Addressing Objects and Their Relations:
The Conversational Entity Dialogue Model

Stefan Ultes, Paweł Budzianowski, Iñigo Casanueva, Lina Rojas-Barahona,
Bo-Hsiang Tseng, Yen-Chen Wu, Steve Young and Milica Gašić
Cambridge University Engineering Department
Cambridge, United Kingdom
{su259,pfb30,ic340,lmr46,bht26,ycw30,sjy11,mg436}@cam.ac.uk

Abstract

Statistical spoken dialogue systems usually rely on a single- or multi-domain dialogue model that is restricted in its capabilities of modelling complex dialogue structures, e.g., relations. In this work, we propose a novel dialogue model that is centred around entities and is able to model relations as well as multiple entities of the same type. We demonstrate in a prototype implementation benefits of relation modelling on the dialogue level and show that a trained policy using these relations outperforms the multi-domain baseline. Furthermore, we show that by modelling the relations on the dialogue level, the system is capable of processing relations present in the user input and even learns to address them in the system response.

1 Introduction

Data-driven statistical spoken dialogue systems (SDS) (Lemon and Pietquin, 2012; Young et al., 2013) are a promising approach for realizing spoken dialogue interaction between humans and machines. Up until now, these systems have successfully been applied to single- or multi-domain task-oriented dialogues (Su et al., 2017; Casanueva et al., 2017; Lison, 2011; Wang et al., 2014; Papangelis and Stylianou, 2017; Gašić et al., 2017; Budzianowski et al., 2017; Peng et al., 2017) where each dialogue is modelled as multiple independent single-domain sub-dialogues. However, this multi-domain dialogue model (MDDM) does not offer an intuitive way of representing multiple objects of the same type (e.g., multiple restaurants) or dynamic relations between these objects. To the best of our knowledge, neither problem has yet been addressed in statistical SDS research.

The goal of this paper is to propose a new dialogue model—the conversational entity dialogue model (CEDM)—which offers an intuitive way of modelling dialogues and complex dialogue structures inside the dialogue system. Inspired by Grosz (1978), the CEDM is centred around objects and relations instead of domains thus offering a fundamental change in how we think about statistical dialogue modelling. The CEDM allows

- to model dynamic relations directly, independently and persistently so that the relations may be addressed by the user *and* the system,

- the system to talk about multiple objects of the same type, e.g., multiple restaurants,

while still allowing feasible policy learning.

The remainder of the paper is organized as follows: after presenting a brief motivation and related work in Section 2, Section 3 presents background information on statistical SDSs. Section 4 contains the main contribution and describes the conversational entity dialogue model in detail. Looking at one aspect of the CEDM, the modelling of relations, Section 5 describes a prototype implementation and shows the benefits of the CEDM in experiments with a simulated user. Section 6 concludes the paper with a list of open questions which need to be addressed in future work.

2 Motivation and Related Work

To introduce the terminology that will be used in this work and to illustrate the necessity of adequate modelling of relations, Figure 1 shows an example dialogue about hotels and restaurants in Cambridge with the relation *in the same area*. Instead of talking about a sequence of domains, the system and the user talk about different objects and relations. Each part of the dialogue thus may be
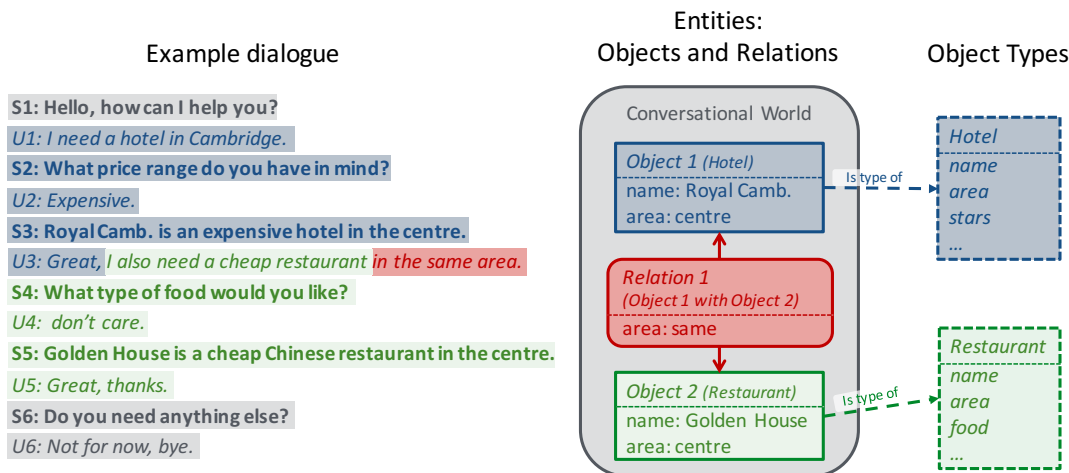
Figure 1: A dialogue between the system (S) and a user (U) about a restaurant and a hotel *in the same area* along with the mapping of fractions of the dialogue to the respective objects (of predefined types) and the relation. All objects and relations reside inside a conversational world.

mapped to an object or a relation in the conversational world or may be mapped to the world itself (grey). In the example, the first part (blue) is about *Object 1* of type *hotel*. When the focus shifts towards *Object 2* of type *restaurant* (green) at U3, the user also addresses the relation (red) *in the same area* between *Object 1* and *Object 2*.

