Audio Embedding-Aware Dialogue Policy Learning

Asier López Zorrilla, M. Inés Torres, Heriberto Cuayáhuitl

***Abstract*—Following the success of Natural Language Processing (NLP) transformers pretrained via self-supervised learning, similar models have been proposed recently for speech processing such as Wav2Vec2, HuBERT and UniSpeech-SAT. An interesting yet unexplored area of application of these models is Spoken Dialogue Systems, where the users' audio signals are typically just mapped to word-level features derived from an Automatic Speech Recogniser (ASR), and then processed using NLP techniques to generate system responses. This paper reports a comprehensive comparison of dialogue policies trained using ASR-based transcriptions and extended with the aforementioned audio processing transformers in the DSTC2 task. Whilst our dialogue policies are trained with supervised and policy-based deep reinforcement learning, they are assessed using both automatic task completion metrics and a human evaluation. Our results reveal that using audio embeddings is more beneficial than detrimental in most of our trained dialogue policies, and that the benefits are stronger for supervised learning than reinforcement learning.**

***Index Terms*—Spoken Dialogue Systems, Audio Embeddings, Transformer Neural Networks, Deep Reinforcement Learning**

I. Introduction

SPOKEN Dialogue Systems (SDSs) aim at providing a convenient response to the user based on their speech's audio signal and dialogue context. Due to the advances in Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) and given the lack of effective tools for working directly with audio signals in this context, audio signals are often mapped into words first, and then NLP techniques are applied to understand the user and act accordingly. However, this approach is very dependent on the ASR providing a correct transcription, which might not be the case in noisy environments, or if the user is non-native or has an uncommon accent [1]. More importantly, it ignores important information in the users' speech, such as their emotional mood, prosody, or the noise level of the environment, which could be key to carry out a better dialogue strategy. This paper aims at including this information in end-to-end SDSs via cutting edge audio embeddings (also referred to as 'speech representations') as illustrated in Figure 1.

In fact, recent advances in self-supervised speech representation learning have opened the door to new ways of including acoustic information in Artificial Intelligence systems. Motivated by the success of similar approaches in NLP, these speech representations are learnt by transformer-based neural networks using unlabelled data only. They have demonstrated to be really powerful. State-of-the-art results (or close to that) can be obtained relatively easily in a variety of audio-related tasks using them, even if small domain specific data is available [2]–[4]. Preliminary experiments by [5] performed on the DSTC2 [6] spoken dialogue corpus indicate that, audio embeddings might actually encode useful acoustic information and exploit it to learn better dialogue policies, when combined with GPT-2 transformer [7] based neural dialogue policies.

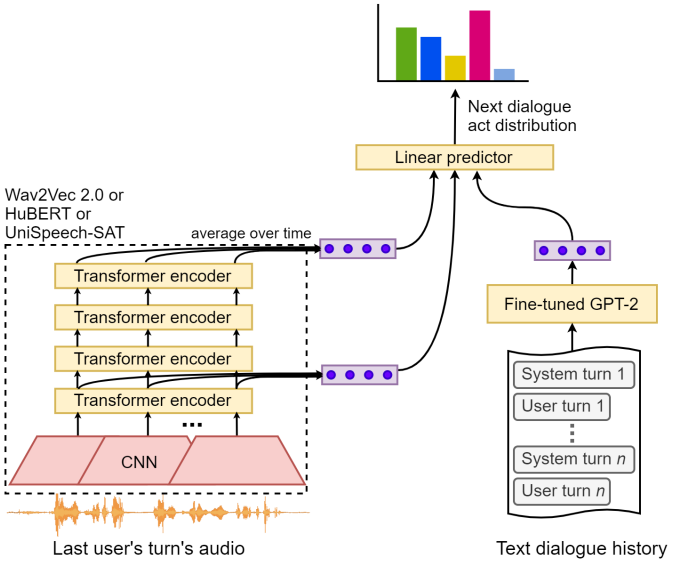


Fig. 1. Proposed dialogue manager architecture using audio-textual features, see Section III for further details.

This article extends the previous work above to further validate and, even more importantly, understand the effects of including speech representations in SDSs. First, we provide a substantially larger experimentation. We compare three of the latest audio embedding models (Wav2Vec2 [2], HuBERT [8] and UniSpeech-SAT [9]) and two different methodologies to extract the speech representations from them. Second, we analyse the consequences of adding audio embeddings to dialogue policies in a number of conditions: (a) combined with text representations obtained from two ASRs' output (of different qualities) and manual transcription, and (b) trained with three learning algorithms a number of times to extract statistically meaningful results. Third, we validate the results obtained with automatic metrics via a human evaluation. Last, we study differences in the behaviour and performance of policies that use and do not use acoustic information.

Consequently, we identify under which conditions audio embeddings help to learn better dialogue policies: they help the most with noisy ASRs, especially when the policies are learnt via Supervised Learning. Whilst speech representations allow a better user understanding in many occasions (e.g. identify what kind of information is being requested), they are also able to indicate the system that a turn has been noisy and that the ASR transcription might not be very reliable in some cases.

Asier López Zorrilla and M. Inés Torres are with the University of the Basque Country UPV/EHU, Spain (e-mails: {asier.lopezz, manes.torres}@ehu.eus). Heriberto Cuayáhuitl is with the University of Lincoln, U.K. (e-mail: HCuayahuitl@lincoln.ac.uk).

We have also found that the improvements are higher when learning policies via Supervised Learning (SL) as opposed to Reinforcement Learning (RL), probably because RL policies adapt better to the uncertainty in the ASR output.

The rest of the paper describes related works in Section II, our proposed approach for audio-based policy learning in Section III, our experimental framework (corpus, simulation pipeline, evaluation metrics and learning algorithms) in Section IV, experimental results and analysis (automatic evaluation, audio embedding comparison, human evaluation and a manual inspection of the models) in Section V, and Section VI presents our conclusions.

II. Related Work

Recent works on dialogue management have focused on improving different aspects of text-based models. For instance, one very interesting research area is open domain response generation with common-sense reasoning. It aims at providing dialogue models with general knowledge, which can enhance user understanding and lead to more diverse and informative response generation [10], [11]. Furthermore, it can also serve as a tool to model long-term dialogue goals [12], which is a very novel trend on dialogue modelling [13]; and even for dialogue emotion recognition [14], [15], another contemporary research topic [16]. On the other hand, latent-variable models have also been adopted to build open domain dialogue systems that produce more diverse [17] or contextually coherent [18] responses. Moreover, these models have also attracted attention for task-oriented dialogue management, where they have been used for joint state tracking and dialogue response generation [19], and also combined with pretrained transformer language models [20]. Although most works focus on either open domain or task-oriented dialogue management, there are also proposals to fuse both modalities [21]. In either case, all these works describe text-based dialogue systems only.

As for SDSs, the inconveniences caused by relying only on ASR outputs to make decisions have been previously treated in different ways. Some classical approaches to deal with this problem have focused on extracting as much information as possible from the ASR at hand. For example, a conventional methodology to build more robust SDSs consists of processing the top N hypotheses of the ASR rather than just the main output [22]. Some other alternatives make decisions based on ASR word confidence scores [23] or word confusion networks [24], which were proposed around two decades ago and are still in use nowadays [25] in SDSs and in Spoken Language Understanding [26], [27]. In the same vein and though hard to scale up, POMDP-based dialogue managers [28] were developed to cope with the uncertainties related to SDSs, including ASR outputs. Other efforts to include information present in the users' audio signals but absent in the ASR transcription can be found in the area of emotion aware dialogue systems [29], [30]. This kind of systems often include a module devoted to emotion recognition from audio, whose output is employed by the dialogue manager in the decision making step. However, none of the methods presented in these works explicitly process speech representations and make decisions based on them.

Closer to our work, we can find the research area of end-to-end spoken language understanding, where an audio is mapped into semantic labels directly. The encoder-decoder approach was the first way to tackle this problem [31]. Lately Wav2Vec2, one of the audio embedding networks that we use, has also been used to this end [4], showing the potential of this transformer network. But these studies focus on classifying audio signals, not on making decisions based on them.

The number of previous works describing dialogue systems that process the users' audio directly (without an ASR) is rather scarce. [32] and [33] present sequence-to-sequence models that process audio features in the context of audio visual scene-aware dialogue [34], [35], where the system has to answer a number of questions related to an audio visual scene. However, the audio to be analysed is not the users' audio, but the scenes' one. Closer to our approach is [36], who explore the inclusion of user sentiments in end-to-end dialogue systems. They train a dialogue policy that takes some audio features as input via SL. They found however that using the output of an external sentiment classifier worked better than the raw features. They also fine-tuned their dialogue manager using RL, but without including audio features.

The work presented in [37] is probably the closest to ours. They investigate the inclusion of users' audio in an LSTM-based encoder-decoder network for response generation in open-domain dialogue using SL only, not RL. To this end, they first train word-level audio embeddings in a response selection task and then concatenate those to traditional word embeddings to form the input to the network. In contrast to their work, our approach is simpler in terms of implementation, we use novel audio embedding networks which require no further pretraining, and we apply it to task-oriented dialogue data. In addition, the audio representations used in [37] were trained without taking into account the word order in dialogue turns, which could miss valuable audio information such as prosody.

