using the output of the [CLS] token produces rather poor sentence embeddings, often worse than averaging GloVe word embeddings (even though BERT word embeddings are generally considered superior to GloVe word embeddings). However, this was not the case in our experiments where BERT most times (depending on the layer combination) worked better than GloVe.

Overall, 'glove-6B-300d', 'all-distil-roberta-v1', 'all-mpnet-base-v2', 'all-MiniLM-L6', and 'all-MiniLM-L12' did not perform well compared to the rest of the models. For 'e5', 'gte', and 'bge', the small versions performed well, and that was the case also for the large version of 'bge'. This is interesting because it shows that larger models do not necessarily perform better than smaller models. The question that arises is whether larger models

[1] https://gpy.readthedocs.io/en/deploy/

	TaskEase		SysComEase		UserExp		SysBehExp		SysFutUse	
	man	auto	man	auto	man	auto	man	auto	man	auto
bert-L1	1.259	1.28	**1.163**	1.174	**1.287**	1.303	1.291	1.324	**1.363**	1.379
bert-L1-10-11	**1.254**	1.256	1.173	**1.167**	1.302	**1.289**	**1.283**	1.285	1.366	1.371
bert-L1-11-12	1.255	1.256	1.174	1.173	1.302	1.298	1.288	1.287	1.364	1.373
bert-L1-10-11-12	1.255	**1.253**	1.173	1.169	1.305	1.291	1.285	**1.28**	1.367	**1.37**
glove-6B-300d	1.294	1.286	1.177	1.176	1.296	1.294	1.33	1.321	1.385	1.389
all-distilroberta-v1	1.28	1.296	**1.171**	**1.174**	1.306	1.309	1.317	1.325	1.388	1.392
all-mpnet-base-v2	**1.271**	**1.278**	1.179	1.181	1.29	**1.289**	**1.305**	**1.317**	**1.377**	**1.379**
all-MiniLM-L6	1.276	1.282	1.172	**1.174**	**1.287**	1.295	1.312	1.318	1.381	1.387
all-MiniLM-L12	1.273	1.282	1.188	1.187	**1.287**	1.291	1.318	1.321	1.389	1.388
e5-small-v2	**1.254**	1.272	**1.175**	1.183	**1.285**	**1.286**	1.311	1.316	**1.376**	1.38
e5-base-v2	1.281	1.291	1.195	1.192	1.303	1.29	1.32	1.333	1.39	1.387
e5-large-v2	1.27	**1.271**	1.176	**1.181**	1.293	1.302	**1.299**	**1.308**	1.378	**1.375**
gte-small	**1.26**	**1.264**	**1.168**	**1.175**	**1.285**	**1.283**	1.303	1.306	**1.37**	1.373
gte-base	1.284	1.294	1.186	1.189	1.3	1.302	1.323	1.326	1.385	1.387
gte-large	1.265	1.27	1.172	1.178	1.29	1.286	**1.299**	**1.3**	1.374	**1.369**
bge-small-en-v1.5	**1.262**	**1.261**	1.181	1.185	1.296	1.29	1.299	1.31	1.374	1.372
bge-base-en-v1.5	1.281	1.283	1.186	1.186	1.308	1.304	1.315	1.311	1.392	1.386
bge-large-en-v1.5	**1.262**	1.271	**1.168**	**1.175**	**1.284**	**1.282**	**1.285**	**1.292**	**1.363**	**1.367**
openai-small-50	1.323	1.32	1.191	1.191	1.305	1.306	1.35	1.344	1.399	1.398
openai-small-256	1.31	1.297	1.183	1.183	1.302	1.299	1.347	1.326	1.396	1.387
openai-small-1536	1.263	1.268	1.161	1.163	1.285	**1.284**	1.313	1.312	1.38	1.374
openai-large-50	1.27	1.286	**1.149**	**1.158**	1.303	1.298	**1.3**	**1.308**	1.37	1.379
openai-large-512	**1.258**	**1.266**	1.161	1.165	**1.283**	1.285	1.303	1.312	1.362	1.361
openai-large-3072	1.264	1.268	1.164	1.166	1.297	1.29	1.31	1.317	**1.36**	**1.36**

Table 2: RMSE values when using only pre-trained embeddings (not including domain-independent features), based on the human transcriptions ('man') and the ASR outputs ('auto'). For each block, the best value for each column is shown in a different color (specific to that block) and in bold. **The best value for each column across all blocks is shown in black and in bold.**

were negatively affected by being compressed more than smaller models, given that we used only 50 PCA components. However, as we see with the OpenAI models, this is not the case. The 'openai-large-3072' model was significantly compressed and yet performed well. When we experimented with different numbers of components the trends were the same, i.e., the small versions of 'e5', 'gte', and 'bge' still worked better than their base and large counterparts, with the exception of 'bge' where the large version also performed well.