Addressing a relation in this way could still be captured by the semantic interpretation of the user input as the information *area=centre* may be derived from the context. However, if the user said *I need a hotel and a restaurant in Cambridge in the same area* right in the beginning of the dialogue (U1), no context information would be available. To capture these dialogue structures, the dialogue model and the corresponding dialogue state must be able to represent them adequately.

The proposed CEDM achieves this by modelling state information about conversational entities instead of domains. More precisely, it models separate states about the objects (e.g., the hotel or restaurant) and the relations. Previous work on dialogue modelling already incorporated the idea of objects or entities to be the principal component of the dialogue state (Grosz, 1977; Bilange, 1991; Montoro et al., 2004; Xu and Seneff, 2010; Heinroth and Minker, 2013). However, these dialogue models are not based on statistical dialogue processing where a probability distribution over all dialogue states needs to be modelled and maintained. This additional complexity, though, cannot be incorporated in a straight-forward way into the proposed models. In contrast, the CEDM offers

a comprehensive and consistent way of modelling these probabilities by defining and maintaining entity-based states. Work on statistical dialogue state modelling (Young et al., 2010; Lee and Stent, 2016; Schulz et al., 2017) also contain a variant of objects but is still based on the MDDM thus not offering any mechanism to model multiple entities or relations between objects. Ramachandran and Ratnaparkhi (2015) proposed a belief tracking approach using relational trees. However, they only consider static relations present in the ontology and are not able to handle dynamic relations.

3 Statistical Spoken Dialogue Systems

Statistical SDS are model-based approaches[1] and usually assume a modular architecture (see Fig. 2). The problem of learning the next system action is framed as a partially-observable Markov decision process (POMDP) that accounts for the uncertainty inherent in spoken communication. This uncertainty is modelled in the belief state $b(s)$ representing a probability over all states s.

Reinforcement learning (RL) is used in such a sequential decision-making process where the decision-model (the policy π) is trained based on

---

[1]Model-free approaches like end-to-end generative networks (Serban et al., 2016; Li et al., 2016) have interesting properties (e.g., they only need text data for training) but they still seem to be limited in terms of dialogue structure complexity (not linguistic complexity) in cases where content from a structured knowledge base needs to be incorporated. Approaches where incorporating this information is learned along with the system responses based on dialogue data (Eric and Manning, 2017) seem hard to scale.
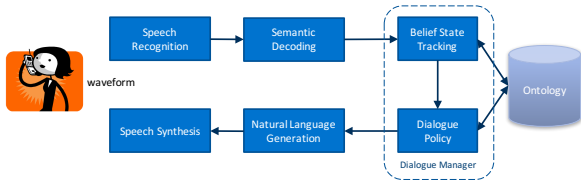
Figure 2: The modular statistical dialogue system architecture. The dialogue manager takes the semantic interpretation as input to track the belief state. The updated state is then used by the dialogue policy to decide on the next system action.

sample data and a potentially delayed objective signal (the reward r) (Sutton and Barto, 1998). The policy selects the next action $a \in A$ based on the current system belief state b to optimise the accumulated future reward R_t at time t:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \ . \tag{1}$$

Here, k denotes the number of future steps, γ a discount factor and r_τ the reward at time τ.

The Q-function models the expected accumulated future reward R_t when taking action a in belief state b and then following policy π:

$$Q^\pi(b, a) = E_\pi[R_t | b_t = b, a_t = a] \ . \tag{2}$$

For most real-world problems, finding the exact optimal Q-values is not feasible. Instead, RL algorithms have been proposed for dialogue policy learning based on approximating the Q-function directly or employing the policy gradient theorem (Williams and Young, 2006; Daubigney et al., 2012; Gašić and Young, 2014; Williams et al., 2017; Su et al., 2017; Casanueva et al., 2017; Papangelis and Stylianou, 2017).

Aside from the policy model, the dialogue model plays an important role: it defines the structure and internal links of the dialogue state as well as the system and user acts (i.e., the semantics the system can understand). Thus, the policy model is only able to learn system behaviour based on what is defined by the dialogue model. By defining the dialogue state, the dialogue model further represents an abstraction over the task ontology or knowledge base restricting the view on the information that is relevant so that the system is able to converse[2]. Most current dialogue models are built

around *domains* which encapsulate all relevant information as a section of the dialogue state that belongs to a given topic, e.g., finding a *restaurant* or *hotel*. However, the resulting flat state that is widely used (Williams et al., 2005; Young et al., 2010; Thomson and Young, 2010; Lee and Stent, 2016; Schulz et al., 2017, e.g.) is not intuitive to model complex dialogue structures like relations.