Last, this study presents several novelties compared to our preliminary work [5]. We provide a larger experimentation and a much deeper analysis of how, when and why speech representations help to learn better dialogue policies. We compare three audio embedding models and two ways to integrate them as opposed to just one model and one method. Moreover, we use two speech recognisers instead of only one to analyse the influence of their quality. In addition, we present automatic results after more runs (6 instead of 1 in SL, and 30 instead of 7 in RL) to draw statistically stronger conclusions. More importantly, we further support those results with a human evaluation, and we study the differences between policies that use and do not use audio embeddings with a manual inspection of their behaviour and other metrics.

In this paper we also show that our dialogue policies can be easily fine-tuned using both SL and RL. Even though (deep) RL has established methodologies to train different types of dialogue systems [38]–[41], the user simulators employed in previous works only generate either words or dialogue acts—not audio. In contrast, our simulation pipeline is capable of providing audio-based dialogue turns. This is new in the area of dialogue policy learning. We refer the readers to [42] for a more-in-depth analysis of RL in spoken dialogue systems.

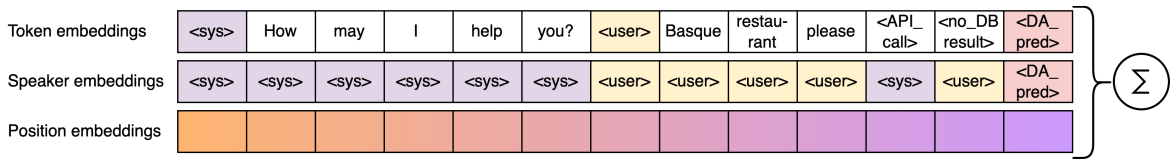


Fig. 2. Example set of inputs as part of the 'text dialogue history' in Figure 1 showing how it is represented in the proposed dialogue manager architecture.

III. Audio-Aware Dialogue Management

To measure the impact of including speech representations in dialogue managers, we compare policies that make decisions based on text-based dialogue history only against policies that process the history in the exact same way but also include audio embeddings to represent the last user's turn audio signal. We use a simple but contemporary architecture for our dialogue managers, see Figure 1. First, fixed-length representations from the dialogue history and the last user's turn audio are obtained with different architectures of transformer networks. Then, a linear predictor is used to compute the unnormalised probability distribution of the system's next dialogue act. Dialogue acts (as meanings of utterances) are used as output because they facilitate the integration of a user model and the policy optimisation with RL.

Text dialogue history. A pretrained GPT-2 transformer network [7] is used to process the text dialogue history and is fine-tuned during the training process. This approach has shown great success in both open domain [43] and goal oriented [44], [45] dialogue management. Each turn in the dialogue history is represented as raw text, i.e. no dialogue acts or named entities are used as input to the policies—to keep our approach as simple as possible. We employ a similar strategy to [46] to build the input to the GPT-2 network; three sequences of embeddings are added before being fed to the transformer, as represented in the example of Figure 2. First, the sequence of token embeddings is generated by concatenating the text of the turns in the dialogue history and processing it with a pretrained byte-level Byte-Pair-Encoding tokenizer (first row in Figure 2). Dialogue turns are separated with special tokens (<sys> or <user>) that indicate when system or user turns start. The second input sequence is made of segment/speaker embeddings, and is devoted to underline whose turn is (second row in Figure 2). The aforementioned <sys> and <user> tokens are used to this end. Last, the position embeddings provide the notion of order, as in most transformer networks [7] (third row in Figure 2). A <DA_pred> token is appended to the token and segment embeddings to indicate that the input sequences are complete and the dialogue act prediction should be made. In the example, the <API_call> and <no_DB_result> are used to log database searches in the dialogue history, as explained later in Section IV-B.

Last user's turn audio. We combine text with speech-based representations of the last user's dialogue turn. We compare three audio embedding models trained with self-supervised learning: Wav2Vec2 (W2V2 in short) [2], HuBERT [8] or UniSpeech-SAT (also referred to as 'UniS.') [9]. Even though each model is trained in a particular manner and has unique features, they all share a similar neural network architecture: a Convolutional Neural Net to digest the raw audio signal, and a multi-layer transformer on top of it to produce representations at different levels of abstraction, depending on the layer.

We keep the audio embedding models frozen during training, as recent studies [47] have shown that great success can be achieved in a number of tasks via linear predictions from the audio embeddings only, without any need of fine-tuning. Our three models employ a 12-layer transformer with a hidden size of 768. Thus, they output 768×12 values per time frame. They output 50 sets of vectors per second, and so the total is too high to directly perform predictions from them. In order to reduce the size of the representations to enhance our training procedure, we average the output of each layer in the time dimension, as suggested by [47] and [9]. We further reduce the dimensionality of the speech representations by selecting the output of a subset of layers. We do not just use the last layer because its representations might well not be the best [47], depending on the task. In Section V-B, we study which are the best layers for each model in our case.

Furthermore, we also explored the option of fine-tuning the audio transformers instead of keeping them frozen, in Section V-B. However, we obtained poorer results, and therefore the experiments presented in this paper are carried out without fine-tuning the audio embedding models.

IV. Experimental framework

A. Corpus

Recent dialogue corpora released in the last few years (e.g. MultiWOZ [48], STAR [49] or SGD [50]) have focused on text based dialogue modelling and none include audio. Some of the largest spoken dialogue corpora are the DSTC 1, 2 and 3 datasets [6], [51], [52]. Among these, the DSTC2 dataset [6] is by far the most used corpus for research in spoken dialogue technology and therefore we use this corpus in this work.

DSTC2 contains 3235 human-machine dialogues in the domain of restaurant search, acquired with three different dialogue systems. The corpus makes use of 8 slot types: *area*, *food type*, *restaurant name*, *price range*, *address*, *phone*, *post-code* and *signature*. All the slots are requestable, which means that users can ask for information about any of those. For example, they may ask about the address of a given restaurant. In contrast, only the first 4 slot types are informable, i.e. they can be used to constraint the restaurant search. This means that users can look for restaurants by area, food type or price range, but not by postcode for example. The corpus is split into three partitions: *train*, *dev* and *test*, which contain 1612, 506 and 1117 dialogues respectively. We merge the original *dev* and *test* partitions to build our testing data. In that way the training and test partitions have the same amount of data. This

is important because the module in charge of sampling user audios—the User Audio Sampler presented in Section IV-B—is sensitive to the number of available audio turns to sample from. This is done to prevent biased user behaviour.

B. Dialogue pipeline for simulations

We use dialogue simulations to evaluate the performance of our dialogue policies and to train them using RL. The simulation pipeline is illustrated in Figure 3. The remaining of this subsection describes its components.

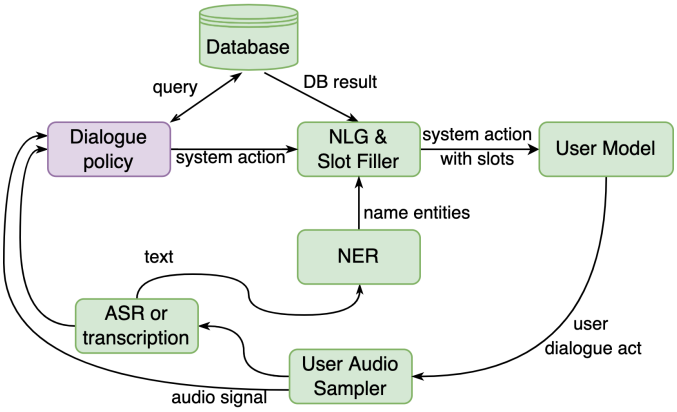


Fig. 3. Diagram of the simulation pipeline.

Dialogue Policy. The dialogue polices generated by the proposed dialogue manager in Figure 1 (Section III) use the publicly available *small* pretrained GPT-2 checkpoint and the so-called *Base* one for Wav2Vec2, HuBERT and UniSpeecch-SAT. Our dialogue states take into account a dialogue history truncated to the last 9 turns to avoid excessive GPU memory consumption. Our dialogue actions use composite dialogue acts to support multiple dialogue acts in a single dialogue turn (e.g. *confirm* | *area* + *request* | *food*), similarly to the procedure in DeepPavlov DSTC2 [53]. The reward function and learning algorithms used for selecting the best dialogue act in each state are described in subsection IV-D.

Named Entity Recogniser. Since our dialogue policies output dialogue acts containing one or a few slots, we use a Named Entity Recogniser (NER) to extract named entities from user turns to fill the slots of the dialogue acts. Our NER component, based on fuzzy matching, is a slightly improved version of DeepPavlov's NER for this task.

Database. Although no database was released as part of DSTC2, database calls can be inferred from the data to form a large enough dataset to perform dialogue simulations with it. Our policies are thus able to make database queries. In order to log this activity in the text dialogue history, every time a database query is made, the token `<API_call>` is added to the dialogue history. If the query is successful, the token `<DB_result>` is added. The token `<no_DB_result>` is concatenated otherwise. When multiple restaurants are retrieved, only one is selected (randomly). Thus, all information that the system may provide in subsequent dialogue turns would correspond to that result. If new user constraints are detected and the dialogue manager makes a new successful API call, the information retrieved from that point onwards would correspond to the latest search result.