For the OpenAI models, we can see that the models based on 'text-embedding-3-large' worked better than the models based on 'text-embedding-3-small'. Interestingly, 'openai-large-50' works very well. Note that this is the model where the compression was done by OpenAI (not by our using of PCA). It is not clear what kind of dimensionality reduction algorithm OpenAI uses. For some ratings, we can see that applying PCA on 'openai-large-512' and 'openai-large-3072' works better than 'openai-large-50'.

Overall, differences in results across models are small, but there are trends:

- Larger models are not necessarily better than smaller models.
- More advanced/recent models do not always perform the best.
- Pre-trained embeddings are quite robust to ASR errors for our task, given that differences in RMSE values between corresponding 'man' and 'auto' models are small.

4.2 Comparing Pre-Trained Embeddings and Domain-Independent Features

Table 3 shows the full results for the best performing embedding models from Table 2. So, for example, 'orig-em-man' means manual and automatic

	bert -L1	bert-L1-10 -11	e5-small -v2	gte-small	bge-small-en-v1.5	bge-large-en-v1.5	openai-large -50	openai-large -512	openai-large -3072
Task-Ease									
orig-man: 1.292 **ext-man:** 1.276 **orig-auto:** 1.311 **ext-auto:** 1.284									
em-man	1.259	**1.254**†	**1.254**†	1.26	1.262	1.262	1.27	1.258	1.264
orig-em-man	1.236	1.242	**1.235**†	1.238	1.24	1.249	1.244	1.241	1.246
ext-em-man	1.235	1.241	**1.233**†	1.237	1.239	1.249	1.245	1.241	1.245
em-auto	1.28	**1.256**‡	1.272	1.264	1.261	1.271	1.286	1.266	1.268
orig-em-auto	1.253	1.245	1.248	**1.236**‡	1.237	1.258	1.256	1.248	1.25
ext-em-auto	1.252	1.244	1.244	**1.233**‡	1.235	1.257	1.256	1.247	1.248
System-Comprehend-Ease									
orig-man: 1.174 **ext-man:** 1.156 **orig-auto:** 1.178 **ext-auto:** 1.158									
em-man	1.163	1.173	1.175	1.168	1.181	1.168	**1.149**†	1.161	1.164
orig-em-man	1.138	1.152	1.15	1.141	1.152	1.149	**1.134**†	1.143	1.147
ext-em-man	1.136	1.151	1.148	1.14	1.15	1.148	**1.133**†	1.143	1.147
em-auto	1.174	1.167	1.183	1.175	1.185	1.175	**1.158**‡	1.165	1.166
orig-em-auto	1.148	1.15	1.16	1.149	1.154	1.155	**1.14**‡	1.149	1.15
ext-em-auto	1.147	1.149	1.157	1.148	1.152	1.153	**1.139**‡	1.15	1.15
User-Expertise									
orig-man: 1.286 **ext-man:** 1.295 **orig-auto:** 1.286 **ext-auto:** 1.293									
em-man	1.287	1.302	1.285	1.285	1.296	1.284	1.303	**1.283**	1.297
orig-em-man	1.276	1.296	1.274	**1.268**†	1.278	1.272	1.284	1.269	1.281
ext-em-man	1.282	1.305	1.278	**1.274**†	1.284	1.279	1.289	**1.274**†	1.286
em-auto	1.303	1.289	1.286	1.283	1.29	**1.282**	1.298	1.285	1.29
orig-em-auto	1.288	1.283	1.27	**1.262**‡	1.269	1.268	1.28	1.271	1.275
ext-em-auto	1.294	1.288	1.272	**1.266**‡	1.273	1.273	1.286	1.275	1.279
System-Behaved-As-Expected									
orig-man: 1.301 **ext-man:** 1.278 **orig-auto:** 1.33 **ext-auto:** 1.286									
em-man	1.291	**1.283**	1.311	1.303	1.299	1.285	1.3	1.303	1.31
orig-em-man	1.268	1.269	1.282	1.271	1.27	**1.267**†	**1.267**†	1.282	1.288
ext-em-man	1.262	1.262	1.273	1.264	1.263	**1.259**†	1.26	1.274	1.279
em-auto	1.324	**1.285**‡	1.316	1.306	1.31	1.292	1.308	1.312	1.317
orig-em-auto	1.291	**1.266**‡	1.287	1.274	1.277	1.273	1.273	1.29	1.294
ext-em-auto	1.282	**1.259**‡	1.278	1.265	1.267	1.265	1.266	1.281	1.284
System-Future-Use									
orig-man: 1.397 **ext-man:** 1.394 **orig-auto:** 1.41 **ext-auto:** 1.395									
em-man	1.363	1.366	1.376	1.37	1.374	1.363	1.37	1.362	**1.36**†
orig-em-man	**1.339**†	1.353	1.357	1.345	1.35	1.346	1.36	1.344	1.342
ext-em-man	**1.337**†	1.348	1.351	1.341	1.347	1.344	1.36	1.339	1.339
em-auto	1.379	1.371	1.38	1.373	1.372	1.367	1.379	1.361	**1.36**‡
orig-em-auto	1.358	1.355	1.36	1.346	1.347	1.351	1.362	1.344	**1.343**‡
ext-em-auto	1.354	1.35	1.354	1.341	**1.34**‡	1.347	1.362	**1.34**‡	**1.34**‡