To overcome this limitation, we propose the conversational entity dialogue model which will be described in detail in the following section.

4 Conversational Entity Dialogue Model

The conversational entity dialogue model (CEDM) is proposed as an alternative way of statistical dialogue modelling having the concept of entities at the core of the model. Entities being objects or relations offer an intuitive way of modelling complex task-oriented dialogues.

4.1 Objects and Relations

Objects are entities of a certain object type (e.g., *Restaurant* or *Hotel*) where each type defines a set of attributes (see Fig. 1). This type definition matches the contents of the back-end knowledge base and thus the internal representation of real-world objects. This is similar to the definition of domains. In contrast to domains, though, this notion allows the modelling of multiple objects of the same type within a dialogue as well as the modelling of a type hierarchy which may be exploited during policy learning.

Relations are also entities that connect objects or attributes of objects. An example is shown in Figure 3: the two objects *obj1* and *obj2* of types *Hotel* and *Restaurant* respectively are connected through the attribute *area* with the *equals* relation.

Possible relations may directly be derived from the object type definitions, e.g., by allowing only connections for attributes that represent the same concepts like area. Note that these relations are dynamic relations that may be drawn between objects in a conversation. This is different to static relations which are often used in knowledge bases to describe how concepts relate to each other.

4.2 Conversational Entities in a Conversational World

A conversational entity is a virtual entity that exists in the context of the current conversation and is either a conversational object or a conversational
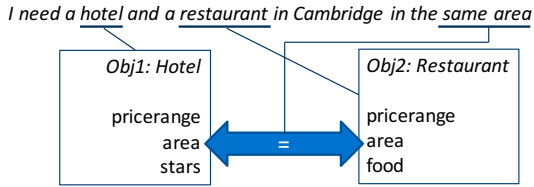
---

[2]Using the knowledge base directly to model the (noisy) dialogue state (Pragst et al., 2015; Meditskos et al., 2016) usually results in high access times.

Figure 3: Example mapping of a user utterance to two objects and one relation.
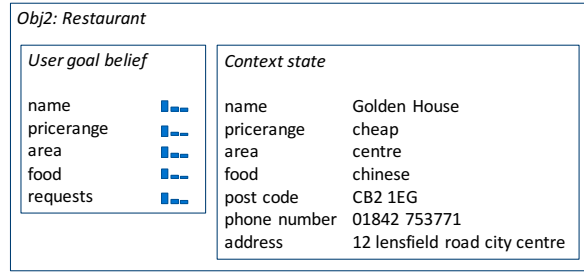

Figure 4: Example of a conversational entity representing object *obj2* of type *Restaurant*. The user goal belief models the search constraints the user has provided to the system and the context state represents the most recent real-world match offered by the system.
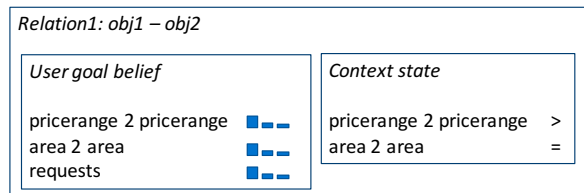

Figure 5: Example of the conversational entity *Relation1* between *obj1* and *obj2*. The user goal belief models the search constraints the user has provided to the system and the context state represents the relations based on the most recent real-world matches for both objects offered by the system.

relation. A conversational object may match a real-world entity but does not need to. In fact, the task of a goal-oriented dialogue is often to find a matching real-world entity based on the information acquired by the system during the dialogue. In the example dialogue (Fig. 1), matching entities have already been found for both objects. However, a conversational object exists independently of whether a matching real-world entity has been found yet or even exists.

Derived from the object type definition, a conversational object comprises an internal state that consists of the user goal belief s_u and the context state s_c as shown in the example in Figure 4. There, s_u is depicted using marginal probabilities for each slot (which is common in recent work on statistical SDS). While the user goal belief models the system's belief of what the user wants based on the user input, the context state models information that the system has shared with the user. In the example of Figure 4, the system has already offered a matching real-world object based on the user goal belief of the conversational object. If no offer has been made yet, the context state is empty.

The context state plays an important role as addressed relations usually refer to the object offered by the system instead of search constraints represented by the user goal belief. The context state further allows to relate to attributes that have not been mentioned in the dialogue.

One key aspect of the CEDM is that relations are also modelled as a conversational entity. Thus, these conversational relations also define a user goal belief and a context state as shown in Figure 5. The attributes of the relation are created out of the attributes of the objects they connect. In the given example, the attributes *area* and *pricerange* of the two objects are connected resulting in the relation attributes *area2area* and *pricerange2pricerange*. The values of these attributes are the actual relations, e.g., *equals* or *greater/less than*. Similar to the slot belief of con-

versational objects, each attribute is modelled with a marginal probability over all possible relations.