Slot Filler and NLG. We use a rule-based slot filler to select the slot values associated to a dialogue act. As a dialogue progresses, we keep track of the recognised named entities by the NER module and the output of database searches. Depending on the dialogue act, we fill the slots with the last values produced by the NER or database modules. Our Natural Language Generation (NLG) module produces text corresponding to the system turns given a pair of dialogue act and selected slots using predefined templates. Since the user model works at the dialogue act level, the generated text is only used to fill the dialogue history. The slot filler is also in charge of selecting the search criteria for the database searches, based on the last entities recognised from the user. For instance, in the previously shown example of Figure 2, the only recognised entity would be *Basque*. Therefore, the only condition in the consequent database search would be that the food type is Basque, and there would not be any constraint regarding the area or the price range.

User Model (UM). Our UM is based on Attributed Probabilistic Finite State Bi-Automata [54]–[56]. It is data-driven and works at the dialogue act level, and its goal is selected at the beginning of the simulations according to the goal probability distribution found in the corpus. We built a UM with the training data to learn RL policies, and a UM with the test data to evaluate the performance of all policies.

User Audio Sampler. Since audio signals need to be fed to the proposed dialogue policies, we employ the User Audio Sampler proposed in [5] to sample an audio turn from the corpus. First, it selects output candidates filtering the audios of dialogue turns labelled with the same dialogue acts and slots generated by the UM. In order to enhance the behaviour of the sampler, the sampling probability of each candidate is adjusted in terms of the turn number (of the simulated dialogue and the candidate) and the number of repetitions. The turn number adjustment is introduced because users may speak in different ways depending on the phase of the dialogue. Similarly, if the same dialogue act/slot combination has been used more than once in a dialogue, it is probably due to the system not understanding it correctly and requesting it again. Last, it may happen that the dialogue act/slot combination output by the UM is not present in the corpus. In this case no audio signal can be fed to the dialogue policy, and the simulation ends prematurely. This happens in 20% of the dialogues, but fortunately, the sampling errors occur in the very first user turn almost exclusively (96% of the times)—due to the constraints of the user goal not appearing in the dataset. This means that only very rarely (0.8% of the simulations) computation time is wasted without adverse effects.

ASR/Transcription. Our experiments use three types of textual inputs: manual transcriptions (TRS), and two automatic speech recognition systems of different quality. The noisiest ASR (ASR 1) is based on a Wav2Vec2 network, where the *wav2vec2-base-960h* checkpoint was fine-tuned using 960 hours of Librispeech [2]. It achieved a Character Error Rate (CER) of 24.5 in the DSTC2 corpus, and a Word Error Rate (WER) of 45.8. For the second and better performing ASR

(ASR 2) we chose the best English model provided in the Vosk toolkit[1], the *vosk-model-en-us-0.22* model. The CER and WER errors were lower for this ASR, 10.1 and 21.0 respectively.

C. Automatic evaluation metrics

Our dialogue policies are evaluated automatically via simulated dialogues using the test user model. Their performance is measured with three common task-completion metrics [55], [57]—bounded between 0 and 1.

User Request Score (URS) indicates whether the system answers to the user in focus. It is the ratio between user informs answering a user request and user requests. For example, this score is high if the system provides an address after the user has requested it. This metric does not take into account, however, whether that address is correct or not, i.e., if it corresponds to the restaurant they are talking about or not. Whenever the user does not explicitly request any information, this score is not computed. This typically happens when the system provides information without the user requesting it.

System Offered Valid Venue (SOVV) indicates the correctness of system informs. It is the ratio between system informs that satisfy the constraints of the user and the total informs.

Can't Help Score (CHS) is only computed in a small fraction of the dialogues, about 20% approximately. Sometimes the UM has unreachable goals; for example, a user may want to find a Basque restaurant in the south of town, but there is none. In that case, the system should inform that there is no way to find such a restaurant. This score is 1 if the system provides this information, and 0 otherwise.

For simplicity and completeness, we use a combination of the three scores, which we call *Evaluation score*. Instead of defining it as the average of the three scores, we perform a weighted average with a lower weight for URS because it is a simpler task in which all the policies achieve close to perfect (>0.95) results. Lowering the URS weight gives more importance to the other two scores, which differ more across policies. In this way, the evaluation score aims to reflect more clearly the differences between policies. It is defined as:

$$\text{Evaluation score} = 0.2 \cdot \text{URS} + 0.4 \cdot \text{SOVV} + 0.4 \cdot \text{CHS}.$$

If any of the three scores above is not computed, the weights of the remaining scores are increased proportionally to keep the score bounded between 0 and 1.

Last, we also report the results in terms of a SOVV-CHS combined score, which facilitates the comparison between automatic and human metrics (see Section V-C). This score is the average of CHS and SOVV when both are computed, and SOVV otherwise. It measures the accuracy of the system at providing the right restaurant or correctly informing that there is no option given the user's search constraints.

D. Experiments overview

We carry out four sets of experiments. First, in Section V-A we perform a detailed study of the effects of adding different audio embeddings to dialogue policies with different text inputs and with three learning algorithms. We report the results in terms of automatic metrics. Second, we compare different ways to add audio embeddings to dialogue policies, in Section V-B. Third, in Section V-C we perform a human evaluation to further validate the results obtained in the first experiment, particularly the models that benefited the most by the inclusion of audio embeddings. Last, in Section V-D, we inspect some of the resulting dialogue policies to further understand in which cases the decisions led by the audio part of the networks result in more successful dialogues.

1) Training procedure: The automatic metrics are computed and averaged after a number of independent training runs to provide statistically meaningful results. The procedure followed to train and evaluate the different SL and RL policies is as follows:

1) We start by training text-only baselines for each input type (2 ASRs and TRS). For each input type, we train 6 different models and provide average results. Each model is evaluated with 5K dialogues with the test UM.
2) For each text-only SL model, we train each audio embedding part 5 independent times using SL. Thus, 30 (6×5) models are trained for each input type and audio embedding transformer (Wav2Vec2, HuBERT and UniSpeech-SAT). The GPT-2 network is kept intact in this stage to directly measure the impact of audio embeddings. Each model is evaluated with 1K dialogues.
3) For each text-only SL model, run REINFORCE [58] and Actor-Critic [59] RL algorithms 5 times without including any speech representations. As a result, 30 models are trained per text input type and RL algorithm. Each model is evaluated with 1K dialogues.
4) Finally, re-train every output model of step two using REINFORCE and Actor-Critic and evaluate its performance with 1K dialogues. In this case, both the text and audio parts are trained jointly.

Thus, for each combination of learning algorithm, text input type, and audio embedding model or just text input, 30K test dialogues are obtained in total (6 models × 5K dialogues for the text-only baselines, 30 models × 1K dialogues otherwise). We also attempted training the RL policies from scratch without a SL baseline, but it was much harder to make them converge and the results were a lot poorer.

2) SL details: We use 4 epochs of SL training for the text only baselines, and 2 epochs when training the audio part only. A batch size of 4 is used throughout all the experiments, and the Cross Entropy loss at the dialogue act level is minimised using the Adam optimiser with a learning rate of 5e-5.

3) RL details: REINFORCE and Actor-Critic are policy gradient RL algorithms that learn a set of weights θ in order select action a in state s according to policy $\pi_\theta(a|s)$. Our reward function use dense rewards, since previous works suggest that stronger policies are learnt in comparison with sparse rewards [5]. The reward function is as follows:

$$R = \begin{cases} 100 \cdot \textit{score} - 50 \cdot (1 - \textit{score}) & \text{if end of dialogue,} \\ 50 \cdot \ \textit{score} - 25 \cdot (1 - \textit{score}) & \text{if } \textit{score} \text{ is updated,} \\ -0.1 & \text{otherwise,} \end{cases}$$

[1]https://alphacephei.com/vosk/models.

TABLE I
AVERAGED EVALUATION METRICS USING THE TEST UM AFTER SUPERVISED LEARNING (SL), REINFORCE AND ACTOR-CRITIC, WITH DIFFERENT TEXT INPUTS AND AUDIO EMBEDDING MODELS. THE POLICIES WITH RESULTS IN PURPLE WERE PART OF THE HUMAN EVALUATION.