Table 3: RMSE values for different combinations of embedding models and features. The best value for each row (i.e., best model) is shown in a color specific to that rating and in bold. **The best value for each rating is shown in black and in bold;** † means that 'em-man', 'orig-em-man', or 'ext-em-man' are significantly better than either 'orig-man' or 'ext-man' ($p < 0.05$ or better); ‡ means that 'em-auto', 'orig-em-auto', or 'ext-em-auto' are significantly better than either 'orig-auto' or 'ext-auto' ($p < 0.05$ or better). Also, 'em-auto' means only embeddings from ASR outputs, 'ext-em-man' means manual and automatic annotations from the extended corpus plus embeddings from human transcriptions, etc.

annotations from the original corpus ('orig-man') plus embeddings extracted from human transcriptions, 'ext-em-auto' means only automatic annotations from the extended corpus ('ext-auto') plus embeddings extracted from ASR outputs, 'em-man' means only embeddings extracted from human transcriptions, etc. Here, for each rating, we also see results using only the domain-independent features without embeddings; these results are slightly different from the results reported by Georgila (2022) because we additionally use the 'start-over' feature.

For all ratings, we measure statistical significance between the best values of 'em-man/orig-em-man/ext-em-man', and either 'orig-man' or 'ext-man'. Sometimes, the difference between 'em-man/orig-em-man/ext-em-man' and 'orig-man' is significant, but the difference between 'em-man/orig-em-man/ext-em-man' and 'ext-man' is not significant (or vice versa). In this case, we still mark the difference as significant in Table 3 (to avoid over-crowding Table 3 with too many different markings). We also measure statistical significance between the best values of 'em-auto/orig-em-auto/ext-em-auto', and either 'orig-auto' or 'ext-auto'. We mark differences as significant in the same way as explained above.

For all statistical significance calculations, for comparing models, we use the squared error values and the Wilcoxon signed-rank test with Holm-Bonferroni correction for repeated measures. We did not test for significance all combinations, but roughly a difference in the RMSE values of 0.02 or larger is likely to be significant at $p < 0.05$ or better (depending on the variance of course).

For 'Task-Ease' the best models are 'bert-L1-10-11', 'e5-small-v2', and 'gte-small'. Using embeddings (with or without domain-independent features), based on human transcriptions ('man') or ASR outputs ('auto'), outperforms using only domain-independent features.

For 'System-Comprehend-Ease', the best model is 'openai-large-50' for all feature combinations. There are significant differences between the RMSE values of this model and the RMSE values of the domain-independent features.

For 'User-Expertise', the best models are 'gte-small', 'bge-large-en-v1.5', and 'openai-large-512'. Here differences between using only domain-independent features and using only embeddings are small and not significant, but they become significant once we combine domain-independent features and embeddings.