Assigning part of the belief state to the relations enables the system to specifically react to these relations and even to address them in a system utterance. Furthermore, if the context state of one of the related objects changes (e.g., because the user changed their mind), the relation may still persist.

Each conversational entity resides within a conversational world w (see Fig. 1) that defines the number of objects and the type of each object (relations may be derived from this) as well as general state information. This world may either be predefined or needs to be derived from the user input. In the latter case, the user input is usually noisy and an uncertainty needs to be modelled within the dialogue state. As this work focuses on relation modelling, a predefined conversational world is used leaving the uncertainty modelling of conversational worlds for future work.

4.3 Belief Tracking and Focus of Attention

The task of belief tracking is to update the probability distribution $b'(s)$ over the states s based on

the system action a, the observation o of the user input and the previous probability distribution b:

$$b'(s) = P(s|o, a, b) . \qquad (3)$$

With the additional complexity of the CEDM having an unknown number of entities in a conversational world, we propose to decompose the state s in the spirit of work by Williams et al. (2005). The belief update for each entity e is then defined as

$$b'_e(u, s_u, s_c, h_e) = P(u, s_u, s_c, h_e|a, o, b_e) , \qquad (4)$$

where s_u is the user goal state of entity e, s_c the context state of e, h_e the dialogue history of e and u the last user action[3].

The belief update for the world belief b_w is

$$b'_w(u, s_w, h_w) = P(u, s_w, h_w|a, o, b_w) , \qquad (5)$$

where s_w is the world state of world w, h_w the dialogue history and u the last user action.

This multi-part belief allows hierarchical dialogue processing on the world level and the entity level as depicted in Figure 6. Each level produces its own belief and based on that, the system is able to act on each level. On the world level, the system might produce general dialogue behaviour like greetings or engage in a dialogue to adequately identify the entity which is addressed by the user input. On the entity level, the system talks to the user to acquire information about the concrete entity the user is talking about, e.g., to find a matching entity in the knowledge base.

In addition to belief tracking, we would like to introduce another concept called focus of attention. Based on work by Grosz (1978), we define the current focus of attention $\mathcal{F}$ for each conversational world as a subset of conversational entities in this world $\mathcal{F} \subseteq W$. Hence, the task of focus tracking is to find the new set of conversational entities which is in the current focus of attention based on the user input and the updated belief state. Even though the concept of focus is not mandatory, it may be helpful when framing the reinforcement learning problem as it allows to limit the size of the input to the reinforcement learning algorithm as well as the number of actions available to the learning algorithm at a given time. Using $\mathcal{F}$ may also prevent the system from acting in parts of the belief state that are completely irrelevant to the current part of the conversation.

---

[3]In case of an unknown number of entities represented by a probability over worlds, the probability in Equation 4 needs to be extended to depend on the conversational world and needs to be multiplied by a probability over all worlds.

world level	world general behaviour	b_w
entity level	entity specific behaviour	b_e

Figure 6: The layered model of the CEDM with the respective components of the belief state.

4.4 The Conversational Entity vs. the Multi-Domain Dialogue Model

The functionality and the modelling possibilities of the proposed CEDM go beyond (and thus include) the possibilities of the multi-domain dialogue model (MDDM). To demonstrate this, we will outline how a dialogue using the MDDM may be modelled using the CEDM. The core concept *domain* of the MDDM may be mapped to one conversational object of a specific type where the slots of the domain are the attributes of the type. Since the number of domains is predefined, there is only one conversational world with a set number of conversational objects. Relations may not be modelled using the MDDM. Belief update is reduced to finding the right entity for the user input and updating its state. In the CEDM, the semantic decoding of user input includes the entity (or entity type) it refers to, which is similar to the topic tracker of the MDDM where the topic tracker also defines the domain the system acts in. Hence, the focus of attention will always contain only the entity that has been addressed by the user. By that, a policy for each conversational object (and thus object type) may be trained which is the same as the domain policies of the MDDM.

5 Relation Modelling Evaluation

To demonstrate the capabilities and benefits of the conversational entity dialogue model (CEDM), the aspect of relation modelling has been selected as it is a core concept of the CEDM. For this, we built upon the mapping to the multi-domain dialogue model (MDDM) as described in Section 4.4 and extend it with conversational relations. After a brief description of the model implementation, the experiments and their results are presented using two conversational objects of different types. Note that only the *equals* relation is considered here due to limitations of the marginal belief state model.

5.1 Model Implementation

To implement all relevant aspects of the CEDM, the publicly available open-source statistical dialogue system toolkit PyDial (Ultes et al., 2017) is

used which originally follows the MDDM.

The main challenge for policy implementation is to integrate both the state of the object in $\mathcal{F}$ as well as the states of all corresponding relations into the dialogue decision. To achieve this, a hierarchical policy model based on feudal reinforcement learning (Dayan and Hinton, 1993) has been implemented following the approach of Casanueva et al. (2018). For each object type, a master policy decides whether the next system action addresses a conversational relation or the conversational object. A respective sub-policy is then invoked in a second step where each object type and each relation type are modelled by an individual policy. Thus, the model decomposes the action selection problem to take account for the specificities of the object policy and relation policies respectively and is able to handle a variable number of relations. During training, all policies (master and sub-policies) receive the same reward signal.