	SL				REINFORCE				Actor-Critic			
	Text	+W2V2	+UniS.	+HuBERT	Text	+W2V2	+UniS.	+HuBERT	Text	+W2V2	+UniS.	+HuBERT
Evaluation score												
ASR 1	0.771	0.790*	0.792*	**0.795***	0.792	0.796	**0.805***	0.796	0.818	0.820	0.822	**0.823**
ASR 2	0.934	0.935	0.932	**0.937**	0.916	0.918	**0.920**	0.911	0.927	0.930	0.931	**0.934***
TRS	0.940	**0.951***	0.947*	0.948*	0.928	0.931	**0.932**	0.928	0.947	**0.953***	0.953	0.950
Cumulative reward												
ASR 1	83.2	89.5*	90.4*	**91.4***	97.3	100.5*	**101.4***	98.3	**109.2**	105.7	106.1	108.1
ASR 2	137.8	140.3*	141.1*	**141.6***	138.9	**141.0**	139.7	139.5	**169.1**	165.3	168.6	168.3
TRS	135.4	147.0*	**147.4***	146.1*	142.9	144.4	**146.3***	144.0	**183.9**	182.9	182.1	180.6
User Request Score (URS)												
ASR 1	0.945	0.962*	**0.975***	0.969*	0.958	0.964	**0.971***	0.967*	0.987	**0.988**	0.987	0.988
ASR 2	0.984	0.988*	**0.991***	0.991*	0.982	0.984	**0.988***	0.979	0.991	0.993	0.993	**0.993**
TRS	0.974	0.986*	**0.989***	0.987*	0.975	0.978	**0.981**	0.979	0.991	0.992	**0.992**	0.992
System Offered Valid Venue (SOVV)												
ASR 1	0.750	0.766*	0.762*	**0.768***	0.773	0.774	**0.783**	0.775	0.791	0.793	0.795	0.796
ASR 2	0.917	0.920	0.912	**0.921**	0.894	0.901	**0.902**	0.896	0.909	0.911	0.913	**0.917***
TRS	0.880	**0.938***	0.932*	0.935*	0.912	**0.919**	0.919	0.918	0.936	**0.943**	0.942	0.938
Can't Help Score (CHS)												
ASR 1	0.668	0.701*	**0.721***	0.703*	0.629	0.643	**0.651**	0.629	0.674	0.703*	**0.713***	0.695
ASR 2	0.967	**0.968**	0.965	0.966	**0.922**	0.906	0.905	0.895	0.940	0.942	**0.954***	0.950
TRS	**0.989**	0.988	0.978	0.983	**0.925**	0.915	0.915	0.887	0.943	0.963*	**0.965***	0.959*
SOVV-CHS combined score												
ASR 1	0.747	0.766*	0.766*	**0.768***	0.758	0.761	**0.770***	0.760	0.779	0.788	**0.791**	0.789
ASR 2	0.925	**0.928**	0.921	0.928	0.899	0.902	**0.903**	0.897	0.913	0.918	0.921	**0.924***
TRS	0.900	**0.945***	0.939*	0.942*	0.914	0.917	**0.917**	0.912	0.934	0.946	**0.964***	0.941

where $score$ is the evaluation score described in section IV-C, which provides intermediate rewards after every system turn—especially if there are any new components in any of the aforementioned task completion metrics.

A discount factor of 0.95 and the Adam optimiser were used with a learning rate of 5e-6 in all the RL experiments. In the case of the Actor-Critic algorithm, the Actor (policy) and the Critic (estimated value function) use separate networks initialised with the resulting weights after the SL stage, except the linear predictor. We experienced some convergence problems with the actor-critic algorithm, which were solved by implementing two separate losses (one for the actor and the other for the critic). In addition to that, we used gradient clipping to prevent gradient exploding.

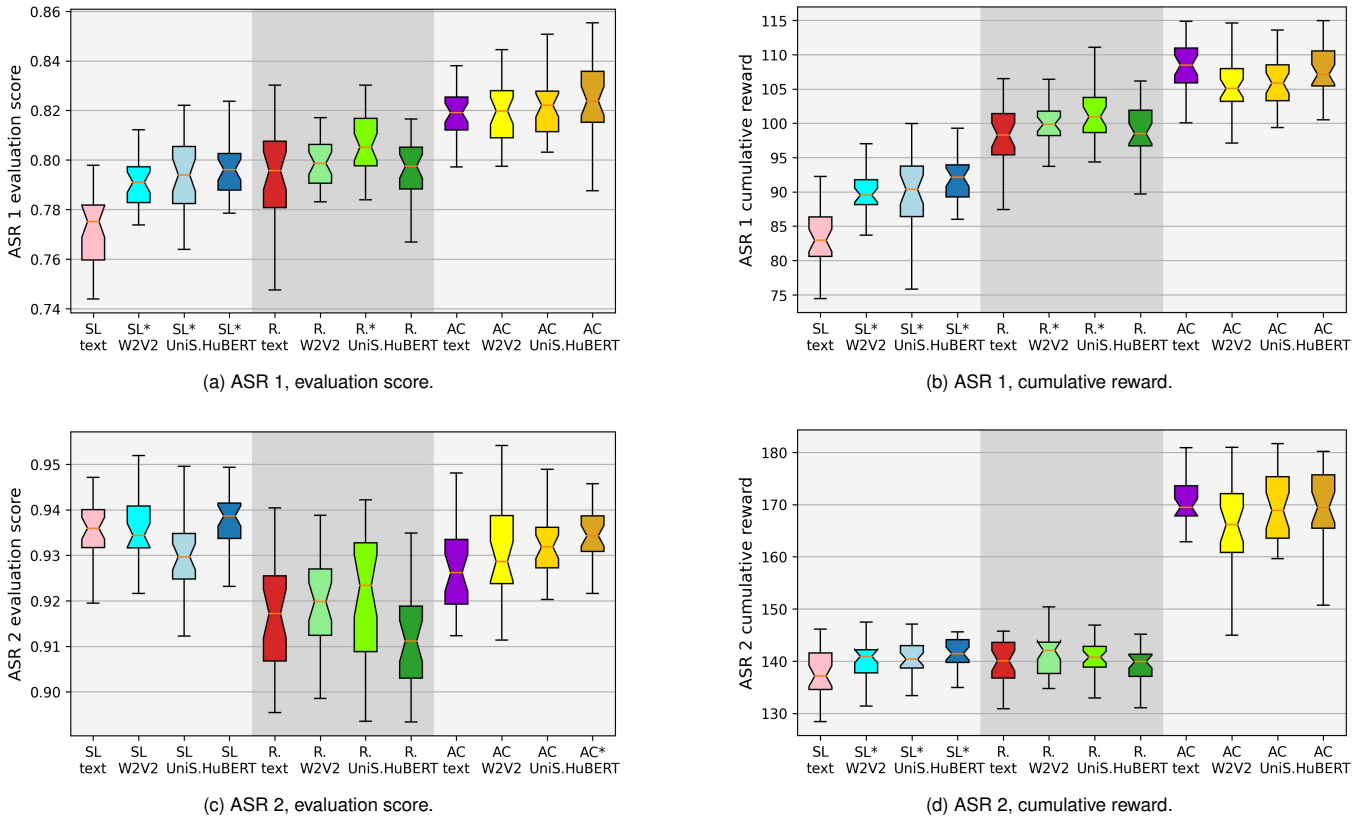
V. RESULTS

A. Automatic evaluation

Table I and Figure 4 show the performance of our learnt policies using the test UM according to evaluation scores and cumulative rewards. The bottom half of Table I shows results broken down into the three task completion metrics used to compute the evaluation score. The star symbol (*) indicates that values obtained using audio embeddings are significantly better than the ones corresponding to text only policies. More specifically, they mean that p-value$\leq$0.05 using the Welch's t-test, which tests whether two populations have equal means, without assuming equal variances. We use such a statistical test and p-value threshold in all the comparisons in this paper. In addition to the above, Table I shows results of our dialogue policies using manual transcriptions (see values in grey)—n.b. those policies do not compete against the ones processing ASR outputs because they do not make decisions based on noisy inputs. The values in purple correspond to policies to be rated in the human evaluation described in Section V-C.

These metrics show that including speech representations can help to learn better dialogues policies. But that depends on the learning algorithm and the quality of the ASR transcriptions. Regarding learning algorithms, SL policies clearly benefit the most by the inclusion of audio embeddings. This can be noted in SL policies including speech representations generated with either of the three audio embedding models, which significantly improve their performance. This is especially true for ASR 1, which suggests that audio embeddings are more beneficial in the case of noisier ASRs. Analysing the results per task completion metric, URS consistently improves significantly when using speech representations, SOVV in the case of ASR 1 and manual transcription, and CHS only with ASR 1-based text input.

RL-based policies do not benefit as much from audio embeddings as SL policies. The biggest improvement happens with REINFORCE and ASR 1 text inputs, where the evaluation score improves significantly by adding the speech information generated by UniSpeech-SAT, and so do the cumulative reward (with Wav2Vec2 and UniSpeech-SAT embeddings) and URS (with UniSpeech-SAT and HuBERT). The improvements in Actor-Critic policies are much more scarce. Despite some exceptions, adding audio embeddings does not seem to help much. In fact, the absolute better results in terms of cumulative reward are obtained by polices that do not use audio embeddings—but the differences between policies using and not using audio embeddings are not significant. This is in contrast with REINFORCE, where the better results were



(a) ASR 1, evaluation score. (b) ASR 1, cumulative reward.

(c) ASR 2, evaluation score. (d) ASR 2, cumulative reward.

Fig. 4. Performance of dialogue policies using a test UM after Supervised Learning, REINFORCE and Actor-Critic with different audio embedding models.

always obtained with policies processing audio embeddings in addition to the text input (with some exceptions in the CHS metric), even though the differences in performance were not always statistically significant.

Why do audio embeddings help more when learning policies via SL instead of RL, especially with the noisier ASR? A dialogue policy trained via SL mimics the behaviour of the system in the corpus. In contrast, policies trained via RL learn their behaviour through interaction with the UM. This allows them to develop alternative strategies to avoid misunderstandings and deal with the input's noise. In general, such strategies, particularly the Actor-Critic ones, are more conservative than the SL policies. Briefly, the RL policies perform more confirms and ask the user to repeat their constraints more (one way or another) before trying to look for a suitable venue. Thus, these polices are less sensitive to the input noise (whether environmental or introduced by the ASR), and therefore benefit less from the inclusion of speech representations. We discuss this topic further in Section V-C.