	Bas 3	Bas maj	BM
Task-Ease	1.471	1.721	**1.233**
Sys-Compr-Ease	1.421	1.285	**1.133**
User-Expertise	1.431	1.41	**1.262**
Sys-Behave-Exp	1.433	1.705	**1.259**
Sys-Future-Use	1.516	2.321	**1.337**

Table 4: RMSE values for the baseline always predicting score 3 (Bas 3) and the majority baseline (Bas maj), and the best of our models (BM). **The best value for each row is shown in bold.**

For 'System-Behaved-As-Expected', the best models are 'bert-L1-10-11', 'bge-large-en-v1.5', and 'openai-large-50'. For the best model ('bert-L1-10-11') and for 'auto' using only embeddings significantly outperforms using only domain-independent features. Combining domain-independent features and embeddings results in significant differences for both 'man' and 'auto'.

For 'System-Future-Use', the best models are 'bert-L1', 'bge-small-en-v1.5', 'openai-large-512', and 'openai-large-3072'. Using only embeddings performs much better than using only domain-independent features ($p < 0.01$ for 'man' and $p < 0.001$ for 'auto'). Combining domain-independent features and embeddings further improves performance ($p < 0.001$ for both 'man' and 'auto').

Similarly to Georgila (2022), we also implemented simple baselines. Table 4 shows results for RMSE for each type of rating, for the baseline that always predicts score 3, the majority baseline, and the best result of our models taken from Table 3. As expected, our models significantly outperform the baselines ($p < 0.001$).

Below we summarize our findings from comparing pre-trained embeddings with domain-independent features:

- For most types of user satisfaction ratings and advanced/recent pre-trained embedding models, using only pre-trained dialogue embeddings outperforms using only domain-independent features.
- Combining pre-trained embeddings and domain-independent features is better than just using pre-trained embeddings.
- Differences between corresponding 'man' and 'auto' models are small, and thus, we conclude that pre-trained dialogue embeddings are quite robust to ASR errors for our task.

- Using domain-independent features from the extended annotations sometimes helps, but overall, performance is similar to using features from the original annotations.

5 Conclusions and Discussion

We used GPR for predicting user satisfaction ratings. We used both domain-independent features and pre-trained dialogue embeddings extracted from system and user utterances. Our pre-trained dialogue embeddings were computed by averaging over sentence embeddings for each dialogue. Sentence embeddings were created using various models based on sentence transformers (appearing on the Hugging Face MTEB leaderboard) or by averaging over BERT word embeddings (varying the BERT layers used).

Our results showed that overall, for most types of user satisfaction ratings and advanced/recent pre-trained embedding models, using only pre-trained dialogue embeddings outperforms using only domain-independent features. This is very interesting, because it shows that we can do quite well relying only on information from words and system and user utterances, without any additional features. Combining embeddings and domain-independent features performed the best. This is also very interesting and could potentially revive interest in using domain-independent features. Although overall extracting domain-independent features from the extended annotations helped, performance was similar to using domain-independent features from the original annotations.

Interestingly, some simpler models (e.g., 'bert-L1') performed better than more complex and more recent models. Also, larger models did not necessarily outperform smaller ones. Because differences between corresponding 'man' and 'auto' models were small, we conclude that pre-trained embeddings are quite robust to ASR errors for our task. Overall, RMSE values ranged roughly from 1.1 to 1.4 depending on the model and feature combination.

Our overall contributions are as follows:

- To our knowledge, our work is one of a few studies (if not the first) to compare such a large variety of pre-trained embeddings (including the most recent embedding models by OpenAI) under the same conditions.

- Our work is the first study to compare such a large variety of pre-trained embeddings (including the most recent embedding models by OpenAI) under the same conditions for predicting user ratings in task-oriented dialogue.

- Our work is also the first study to compare all these different types of pre-trained embeddings and various domain-independent features for user ratings' prediction in task-oriented dialogue.

- Finally, this is one of a very limited number of studies comparing the performance of pre-trained embeddings on human transcriptions versus ASR outputs, and the first study to do so for user ratings' prediction.

Throughout our experiments, to construct dialogue embeddings we used averaging (Wieting et al., 2016; Coates and Bollegala, 2018), but the problem with averaging is that it can result in loss of important conversational information (Reimers and Gurevych, 2019). For example, not all parts of a dialogue are of equal importance, and by trying to encode everything we may end up compressing too much information from parts that really matter.