Aside from the feudal RL architecture which seems to be intuitive for the proposed CEDM, the main problem is the handling of back-end database access. In the MDDM, each domain models all information which is necessary to do the data-base lookup. However, this is not possible in the CEDM as information from different conversational objects and relations need to be taken into account. One way of doing this is to apply a rule-based merging of the state of the conversational object in $\mathcal{F}$ with the states of all other conversational objects that are related through a conversational relation to form the focus state $\hat{b}$:

$$\hat{b}_s(v) = \frac{\sum_i w_i b_s^i(v)}{\sum_i w_i} \ , \qquad (6)$$

where s is the slot, v is the value, and b^i the belief of the i-th conversational entity involved in the merging process. $w_i = 1 - b_s^i(\emptyset)$ is the weight of the i-th conversational entity where $b_s^i(\emptyset)$ represents the probability where no information about slot s has yet been shared with the system. b^i either refers to the belief b^o of the conversational object o in $\mathcal{F}$ or to an already weighted belief $\tilde{b}^{o'}$ originating from the conversational relation $rel^{oo'}$ connecting conversational object o with o':

$$\tilde{b}_s^{o'}(v) = \begin{cases} rel_s^{oo'}(=) \cdot b_s^{o'}(v), & v \neq \emptyset \\ rel_s^{oo'}(=) \cdot b_s^{o'}(v) + rel_s^{oo'}(\emptyset), & v = \emptyset \end{cases}$$

where $b^{o'}$ is the belief of object o'. The relation probability rel is 0 if the slot s has no matching slot in o'. Please note that for $b_s^{o'}(v)$, even though we refer to the belief, the context state of o' is used instead if not empty. The focus state is used as input to the master policy as well as the sub-policy of the conversational object.

As an example, consider $b_s^o = [\emptyset : 0.3, v_1 : 0.7, v_2 : 0.0]$, $b_s^{o'} = [\emptyset : 0.2, v_1 : 0.0, v_2 : 0.8]$, and $rel_s^{oo'} = [\emptyset : 0.1, =: 0.9]$. This results in $\tilde{b}_s^{o'} = [\emptyset : 0.28, v_1 : 0.0, v_2 : 0.78]$ and $\hat{b}_s = [\emptyset : 0.29, v_1 : 0.35, v_2 : 0.36]$. This example also shows that conflicts which may exists between the state of the conversational object and the state defined by the relation are visible at this level. To help the policy to learn in this situation, an additional conflict bit is added to the focus belief state as input to the master policy.

The source code of the CEDM implementation is available at http://pydial.org/cedm.

5.2 Experimental Setup

To evaluate the relation modelling capabilities of the CEDM, the task of finding a hotel and a restaurant in Cambridge has been selected (corresponding to the *CamRestaurants* and *CamHotels* domains of PyDial). The corresponding conversational world consists of two conversational objects of types *hotel* and *restaurant* and one conversational relation. Based on the object type definitions, the conversational relation connects the slots *area* and *pricerange* of both objects. Using a simulated environment, the goals of the simulated user were generated so that at least one of these two slots is related (i.e., contains the same value).

To test the influence of the user addressing the relation instead of the correct value (e.g., "restaurant in the same area as the hotel" vs. "restaurant in the centre"), we have extended the simulated agenda-based user (Schatzmann and Young, 2009) with a probability r of the user addressing the relation instead of the value. The higher r, the more often the user addresses the relation. The user simulator is equipped with an additional error model to simulate the semantic error rate (SER) caused in a real system by the noisy speech channel.

For belief tracking, an extended version of the focus tracker (Henderson et al., 2014)—an effective rule-based tracker—was used for the conversational entities and the conversational world that also discounts probabilities if the respective value has been rejected by the user. As a simulated interaction is on the semantic level, no semantic de-

coder for the relations is necessary. For training and evaluation of the proposed framework, both the master policy and all sub-policies are modelled with the GP-SARSA algorithm (Gašić and Young, 2014). This is a value-based method that uses a Gaussian process to approximate the state-value function (Eq. 2). As it takes into account the uncertainty of the estimate, it is sample-efficient.

To compare the dialogue performance of the CEDM with the MDDM baseline, two experiments have been conducted. All dialogues follow the same structure: the user and the system first talk about one conversational object before moving on to the second object. As the user only addresses a relation to an object that has previously been part of the dialogue, relations are only relevant when talking about the second object. However, there are times where a relation has been addressed by the user before the goal of the first object changed which resulted in the addressed relation being wrong. This could only be resolved by the system by addressing the relation itself.