Figure 5 shows the evolution of the rolling average (over windows of 300 dialogues) of cumulative reward throughout the training process. It shows the learning curves corresponding to the ASR 1 text input and the UniSpeech-SAT audio embedding model—the same policies used in the human evaluation. Note that during SL the cumulative reward is not optimised explicitly. Instead, we minimise the dialogue act level cross entropy loss. But we show it for clarity and completeness. On the left-hand side of the figure, we can

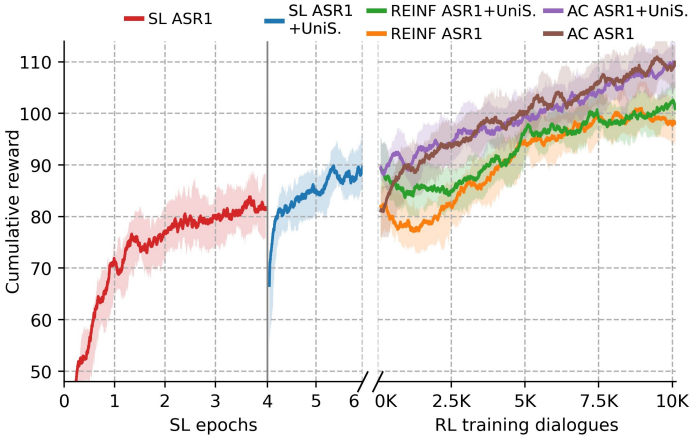


Fig. 5. Learning curves of dialogue policies with/without audio embeddings.

clearly see the impact of adding audio embeddings during SL (blue vs. red curve). After the first two epochs of SL with only text input, the performance of that policy only improves slightly, indicating that there is not much more room for improvement. The audio part of the dialogue manager is then added after the fourth epoch is finished. Since the audio part of the linear predictor is randomly initialised, a drop in performance can be seen at the beginning of the fifth epoch, the first with audio embeddings. Shortly after that drop, the benefits of adding speech representations appear. The policy recovers its performance and improves much quicker

TABLE II
PERFORMANCE COMPARISON OF AUDIO EMBEDDINGS IN OUR TASK BASED ON AVERAGE RESULTS FROM TABLE I.

	Wav2Vec2	UniSpeech-SAT	HuBERT
Evaluation score	0.892	**0.893**	0.892
Cumulative reward	135.1	**135.8**	135.4

than in the third or fourth epochs. At the end of epoch six, the cumulative reward was ∼10 points higher than epoch four. This is worth noting because the text part of the policy was kept untouched during the last two epochs. Thus, the improvements obtained in this period are due to the inclusion of audio embeddings only.

On the right hand side of Figure 5, we can see that RL policies on top of SL policies improve steadily their performance. But the differences between policies using and not using audio embeddings are largely reduced in RL. After several hundred dialogues, the differences in the Actor-Critic policies vanish—this effect is not so sudden with REINFORCE. It can be seen that REINFORCE is more unstable than Actor-Critic, and only in the middle of the training process the policies using ASR 1 output only level up, on average. In the second half of training, the policies combining this input with the UniSpeech-SAT embeddings keep improving, though slightly, whereas the text only policies seem to have converged.

B. Audio embedding comparison

1) Which audio embedding model is best? To answer this question Table II summarises the results shown in Table I, but averaged over the algorithms and input types. It can be seen that UniSpeech-SAT performs slightly better than HuBERT and Wav2Vec2 respectively, as one could expect from previous comparison studies between these networks [9], [47] applied to other tasks. But there are no statistically significant differences across these models in our task.

Nevertheless and even if the three models perform similarly in our task, the best way to extract the audio dense vectors from those models is unknown. To address that, we compared the cumulative reward obtained using the test UM when selecting different layers (or set of layers) from those models. As aforementioned, the output vectors of each selected layer were averaged over the time dimension to obtain easy-to-handle fixed-length vectors. We compared the output of each of the 12-layer transformers individually, and two combinations of 2 and 4 layers. The results are shown in Figure 6 with average rewards over three text input types on SL policies.

It is worth mentioning that every combination of output layer and embedding model outperforms the results obtained with text only. Besides, HuBERT and UniSpeech-SAT follow a similar pattern, which is reasonable since UniSpeech-SAT was trained inspired by the methodology used to create HuBERT [9]. In both cases the worst results are obtained with shallow layers, and the best ones with the last five layers. Moreover, the tested combinations of layers work quite well, presumably because they include layers that worked well individually in both cases. In contrast, Wav2Vec2 has a clear drop in the

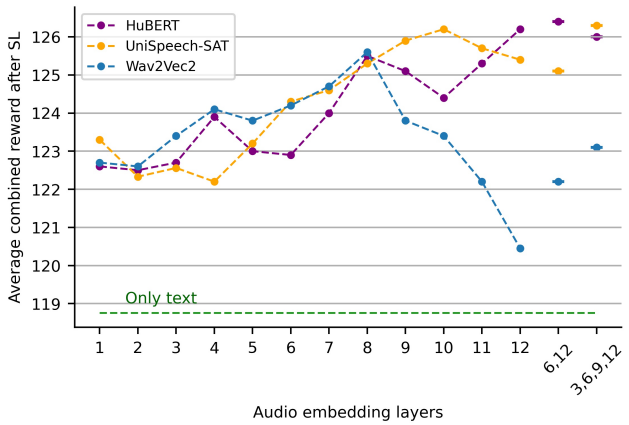


Fig. 6. Dialogue reward per neural layer of three audio embedding models.

TABLE III
PERFORMANCE OF SL DIALOGUE POLICIES USING THE TEST UM AFTER FINE-TUNING THE AUDIO EMBEDDING MODELS.

	Wav2Vec2		UniSpeech-SAT		HuBERT	
	Fine-tuned	Frozen	Fine-tuned	Frozen	Fine-tuned	Frozen
Evaluation score						
ASR 1	0.780	**0.790***	0.778	**0.792***	0.779	**0.795***
ASR 2	0.933	**0.935**	**0.933**	0.932	0.936	**0.937**
TRS	0.940	**0.951***	0.940	**0.947***	0.941	**0.948***
Cumulative reward						
ASR 1	85.6	**89.5***	85.0	**90.4***	85.5	**91.4***
ASR 2	137.8	**140.3***	137.0	**141.1***	138.0	**141.6***
TRS	140.3	**147.0***	140.4	**147.4***	140.8	**146.1***

performance from layer 9 onwards. Consequently, the combinations of layers we tested did not perform too well because they include the last layer. This comparison was performed at the beginning of our experimentation to select the output layers for each model according to Figure 6, and kept them unchanged during the rest of experiments presented in this work. Specifically, the 8th layer was selected for Wav2Vec2; the combination of the 6th and 12th layers for HuBERT and the set of the 3rd, 6th, 9th and 12th layers for UniSpeech-SAT.

2) How about fine-tuning the audio embedding models instead of keeping them frozen? Throughout this work so far, the audio embedding models have been kept frozen and only the linear predictors have been trained. As an additional experiment, also using SL only, we explored an alternative methodology that consists of using the last output vector (in the time dimension) of the last transformer layer as a summary of the whole input. Since this vector contains less information than the average over time, the transformer is fine-tuned while training. This way it should learn to include all the relevant information in that final vector. Table III shows performance results compared to those described in Table I. It can be observed that transformers with frozen layers outperform the fine-tuned ones. We hypothesise that this might be due to the following reasons: 1) the last output vector in the time dimension contains notoriously less information than the average over time, and a simple fine-tuning with small amount of data is not enough to train the network effectively to encode all the necessary information in that vector; and

2) fine-tuning involves the training of an exponentially larger amount of parameters, which could cause overfitting issues, especially due to the limited amount of training data.

3) Audio embeddings versus ASR confidence: We hypothesise that dialogue policies benefit from speech representations in two main ways. First, they allow a better semantic understanding of the user in many occasions, i.e. they can help to recognise what kind of information is being requested. This is supported by the experiments and examples we discuss below in Section V-D (Figure 9a). Second, audio embeddings also provide information about the intelligibility of audios, and thus should be able to inform the system when a turn has been noisy and the ASR transcription might not be very reliable. This information can also be introduced in the system via ASR confidence scores. In fact, if we substitute audio embeddings by the average and standard deviation of the character level ASR confidence (for ASR 1) in SL policies, an evaluation score of 0.780 can be obtained. This value is higher than for the text only baseline (0.771, Table I), but is still far from the best results obtained with audio embeddings (0.795, Table I). This suggests that speech representations not only include information about the potential ASR uncertainty, but also additional semantic information. Nevertheless, future work in other tasks and datasets is needed to confirm this result. Furthermore, learning paradigms such as POMDPs could be considered for future comparisons, since they were specially developed to deal with SDS-related uncertainties.

C. Human evaluation

We further validated the results obtained in Section V-A via a human evaluation. Since these results indicate that audio embeddings help the most with the noisiest ASR, ASR 1, we compared policies using ASR 1 transcriptions without and with audio embeddings. The latter are based on UniSpeech-SAT due to better performance, see Table II. We thus compare 3 pairs of policies: a policy processing only the ASR 1 versus another that also uses UniSpeech-SAT speech representations after training them via SL, REINFORCE and Actor-Critic. We do not compare other combinations because a human evaluation is much more costly than an automatic one.