Very little work has been done on constructing dialogue embeddings using techniques different from averaging. A notable recent attempt to construct dialogue embeddings is Dial2vec (Liu et al., 2022). Dial2vec uses self-guided contrastive learning (leveraging both positive and negative examples) and considers a dialogue as an information exchange process between two interlocutors. It learns embeddings for both interlocutors with the help of each other, and then the dialogue embedding is obtained by an aggregation of embeddings of the interlocutors. Dial2vec was used to construct dialogue embeddings for the tasks of domain categorization, semantic relatedness, and dialogue retrieval. Based on the idea of Dial2vec, an interesting future research direction would be to learn dialogue embeddings for the interlocutors (system and user) participating in successful versus unsuccessful dialogues, and by aggregating the embeddings of the interlocutors learn in turn dialogue embeddings for successful versus unsuccessful dialogues.

Acknowledgments

This work was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-20-2-0053. Many thanks to the anonymous reviewers for their helpful comments.

References

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task Gaussian Processes. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1798–1803, Doha, Qatar.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5016–5026, Brussels, Belgium.

A. Cervone, E. Gambi, G. Tortoreto, E. A. Stepanov, and G. Riccardi. 2018. Automatically predicting user ratings for conversational systems. In *Proc. of CLIC-It*.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 2 (Short Papers)*, pages 194–198, New Orleans, Louisiana, USA.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task Gaussian Processes: An application to machine translation quality estimation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 32–42, Sofia, Bulgaria.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54:755–810.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden Markov models. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 170–177, London, UK.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proc. of the Language Resources and Evaluation Conference (LREC)*, pages 422–428, Marseille, France (Online).

Matthew Frampton and Oliver Lemon. 2006. Learning more effective dialogue strategies using limited dialogue move features. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 185–192, Sydney, Australia.

Kallirroi Georgila. 2022. Comparing regression methods for dialogue system evaluation on a richly annotated corpus. In *Proc. of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial:DubDial)*, pages 81–93, Dublin, Ireland.

Kallirroi Georgila, Carla Gordon, Hyungtak Choi, Jill Boberg, Heesik Jeon, and David Traum. 2019. Toward low-cost automated evaluation metrics for Internet of Things dialogues. In *Proc. of the International Workshop on Spoken Dialogue Systems Technology (IWSDS), Lecture Notes in Electrical Engineering 579*, pages 161–175, Singapore.

Kallirroi Georgila, Carla Gordon, Volodymyr Yanov, and David Traum. 2020. Predicting ratings of real dialogue participants from artificial data and ratings of human dialogue observers. In *Proc. of the Language Resources and Evaluation Conference (LREC)*, pages 726–734, Marseille, France (Online).

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2005a. Learning user simulations for Information State Update dialogue systems. In *Proc. of Interspeech*, pages 893–896, Lisbon, Portugal.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of Interspeech*, pages 1065–1068, Pittsburgh, Pennsylvania, USA.

Kallirroi Georgila, Oliver Lemon, and James Henderson. 2005b. Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations. In *Proc. of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial:DIALOR)*, pages 61–68, Nancy, France.

Kallirroi Georgila, Oliver Lemon, James Henderson, and Johanna D. Moore. 2009. Automatic annotation of context and speech acts for dialogue corpora. *Journal of Natural Language Engineering*, 15(3):315–353.

Sarik Ghazarian, Yijia Shao, Rujun Han, Aram Galstyan, and Nanyun Peng. 2023. ACCENT: An automatic event commonsense evaluation metric for open-domain dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4398–4419, Toronto, Canada.

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 7789–7796, New York, New York, USA.

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.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2005. Hybrid reinforcement/supervised learning for dialogue policies from COMMUNICATOR data. In

Proc. of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 68–75, Edinburgh, UK.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed datasets. *Computational Linguistics*, 34(4):487–511.

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In *Proc. of the International Workshop on Dialogue Systems Technology (IWSDS), Lecture Notes in Computer Science 6392*, pages 48–60, Gotemba, Shizuoka, Japan.

Kate S. Hone and Robert Graham. 2000. Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI). *Journal of Natural Language Engineering*, 6(3-4):287–303.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Online and Punta Cana, Dominican Republic.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online.