Experiment 1 In the first experiment, the influence of r on the dialogue performance is investigated in a controlled environment. Having a fixed order, only the feudal policy of the second object (where relations may occur), the *restaurant*, is learned. To avoid interfering effects of jointly learning both policies at the same time, the first object *hotel* uses a handcrafted policy.

Experiment 2 The second experiment focusses on the joint learning effects. Thus, the order of objects is alternated, all objects use the feudal policy model and are trained simultaneously.

5.3 Results

The experiments have been conducted based on the PyDial simulation environments Env. 1 and Env. 3 specified by Casanueva et al. (2017) where Env. 1 operates on a clean communication channel with an SER of 0% and Env. 3 simulates an SER of 15%. For each experiment, a policy for the respective object types was trained with 4,000 and tested with 1,000 dialogues. The reward was set to +30/+0 for success/failure and -1 for each turn with a maximum of 25 turns per object. The results were averaged over 5 different random seeds.

Experiment 1 As can be seen in Table 1 and Figure 7 on the left, the proposed CEDM with a feudal policy model is easily able to deal with relations addressed by the user for any relation prob-

ability r in both environments. Success rate and reward achieve similar results for all r. Only for very high r, a small reduction in performance is visible. This can be explained with the added complexity of the dialogue itself as well as the system actions that address the relations. A high relation probability for a slot requires the system to address either the relation or the slot value directly. Both actions may have similar or contradicting impact on the dialogue which makes it harder to learn a good policy. In Env. 3, the added noise results in minor fluctuations which may be expected.

In contrast, the baseline (the MDDM) is not able to handle the user addressing relations adequately for higher r: while for low r, the policy is able to compensate by requesting the respective information again, the performance drops at around $r = 0.5$. The reason why the performance of the baseline does not drop as much in Env. 3 as it does in Env. 1 is the way the simulated error model of the simulated user operates. By producing a 3-best-list of user inputs, the chance that the actual correct value is introduced as noise if a relation has originally been uttered is relatively high. As the n-best-list of Env. 1 has the length of one, this does not happen there.

The performance of the hand-crafted hotel policy was similar for all r in Env. 1 with $rew = 23.4, suc = 99.7\%$ and in Env. 3 with $rew = 20.1, suc = 94.5\%$.

Analysing the system actions of the dialogues of the CEDM shows that the system learns to address a relation in up to 28% of all dialogues for $r = 1.0$.

Example dialogues for Env. 1 are shown in Figures 8 and 9.

Experiment 2 The results shown in Table 1 and Figure 7 on the right show the performance of the conversational object policies when the respective object was the second one in the dialogue (where relations occur). Still, policies of both objects were trained in all dialogues. The effects of this added noise become visible in the results as they seem to be less stable. Furthermore, the overall performance for the *restaurant* policy drops a bit, but still shows the same characteristics as in Experiment 1. Learning a *hotel* policy results in worse overall performance (which matches the literature) and in cases where a relation is involved.

The performance of the policy of the first object was similar for all r where the restaurant policy achieved $rew = 21.5, suc = 95.4\%$ and the hotel

Table 1: Reward and success rate of both experiments for different relation probabilities r comparing the proposed CEDM to the MDDM baseline. The measures only show the performance of the second object in the dialogue where the relation is relevant. All results are computed after 4,000/1,000 train/test dialogues and averaged over 5 trials with different random seeds. **Bold** indicates statistically significant outperformance ($p < .05$), *italic* indicates no statistically significant difference.