Six judges (knowledgeable in the area of SDSs) evaluated 82 dialogues for each of our six policies—resulting in 492 dialogues per judge, 2952 dialogues in total. The evaluation was carried out using the Crowdzientzia platform [60]. Both the manual and ASR 1 transcriptions were shown in each of the users turn to allow the judges to assess the dialogues properly, similar to the examples in Figure 9, which we analyse in Section V-D. The judges were not aware of which policy had carried each dialogue to avoid any bias. After reading and analysing a dialogue, judges were asked to fill the multiple-choice 3-question questionnaire shown in Table IV, adapted from [61]. In the questionnaire, Q1 is related to the SOVV-CHS combined score described in Section IV-C, Q2 to the URS score, and Q3 is the most subjective question regarding dialogue naturalness. Table IV also shows the possible answers to each question, as well as their conversion to scalar ratings.

Table V shows averaged results of the human evaluation. We measured the inter-rater reliability with the Krippendorff's Alpha coefficient [62], with the interval metric [63] as the difference function. The values obtained are α_{Q1}=0.715, α_{Q2}=0.802, and α_{Q3}=0.742, which indicate a high agreement among the judges. Overall, the human evaluation supports and complements the conclusions drawn from the automatic evaluation. First, the greatest improvements come after SL, as expected from previous analysis: the policy using UniSpeech-SAT speech representations obtains a significantly higher score in the three questions, and on average. Second and in the case of policies trained via REINFORCE, the policy using audio embeddings improves too—but in this case the differences are not significant. Something similar happened with the automatic metrics, where only in some cases (with some audio embedding models) the cumulative reward or the evaluation score improved significantly. This indicates again that audio embeddings help in REINFORCE, but not always. Last and unsurprisingly, the gap is even narrower when using Actor-Critic as the learning algorithm. In this case, the differences are rather marginal, as happened when measuring their performance with automatic metrics.

Table V, on the other hand, also helps to gain a deeper insight into some other aspects of the policies, especially if we focus on Q3. While Q1 and Q2 focus on task completion, Q3 has more to do with how naturally they complete the task. If we rank the policies only in terms of Q1 and Q2, the ranking is very similar to the one obtained with the automatic evaluation:

- the Actor-Critic policies are the best,
- then REINFORCE with audio embeddings,
- then the SL policy with audio embeddings and the REINFORCE policy processing only text, and finally
- the SL policy based only on the ASR 1 output.

TABLE IV
QUESTIONNAIRE USED BY JUDGES IN THE HUMAN EVALUATION.

Q1:	The system offered a restaurant satisfying the user constraints, or correctly informed that there were no such restaurants. • Yes. (1) • No. (0)
Q2:	The system provided the information the user was looking for (phone number, post code, address...). • Yes. (1) • Partially. (0.5) • No. (0) • None–if there are no user requests.
Q3:	The conversation felt natural. • Strongly agree. (1) • Agree. (0.75) • Neither agree nor disagree. (0.5) • Disagree. (0.25) • Strongly disagree. (0)

TABLE V
HUMAN EVALUATION RESULTS. REINF STANDS FOR REINFORCE AND AC FOR ACTOR-CRITIC.

#	Algo.	Input	**Q1**	**Q2**	**Q3**	**Avg.**
1	SL	ASR1	0.656	0.848	0.535	0.629
2	SL	ASR1+UniS.	**0.760***	**0.902***	**0.601***	**0.716***
3	REINF	ASR1	0.730	0.892	**0.637**	0.716
4	REINF	ASR1+UniS.	**0.762**	**0.901**	0.632	**0.721**
5	AC	ASR1	0.761	**0.919**	**0.605**	0.718
6	AC	ASR1+UniS.	**0.789**	0.907	0.585	**0.719**

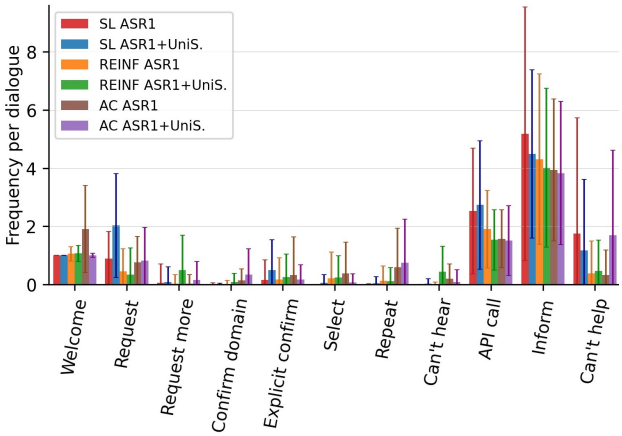


Fig. 7. Dialogue act histogram (with std) comparing six dialogue policies.

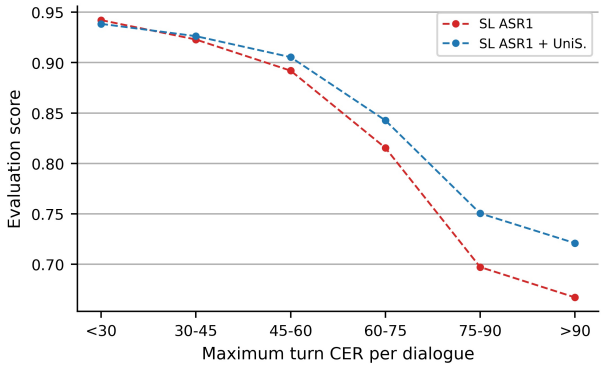


Fig. 8. Evaluation score of SL policies depending on the maximum turn CER (Character Error Rate) per dialogue.

But in terms of naturalness, Actor-Critic policies are worse than the REINFORCE ones, and AC+ASR1+UniS. is even less natural than SL+ASR1+UniS. This is due to Actor-Critic being a better RL algorithm in this task. The resulting policies thus exploit the UM as much as possible, leading to dialogue strategies that are not so natural. For example and since ASR 1 is noisy, Actor-Critic policies learn to extract as much information as possible from the UM, making it repeat the constraints multiple times so the API call is as right as possible. Some of these behaviours can be inferred from Figure 7, which shows the average and standard deviation of the frequency of each dialogue act per dialogue. Note that some dialogue acts have been grouped to make the figure clearer.

The fact that Actor-Critic dialogue policies confirm information provided by the user multiple times to reduce misunderstandings (even with text input only), suggests that other factors (such as amount of repetitions) should be considered in the employed reward function—or the use of learnt rewards. These suggestions could help to realise the full potential of audio embeddings for RL-based dialogue policies in the future.

D. Manual inspection

In this section, we aim at identifying how and when audio embeddings lead to better performance. To this end, we generated and analysed a number of simulated dialogues with policies that share the text processing part. Therefore, if they select a different dialogue act given the same context, it is only due to the audio embeddings. In many cases, dialogue strategies develop similarly whether they use policies with or without speech representations. This happens especially when the ASR transcriptions are more accurate. But in cases where the ASR output is poor, audio embeddings provide crucial information absent in the ASR transcription, allowing the policies perform better. This can be seen in Figure 8, where the correlation between the evaluation score of SL policies and the maximum turn CER per dialogue is plotted. The higher the CER, the more the policies benefit from audio embeddings. Figure 9 shows two simulated dialogues with poor ASR outputs where audio embeddings help to perform better actions. The first example (Figure 9a) is particularly representative, where we can see a typical conversation between the UM and the dialogue manager. The dialogue goes quite smoothly until a breaking point occurs when the UM requests the post code of the offered venue, in a rather noisy turn where the ASR 1 outputs "*hirst*". The text only policy (red box) performs an additional API call, and after some repetitions finally provides a post code, but it corresponds to the second restaurant it searched. Conversely, the policy processing the user's audio via the UniSpeech-SAT network (green box) is able to understand the user's intent even after the "*hirst*" turn, successfully providing the post code of the first venue it had offered. Thus, the dialogue ends in a much more natural manner. Additionally, Figure 9a shows the continuation of a policy that uses the ASR confidence as input (commented in Section V-B3). The low ASR confidence in the noisy turn prevents the policy from performing an API call, and it performs a safe inform instead. After another two post code requests, the system finally retrieves the desired information.

The second example compares the two REINFORCE policies judged in the human evaluation. Although both policies were trained on top of the same SL baseline, the two policies do not share completely the text processing part (because both the text and audio processing parts of the dialogue managers were trained jointly in RL experiments with audio embeddings). The example is still illustrative nonetheless. The initial user's message is not clear, due to a stuttering. After that, the text only REINFORCE policy performs an API call, but the found venue does not satisfy the user's requisites, because that first turn was not clear enough. Eventually the system corrects itself and finds a suitable venue, but the conversation is more messy than the continuation of the policy using audio embeddings. This one does not make an API call immediately, instead, it asks the user to repeat the sentence, because it is probably aware that the user turn was not understood. In addition to that, the text only policy is not able to understand that the user requests the phone after the ASR 1 transcription "*falg in*", and the user is forced to ask for it again. The policy with audio embeddings, on the other hand, is able to provide the address after the turn with ASR 1 output of "*aga*", which further consolidates the hypothesis that the speech representations lead to a better understanding.