Che Liu, Rui Wang, Junfeng Jiang, Yongbin Li, and Fei Huang. 2022. Dial2vec: Self-guided contrastive learning of unsupervised dialogue embeddings. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7272–7282, Abu Dhabi, United Arab Emirates.

Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael 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 *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2122–2132, Austin, Texas, USA.

Sarah McLeod, Ivana Kruijff-Korbayová, and Bernd Kiefer. 2019. Multi-task learning of system dialogue act selection for supervised pretraining of goal-oriented dialogue policies. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 411–417, Stockholm, Sweden.

Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, Samira Shaikh, David Traum, Yi-Ting Yeh, Zhou Yu, Yizhe Zhang, and Chen Zhang. 2022. Report from the NSF Future Directions Workshop on Automatic Evaluation of Dialog: Research Directions and Challenges. In *arXiv preprint arXiv:2203.10012*.

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 225–235, Online.

Shikib Mehri and Maxine Eskenazi. 2020b. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 681–707, Online.

Seyed Mahed Mousavi, Gabriel Roccabruna, Simone Alghisi, Massimo Rizzoli, Mirco Ravanelli, and Giuseppe Riccardi. 2024. Are LLMs Robust for spoken dialogues? In *Proc. of the International Workshop on Dialogue Systems Technology (IWSDS)*, Sapporo, Japan.

Taghi Paksima, Kallirroi Georgila, and Johanna D. Moore. 2009. Evaluating the effectiveness of information presentation in a full end-to-end dialogue system. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 1–10, London, UK.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China.

Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 45–54, Lisbon, Portugal.

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.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. In *arXiv preprint arXiv:2103.15316*.

Stefan Ultes and Wolfgang Minker. 2014. Interaction quality estimation in spoken dialogue systems using hybrid-HMMs. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 208–217, Philadelphia, Pennsylvania, USA.

M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, and S. Whittaker. 2001a. DARPA Communicator dialog travel planning systems: the June 2000 data collection. In *Proc. of Interspeech/Eurospeech*, pages 1371–1374, Aalborg, Denmark.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Journal of Natural Language Engineering*, 6(3-4):363–377.

Marilyn Walker and Rebecca Passoneau. 2001. DATE: A dialogue act tagging scheme for evaluation of spoken dialogue systems. In *Proc. of the Human Language Technologies Conference*, pages 1–8, San Diego, California, USA.

Marilyn Walker, Alex Rudnicky, John Aberdeen, Elizabeth Owen Bratt, John Garofolo, Helen Hastie, Audrey Le, Bryan Pellom, Alex Potamianos, Rebecca Passonneau, Rashmi Prasad, Salim Roukos, Greg Sanders, Stephanie Seneff, and Dave Stallard. 2002. DARPA communicator evaluation: Progress from 2000 to 2001. In *Proc. of Interspeech/ICSLP*, pages 273–276, Denver, Colorado, USA.

Marilyn A. Walker, Rebecca Passoneau, and Julie E. Boland. 2001b. Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 515–522, Toulouse, France.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. In *arXiv preprint arXiv:2212.03533*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proc. of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Appendix

SYSTEM UTTERANCE 1	
System output	what trip would you like to take?
System speech act	request_info
System task	top_level_trip
USER UTTERANCE 1	
ASR output	i'd like to go from atlanta georgia to london england
User transcription	i'd like to go from atlanta georgia to london england
User speech act	[provide_info,provide_info]
User task	[orig_city,dest_city]
SYSTEM UTTERANCE 2	
System output	flying from atlanta to london
System speech act	implicit_confirm
System task	orig_dest_city
SYSTEM UTTERANCE 3	
System output	what day are you leaving atlanta?
System speech act	request_info
System task	depart_arrive_date
USER UTTERANCE 2	
ASR output	september twenty fourth
User transcription	september twenty fourth
User speech act	[provide_info]
User task	[depart_date]
SYSTEM UTTERANCE 4	
System output	om monday, september twenty fourth
System speech act	implicit_confirm
System task	depart_arrive_date
SYSTEM UTTERANCE 5	
System output	about what time do you want to leave?
System speech act	request_info
System task	depart_arrive_time
USER UTTERANCE 3	
ASR output	early evening
User transcription	early evening
User speech act	[provide_info]
User task	[depart_time]
SYSTEM UTTERANCE 6	
System output	leaving in the early evening
System speech act	implicit_confirm
System task	depart_arrive_time
SYSTEM UTTERANCE 7	
System output	please stand by while i look up the flight schedules on the web
System speech act	status_report
System task	retrieval
SYSTEM UTTERANCE 8	
System output	ok, got them
System speech act	status_report
System task	retrieval
SYSTEM UTTERANCE 9	
System output	i can only get you a flight in the early evening on united
System speech act	present_info
System task	flight