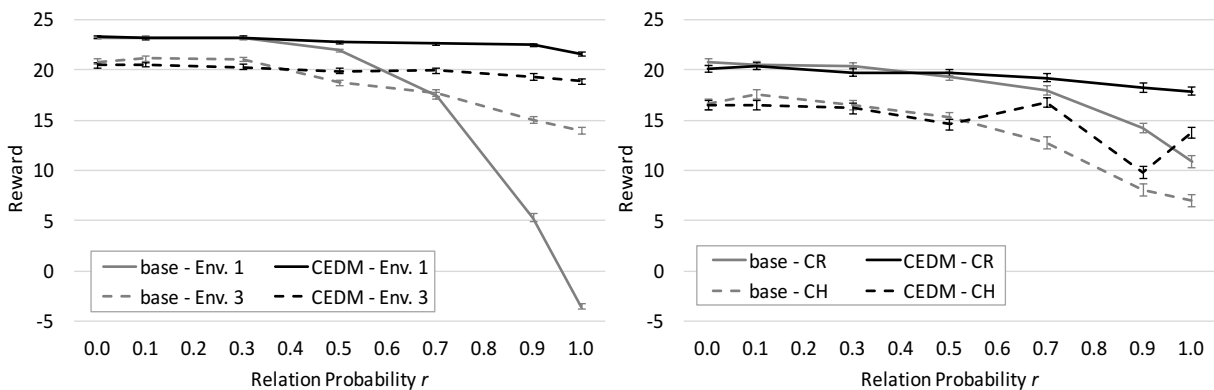| | Experiment 1 | | | | | | | | Experiment 2 | | | | | | | |
	Restaurant - Env. 1				Restaurant - Env. 3				Restaurant - Env. 3				Hotel - Env. 3			
	CEDM		base		CEDM		base		CEDM		base		CEDM		base	
r	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.
0.0	**23.3**	99.3%	23.2	**99.6%**	20.4	94.3%	**20.8**	**96.6%**	20.1	95.0%	**20.7**	**96.1%**	*16.5*	**86.7%**	*16.6*	85.8%
0.1	23.1	**99.5%**	23.2	99.1%	20.5	94.7%	**21.1**	**96.5%**	*20.3*	*94.4%*	*20.4*	*94.4%*	16.5	86.4%	17.5	89.0%
0.3	23.2	**99.5%**	23.1	99.0%	20.2	93.6%	**21.0**	**95.8%**	19.7	93.6%	**20.4**	**95.0%**	*16.2*	85.5%	*16.5*	87.1%
0.5	22.8	**99.6%**	21.9	96.2%	**19.8**	**92.8%**	18.7	89.7%	**19.7**	**92.5%**	19.3	92.0%	14.6	80.8%	15.2	**82.4%**
0.7	22.6	**99.2%**	17.4	82.3%	**19.9**	**92.9%**	17.7	86.8%	**19.2**	**91.6%**	17.9	87.9%	**16.7**	**86.9%**	12.7	75.7%
0.9	22.5	**99.4%**	5.3	41.6%	**19.3**	**91.2%**	15.0	79.8%	**18.2**	**89.5%**	14.2	78.2%	**9.8**	**64.3%**	8.1	61.5%
1.0	21.6	**99.5%**	-3.6	11.7%	**18.9**	**90.2%**	13.9	76.8%	**17.9**	**88.3%**	10.9	67.5%	**13.8**	**79.4%**	7.0	58.2%

Figure 7: Reward and confidence interval of Experiment 1 (left) and Experiment 2 (right) for different relation probabilities r comparing the proposed CEDM to the MDDM baseline. The measures only show the performance of the second object in the dialogue where the relation is relevant. All results are computed after 4,000/1,000 train/test dialogues and averaged over 5 trials with different random seeds.

policy $rew = 18.8$, $suc = 90.2\%$.

Analysing the system actions of the dialogues shows that the CEDM learns to address a relation in up to 24.5% of all dialogues for $r = 1.0$.

6 Conclusion and Future Work

In this paper, we have presented a novel dialogue model for statistical spoken dialogue systems that is centred around objects and relations (instead of domains) thus offering a new way of modelling statistical dialogue. The two major advantages of the new model are the capability of including multiple objects of the same type and the capability of modelling and addressing relations between the objects. By assigning a part of the belief state not only to each object but to each relation as well, the system is able to address the relations in a system response.

We have demonstrated the importance of the aspect of relation modelling—a core functionality of our proposed model—in simulated experiments showing that by using a hierarchical feudal pol-

icy architecture, adequate policies may be learned that lead to successful dialogues in cases where relations are often mentioned by the user. Furthermore, the resulting policies also learned to address the relation itself in the system response.

However, only a small part of the proposed dialogue model has been evaluated in this paper. To explore its full potential, many questions need to be addressed in future work. For creating a suitable semantic decoder that is able to semantically parse linguistic information about relations, an extensive prior work on named entity recognition and dependency parsing already exists and needs to be leveraged and applied to conduct real user experiments. Moreover, relations other than *equals* need to be investigated. Finally, the challenges of identifying all conversational entities in the dialogue and assigning the correct one to each user action as well as finding suitable belief-tracking approaches for the proposed multi-layered architecture along with effective policy models need to be addressed.

References

Eric Bilange. 1991. An approach to oral dialogue modelling. In *The Structure of Multimodal Dialogue; Second VENACO Workshop*.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. 2017. Subdomain modelling for dialogue management with hierarchical reinforcement learning. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 86–92. Association for Computational Linguistics.

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

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina Rojas-Barahona, Bo-Hsiang Tseng, and Milica Gašić. 2018. Feudal reinforcement learning for dialogue management in large domains. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (HLT/NAACL)*.

Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, and Olivier Pietquin. 2012. A comprehensive reinforcement learning framework for dialogue management optimization. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):891–902.

Peter Dayan and Geoffrey E Hinton. 1993. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–278.

Mihail Eric and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrcken, Germany. Association for Computational Linguistics.

Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2017. Dialogue manager domain adaptation using gaussian process reinforcement learning. *Computer Speech and Language*, 45:552–569.

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

Barbara J. Grosz. 1977. The representation and use of focus in dialogue understanding. Technical report, SRI International Menlo Park United States.

Barbara J. Grosz. 1978. Focusing in dialog. In *Proceedings of the 1978 Workshop on Theoretical Issues in Natural Language Processing*, TINLAP '78, pages 96–103, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tobias Heinroth and Wolfgang Minker. 2013. *Introducing Spoken Dialogue Systems into Intelligent Environments*. Springer, Boston (USA).