Context

S: Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you?
U: looking for a restaurant in the east part of town
S: What kind of food would you like?
U: chinese
(ASR 1: try me)
S: <api_call>
S: sitar tandoori is a nice place in the east of town and the prices are expensive.
U: can i have the address of the venue
(ASR 1: and add the atters of devinon)
S: Sure, sitar tandoori is on 43 high street cherry hinton cherry hinton.
U: can i get the phone number
(ASR 1: figet the faw number)
S: The phone number of sitar tandoori is 01223 249955.
U: [noisy] whats the post code
(ASR 1: hirst)

Only ASR 1, SL policy continuation

S: <api_call>
S: pipasha restaurant is a nice place in the east of town and the prices are expensive.
U: [noise]
(ASR 1:)
S: Sure, pipasha restaurant is on newmarket road fen ditton.
U: post code
(ASR 1: i was going)
S: pipasha restaurant is a great restaurant.
U: and for good luck what is the post code
(ASR 1: and for good luck what is the postcod)
S: The post code of pipasha restaurant is c.b 5, 8 p.a.

ASR 1 + UniSpeech-SAT, SL policy continuation

S: The post code of sitar tandoori is c.b 1, 9 h.x.

ASR 1 + ASR confidence, SL policy continuation

S: sitar tandoori is a nice place in the east of town and the prices are expensive.
U: whats the post code
(ASR 1: the poetcan)
S: The phone number of sitar tandoori is 01223 249955.
U: post code of the venue
(ASR 1: postcoad of the venu)
S: The post code of sitar tandoori is c.b 1, 9 h.x.

(a) Three SL policies, with exactly the same text processing part.

Context

S: Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you?
U: [sluttering] yes indian food
(ASR 1: yes in the edfu)

Only ASR 1, REINFORCE policy continuation

S: <api_call>
S: hk fusion serves chinese food
U: indian food
(ASR 1: indico)
S: Sure, hk fusion is on 21 burleigh street city centre.
U: indian
S: <api_call>
S: curry garden serves indian food
U: phone number
(ASR 1: falg in)
S: Sure, curry garden is on 106 regent street city centre.
U: phone number.
(ASR 1: bo number)
S: The phone number of curry garden is 01223 302330.

ASR 1 + UniSpeech-SAT, REINFORCE policy continuation

S: Sorry I can't hear you.
U: indian
S: <api_call>
S: india house serves indian food
U: is there anything else
S: <api_call>
S: cocum serves indian food
U: can i have the address and phone number
(ASR 1: can at the actebundefore number)
S: the phone number of cocum is 01223 366668.
U: address
(ASR 1: aga)
S: Sure, cocum is on 71 castle street city centre.

(b) Two REINFORCE policies, based on the same text SL baseline.

Fig. 9. Sample dialogues where the policy including speech representations carries out a more successful dialogue. The context is the same for both policies.

TABLE VI
REQUEST REPETITIONS BY THE UM PER DIALOGUE WITH THE POLICIES USED IN THE HUMAN EVALUATION, AVERAGED OVER 1K DIALOGUES.

	ASR 1	+ UniSpeech-SAT
Supervised Learning	1.420	**1.258***
REINFORCE	1.411	**1.315**
Actor-Critic	1.183	**1.175**

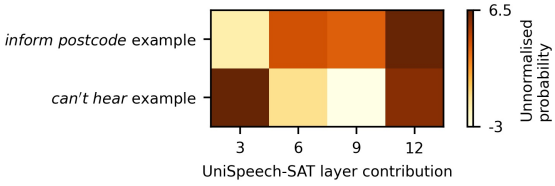


Fig. 10. Layer contribution to the decision taken in the example conversations.

Such a better understanding can be measured by the average number of requests by the UM per dialogue. As shown in Table VI, including audio embeddings leads to a lower number of requests per dialogue, especially after SL, where the difference is statistically significant. This is in line with the rest of the results obtained in our work: including audio embeddings helps the most when training the policies via SL.

Last but not least, we analyse the audio embedding layers' contribution to select the dialogue act in the examples in Figure 9. Since UniSpeech-SAT was used, the output of four layers is taken into consideration: the 3rd, 6th, 9th and 12th, as explained in Section V-B. We can easily compute how much each layer contributed to take the final decision. To this end, we take the output of the linear predictors that process the averaged embeddings of each layer, and select the value corresponding to the predicted dialogue act. This value is the layer's contribution to the unnormalised probability of taking that action. The contributions are shown in Figure 10.

In the *first example*, where the system correctly understands that the user is requesting a post code, the last layer has the biggest contribution, the two intermediate ones contribute less, and the shallower 3rd only barely influences the action. The

layer contribution is different in the *second example*, in which the system informs the user that it cannot hear correctly. In this case, the biggest contributions come from the 3rd and last layers. This makes sense since shallow layers operating closer to the audio signal are known to learn mostly speaker and environmental information, while the last layers contextualise more to learn content and semantic information [9]. Therefore and whilst a greater contribution of the intermediate and last neural layers are related to a better understanding, a greater contribution of the shallow neural layers can be interpreted as the system being aware of some anomalies at the signal level such as those exhibited by noise or stuttering (among others).

VI. Conclusion and future work

We present an in-depth study to analyse under which conditions speech representations (via audio embeddings) help to learn better dialogue policies in the context of the DSTC2 corpus. They help to understand the user better or to inform the system when the user might not be well understood—especially with the noisier ASR prone to providing inaccurate transcriptions. This effect is clearer when training the policies with supervised learning, because reinforcement learning algorithms are able to exploit the UM better and learn strategies to deal with the uncertainty in the text input more successfully.

We hypothesise that our approach could be very helpful in other demanding spoken dialogue tasks where the user is difficult to understand, even with very high-quality ASRs. Some examples include noisy industrial environments [64], SDSs integrated in cars [65], and also systems that interact with non-native users or users with strong local accents [1]. These are target domains for the dialogue community, and we hope that our findings can help to develop SDSs of higher quality in the future in these areas.

Finally, other potential future works could take advantage of the latest advances in Speech Synthesis (TTS) to continue our research with modern—and only text-based—dialogue corpora. The User Audio Sampler in our pipeline could be replaced by a high-quality TTS module. This has been successfully attempted in end-to-end Spoken Language Understanding recently [66]. Not only it would allow to test our approach on more challenging dialogue tasks, but it could also further validate our conclusions if user responses could be simulated taking into account specific background noise, local accents, backchannels, and/or emotions. In those cases, speech representations can provide additional information—absent in the ASR output—to boost the performance of SDSs.

Acknowledgements

This work has been partially funded by the Basque Government and by Ministerio Ciencia e Innovación, Next generation EU, under grants PRE_2017_1_0357 and PLEC2021-008171.

References

[1] D. Litman, H. Strik, and G. S. Lim, "Speech technologies and the assessment of second language speaking: Approaches, challenges, and opportunities," *Language Assessment Quarterly*, vol. 15, no. 3, 2018.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[3] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using wav2vec2.0 embeddings," *arXiv preprint arXiv:2104.03502*, 2021.

[4] S. Seo, D. Kwak, and B. Lee, "Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding," *arXiv preprint arXiv:2104.07253*, 2021.

[5] A. López Zorrilla, M. I. Torres, and H. Cuayáhuitl, "Audio embeddings help to learn better dialogue policies," in *ASRU*, 2021, pp. 962–968.

[6] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *SIGDIAL*, 2014.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[9] S. Chen, Y. Wu, C. Wang, Z. Chen, Z. Chen, S. Liu, J. Wu, Y. Qian, F. Wei, J. Li *et al.*, "Unispeech-sat: Universal speech representation learning with speaker aware pre-training," *arXiv preprint arXiv:2110.05752*, 2021.

[10] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu, "Diverse and informative dialogue generation with context-specific commonsense knowledge awareness," in *ACL*, 2020.

[11] ——, "Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended topic fact," in *IJCAI*, 2021.

[12] J. Ni, V. Pandelea, T. Young, H. Zhou, and E. Cambria, "Hitkg: Towards goal-oriented conversations via multi-hierarchy learning," in *AAAI*, vol. 36, no. 10, 2022, pp. 11 112–11 120.

[13] A. López Zorrilla and M. I. Torres, "A multilingual neural coaching model with enhanced long-term dialogue structure," *ACM Transactions on Interactive Intelligent Systems*, 2022.

[14] K. Zhang, Y. Li, J. Wang, E. Cambria, and X. Li, "Real-time video emotion recognition based on reinforcement learning and domain knowledge," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1034–1047, 2021.

[15] L. Zhu, G. Pergola, L. Gui, D. Zhou, and Y. He, "Topic-driven and knowledge-aware transformer for dialogue emotion detection," *arXiv preprint arXiv:2106.01071*, 2021.

[16] W. Li, W. Shao, S. Ji, and E. Cambria, "Bieru: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, pp. 73–82, 2022.

[17] W. Chen, Y. Gong, S. Wang, B. Yao, W. Qi, Z. Wei, X. Hu, B. Zhou, Y. Mao, W. Chen *et al.*, "Dialogved: A pre-trained latent variable encoder-decoder model for dialog response generation," *arXiv preprint arXiv:2204.13031*, 2022.

[18] J. Y. Lee, K. A. Lee, and W. S. Gan, "Improving contextual coherence in variational personalized and empathetic dialogue agents," in *ICASP*. IEEE, 2022, pp. 7052–7056.

[19] X. Xing and Z. Wang, "Probabilistic dialogue model combined with vae for task-oriented dialogue system," in *CECIT*. IEEE, 2021.

[20] H. Liu, Y. Cai, Z. Lin, Z. Ou, Y. Huang, and J. Feng, "Variational latent-state gpt for semi-supervised task-oriented dialog systems," *arXiv preprint arXiv:2109.04314*, 2021.