Figure 1: An example dialogue excerpt showing both the ASR output and the transcription of the user's speech, and speech act and task annotations for both the system and the user. User speech act and task annotations, and dialogue context information are computed from the ASR output.

DIALOGUE LEVEL
Turn: user
TurnStartTime: 990207520.875
TurnEndTime: 990207525.562
TurnNumber: 3
Speaker: user
UtteranceStartTime: 990207520.875
UtteranceEndTime: 990207525.562
UtteranceNumber: 3
DialogueActType: user
ConvDomain: about_task
SpeechAct: [provide_info]
AsrInput: <date_time>early evening</date_time>
TransInput: <date_time>early evening</date_time>
Output:
TASK LEVEL
Task: [depart_time]
FilledSlot: [depart_time]
FilledSlotValue: [early evening]
GroundedSlot: [depart_date]
LOW LEVEL
WordErrorRatenoins: 0.00
WordErrorRate: 0.00
SentenceErrorRate: 0.00
KeyWordErrorRate: 0.0
ComputeErrorRatesReturnValue: 0
HISTORY LEVEL
FilledSlotsStatus: [dest_city],[orig_city],[depart_date],[depart_time]
FilledSlotsValuesStatus: [london england],[atlanta georgia],[september twenty fourth],[early evening]
GroundedSlotsStatus: [],[dest_city],[orig_city],[depart_date]
SpeechActsHist: request_info,[provide_info,provide_info],implicit_confirm,request_info,[provide_info],implicit_confirm, request_info,[provide_info]
TasksHist: top_level_trip,[orig_city,dest_city],orig_dest_city,depart_arrive_date,[depart_date],depart_arrive_date, depart_arrive_time,[depart_time]
FilledSlotsHist: [orig_city,dest_city],[depart_date],[depart_time]
FilledSlotsValuesHist: [atlanta georgia,london england],[september twenty fourth],[early evening]
GroundedSlotsHist: [],[orig_city,dest_city],[depart_date]

Figure 2: An example dialogue state generated after user utterance 3 in Figure 1. Empty ('[]') values or 'null' values (not seen here) do not affect the accuracy of the slot values.

	TaskEase		**SysComEase**		**UserExp**		**SysBehExp**		**SysFutUse**	
	man	auto	man	auto	man	auto	man	auto	man	auto
Average of Word Embeddings										
bert-L1	1.259	1.28	**1.163**	1.174	**1.287**	1.303	1.291	1.324	**1.363**	1.379
bert-L1-10-11	**1.254**	1.256	1.173	**1.167**	1.302	**1.289**	**1.283**	1.285	1.366	1.371
bert-L1-11-12	1.255	1.256	1.174	1.173	1.302	1.298	1.288	1.287	1.364	1.373
bert-L1-10-11-12	1.255	**1.253**	1.173	1.169	1.305	1.291	1.285	**1.28**	1.367	**1.37**
Output of [CLS] Token										
bert-L1	**1.276**	1.289	1.184	1.201	**1.305**	1.303	**1.296**	1.323	**1.375**	1.394
bert-L1-10-11	1.291	1.294	1.184	1.178	1.312	1.306	1.312	1.326	1.389	1.399
bert-L1-11-12	1.282	**1.287**	**1.178**	**1.176**	1.313	**1.302**	1.307	**1.304**	1.376	**1.384**
bert-L1-10-11-12	1.285	**1.287**	1.182	1.177	**1.305**	1.303	1.311	1.315	1.384	1.396

Table 5: RMSE values when calculating sentence embeddings as an average of BERT word embeddings versus using the output of the [CLS] token, based on the human transcriptions ('man') and the ASR outputs ('auto'). Domain-independent features are not included. The best value for each column for the output of the [CLS] token is shown in red and in bold. The average of BERT word embeddings always outperforms the output of the [CLS] token. **The best value for each column across both types of models is shown in black and in bold.**