[21] T. Young, F. Xing, V. Pandelea, J. Ni, and E. Cambria, "Fusing task-oriented and open-domain dialogues in conversational agents," in *AAAI*, vol. 36, no. 10, 2022.

[22] Y. He and S. Young, "A data-driven spoken language understanding system," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*. IEEE, 2003, pp. 583–588.

[23] D. Hakkani-Tur and G. Riccardi, "A general algorithm for word graph matrix decomposition," in *ICASSP*, vol. 1. IEEE, 2003, pp. I–I.

[24] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[25] P. Swarup, R. Maas, S. Garimella, S. H. Mallidi, and B. Hoffmeister, "Improving asr confidence scores for alexa using acoustic and hypothesis embeddings," in *Interpeech*, 2019.

[26] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng, P. Molino, M. Namazifar, A. Papangelis, H. Williams, F. Bell *et al.*, "Joint contextual modeling for asr correction and language understanding," in *ICASP*. IEEE, 2020, pp. 6349–6353.

[27] K. Ganesan, P. Bamdev, A. Venugopal, A. Tushar *et al.*, "N-best asr transformer: Enhancing slu performance using multiple asr hypotheses," in *59th Annual Meeting of the ACL and the 11th IJCNLP*. ACL, 2021.

[28] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[29] J. Pittermann and A. Pittermann, "Integrating emotion recognition into an adaptive spoken language dialogue system," in *International Conference on Intelligent Environments*, vol. 1. IET, 2006, pp. 197–202.
[30] J. M. Olaso, A. Vázquez, L. Ben Letaifa, M. De Velasco, A. Mtibaa, M. A. Hmani, D. Petrovska-Delacrétaz, G. Chollet, C. Montenegro, A. López-Zorrilla *et al.*, "The empathic virtual coach: a demo," in *ICMI*, 2021, pp. 848–851.
[31] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *IEEE SLT Workshop*. IEEE, 2018, pp. 720–726.
[32] D. T. Nguyen, S. Sharma, H. Schulz, and L. E. Asri, "From film to video: Multi-turn question answering with multi-modal context," *CoRR*, vol. abs/1812.07023, 2018.
[33] H. Le, D. Sahoo, N. F. Chen, and S. C. H. Hoi, "Multimodal transformer networks for end-to-end video-grounded dialogue systems," in *ACL*, 2019.
[34] H. AlAmri, V. Cartillier, A. Das, J. Wang, A. Cherian, I. Essa, D. Batra, T. K. Marks, C. Hori, P. Anderson, S. Lee, and D. Parikh, "Audio visual scene-aware dialog," in *CVPR*, 2019.
[35] H. AlAmri, V. Cartillier, R. G. Lopes, A. Das, J. Wang, I. Essa, D. Batra, D. Parikh, A. Cherian, T. K. Marks, and C. Hori, "Audio visual scene-aware dialog challenge at DSTC7," *CoRR*, vol. abs/1806.00525, 2018.
[36] W. Shi and Z. Yu, "Sentiment adaptive end-to-end dialog systems," in *ACL*, 2018.
[37] T. Young, V. Pandelea, S. Poria, and E. Cambria, "Dialogue systems with audio context," *Neurocomputing*, vol. 388, 2020.
[38] I. Casanueva, P. Budzianowski, P. Su, S. Ultes, L. M. Rojas-Barahona, B. Tseng, and M. Gasic, "Feudal reinforcement learning for dialogue management in large domains," in *NAACL-HLT*, 2018.
[39] H. Cuayáhuitl, S. Yu, A. Williamson, and J. Carse, "Scaling up deep reinforcement learning for multi-domain dialogue systems," in *IJCNN*, 2017.
[40] R. Takanobu, H. Zhu, and M. Huang, "Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog," in *EMNLP-IJCNLP*, 2019.
[41] J. D. Williams and G. Zweig, "End-to-end lstm-based dialog control optimized with supervised and reinforcement learning," *CoRR*, vol. abs/1606.01269, 2016.
[42] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali, and E. Cambria, "A survey on deep reinforcement learning for audio-based applications," *Artificial Intelligence Review*, pp. 1–48, 2022.
[43] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith *et al.*, "Recipes for building an open-domain chatbot," *arXiv preprint arXiv:2004.13637*, 2020.
[44] D. Ham, J.-G. Lee, Y. Jang, and K.-E. Kim, "End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2," in *ACL*, 2020.
[45] A. López Zorrilla, "Towards structured closed-domain neural spoken dialogue systems," Ph.D. dissertation, UPV/EHU, 2022.
[46] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, "Transfertransfo: A transfer learning approach for neural network based conversational agents," *arXiv preprint arXiv:1901.08149*, 2019.
[47] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.
[48] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, "Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," *arXiv preprint arXiv:1810.00278*, 2018.
[49] J. E. Mosig, S. Mehri, and T. Kober, "Star: A schema-guided dialog dataset for transfer learning," *arXiv preprint arXiv:2010.11853*, 2020.
[50] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," in *AAAI*, vol. 34, no. 05, 2020.
[51] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in *SIGDIAL*, 2013.
[52] M. Henderson, B. Thomson, and J. D. Williams, "The third dialog state tracking challenge," in *SLT*, 2014.
[53] M. Burtsev, A. Seliverstov, R. Airapetyan, M. Arkhipov, D. Baymurzina, N. Bushkov, O. Gureenkova, T. Khakhulin, Y. Kuratov *et al.*, "Deeppavlov: Open-source library for dialogue systems," in *ACL*, 2018.
[54] M. Serras, M. I. Torres, and A. del Pozo, "Goal-conditioned user modeling for dialogue systems using stochastic bi-automata." in *ICPRAM*, 2019.
[55] M. Serras, "Contributions to attributed probabilistic finite state bi-automata for dialogue management," Ph.D. dissertation, UPV/EHU, 2021.
[56] M. I. Torres, "Stochastic bi-languages to model dialogs," in *FSMNLP*, 2013.
[57] F. Kreyssig, I. Casanueva, P. Budzianowski, and M. Gasic, "Neural user simulation for corpus-based policy optimisation of spoken dialogue systems," in *SIGdial*, 2018, pp. 60–69.
[58] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, 1992.
[59] V. R. Konda and J. N. Tsitsiklis, "Actor-citic agorithms," in *NIPS*, 1999.
[60] R. Justo, J. Alcaide, and M. Torres, "Crowdzientzia: Crowdsourcing for research and development," *IberSpeech*, pp. 403–410, 2016.
[61] S. Keizer, N. Braunschweiler, S. Stoyanchev, and R. Doddipatla, "Dialogue strategy adaptation to new action sets using multi-dimensional modelling," in *ASRU*, 2021.
[62] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage publications, 2018.
[63] ——, "Computing krippendorff's alpha-reliability." *Annenberg School for Communication Departmental Papers: Philadelphia*, 2011.
[64] C. Aceta, I. Fernández, and A. Soroa, "Kide4i: A generic semantics-based task-oriented dialogue system for human-machine interaction in industry 5.0," *Applied Sciences*, vol. 12, no. 3, p. 1192, 2022.
[65] M. Schmidt, D. Stier, S. Werner, and W. Minker, "Exploration and assessment of proactive use cases for an in-car voice assistant," *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pp. 148–155, 2019.
[66] L. Lugosch, B. H. Meyer, D. Nowrouzezahrai, and M. Ravanelli, "Using speech synthesis to train end-to-end spoken language understanding models," in *ICASSP*. IEEE, 2020, pp. 8499–8503.

VII. Biography Section

Asier López Zorrilla received his B.S. in electronic engineering (2016) and M.S. in computer science (2017) from the University of the Basque Country, where he is currently a PhD student (last year). His main research interests are speech and language processing (spoken dialogue systems in particular) and machine learning. He has explored many research ideas to improve end-to-end dialogue models during his M.S. and PhD thesis. For his B.S. thesis, he developed a neural ASR for Spanish, with an internship at Intelligence Voice Ltd. in London, U.K.

M. Inés Torres received her PhD in Physics from the UPV/EHU in 1990, including an internship at the CNET-Lanion (France). She was a visiting researcher at the Polytechnic University of Valencia (Spain), visiting Faculty in Carnegie Mellon University and visiting Professor at the University of California granted by the Fulbright program. She is currently a Full Professor of Computer Science at the UPV/EHU. Prof. Torres has a multi-disciplinary academic and industrial experience in the fields of Speech and Language Technologies conducted by data driven approaches. Her current interests are Human-Machine interaction, Speech processing, which covers Emotional Speech Identification, and Spoken Dialogue Systems. She has successfully coordinated the H2020 EMPATHIC project, leadered research contracts and national projects, among others.

Heriberto Cuayáhuitl is a Senior Lecturer in Computer Science at the University of Lincoln and member of the Lincoln Centre for Autonomous Systems (L-CAS). He received a PhD from the University of Edinburgh in 2009, and has an international research profile in academia and industry in the discipline of machine intelligence including (spoken) dialogue systems, (deep) machine learning, and (multimodal) robotics. He has published over 80 research papers in these areas, has been lead organiser of the international workshop on Machine Learning for Interactive Systems (MLIS), and guest editor of the journals ACM Transactions on Interactive Intelligent Systems, and Elsevier Computer Speech and Language. His work in industry has been carried out at SpeechWorks International Inc. (now Nuance Communications Inc.), the German Research Center for Artificial Intelligence (DFKI), and Samsung Research.