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

Sungjin Lee and Amanda Stent. 2016. Task lineages: Dialog state tracking for flexible interaction. In *SIGDial*, pages 11–21, Los Angeles. ACL.

Oliver Lemon and Olivier Pietquin. 2012. *Data-Driven Methods for Adaptive Spoken Dialogue Systems*. Springer New York.

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

Pierre Lison. 2011. Multi-policy dialogue management. In *Proceedings of the SIGDIAL 2011 Conference*, pages 294–300. Association for Computational Linguistics.

Georgios Meditskos, Stamatia Dasiopoulou, Louisa Pragst, Stefan Ultes, Stefanos Vrochidis, Ioannis Kompatsiaris, and Leo Wanner. 2016. Towards an ontology-driven adaptive dialogue framework. In *Proceedings of the 1st International Workshop on Multimedia Analysis and Retrieval for Multimodal Interaction*, pages 15–20. ACM.

Germán Montoro, Xavier Alamán, and Pablo A Haya. 2004. A plug and play spoken dialogue interface for smart environments. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 360–370. Springer.

Alexandros Papangelis and Yannis Stylianou. 2017. Single-model multi-domain dialogue management with deep learning. In *International Workshop for Spoken Dialogue Systems*.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240. Association for Computational Linguistics.

Louisa Pragst, Stefan Ultes, Matthias Kraus, and Wolfgang Minker. 2015. Adaptive dialogue management in the kristina project for multicultural health care applications. In *Proceedings of the 19thWorkshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 202–203.

Deepak Ramachandran and Adwait Ratnaparkhi. 2015. Belief tracking with stacked relational trees. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 68–76.

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

Hannes Schulz, Jeremie Zumer, Layla El Asri, and Shikhar Sharma. 2017. A frame tracking model for memory-enhanced dialogue systems. *arXiv preprint arXiv:1706.01690*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *SIGdial*, pages 147–157, Saarbrcken, Germany. ACL.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*, 1st edition. MIT Press, Cambridge, MA, USA.

Blaise Thomson and Steve J. Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve J. Young. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *ACL Demo*. Association of Computational Linguistics.

Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. 2014. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 57–67.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677, Vancouver, Canada. Association for Computational Linguistics.

Jason D. Williams, Pascal Poupart, and Steve J. Young. 2005. Factored partially observable markov decision processes for dialogue management. In *4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, International Joint Conference on Artificial Intelligence (IJCAI)*, pages 76–82.

Jason D. Williams and Steve J. Young. 2006. Scaling pomdps for dialog management with composite summary point-based value iteration (cspbvi). In *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, pages 37–42.

Yushi Xu and Stephanie Seneff. 2010. Dialogue management based on entities and constraints. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 87–90. Association for Computational Linguistics.

Steve J. Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

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

A Example Dialogues

1:	Sys	hello()
2:	User	inform(CamHotels#type="placetostay", CamHotels#area="west", CamHotels#pricerange="moderate", CamHotels#kind="guesthouse")
3:	Sys	inform(CamHotels#kind="guesthouse", CamHotels#pricerange="moderate", CamHotels#name="hobsons house", CamHotels#area="west")
4:	User	reqalts()
5:	Sys	inform(CamHotels#name="none", CamHotels#name!="hobsons house", CamHotels#kind="guesthouse", CamHotels#pricerange="moderate", CamHotels#area="west")
6:	User	reqalts(CamHotels#area="north")
7:	Sys	inform(CamHotels#kind="guesthouse", CamHotels#pricerange="moderate", CamHotels#name="limehouse", CamHotels#area="north")
8:	User	request(CamHotels#price)
9:	Sys	inform(CamHotels#name="limehouse", CamHotels#price="a cheapest single room is 40 pounds and a cheapest double room is 75 pounds")
10:	User	inform(CamRestaurants#type="restaurant", CamRestaurants#food="british")
11:	Sys	request(CamRestaurants#pricerange)
12:	User	inform(CamRestaurants#pricerange="moderate")
13:	Sys	request(CamRestaurants#area)
14:	User	inform(CamRestaurants#area=CamHotels#area)
15:	Sys	inform(CamRestaurants#name="none", CamRestaurants#food="british", CamRestaurants#pricerange="moderate", CamRestaurants#area="north")
16:	User	negate(CamRestaurants#area="west")
17:	Sys	confirm(CamHotels#area=CamRestaurants#area)
18:	User	negate(CamRestaurants#area="west")
19:	Sys	inform(CamRestaurants#food="british", CamRestaurants#pricerange="moderate", CamRestaurants#name="saint johns chop house", CamRestaurants#area="west")
20:	User	bye()
21:	Sys	bye()

Figure 8: An example simulated dialogue using the proposed CEDM (Experiment 1, Env. 1, $r = 0.7$) about finding a hotel first (blue) followed by a restaurant (green). The system is able to process user input addressing a relation (Turn 14) as well as to address the relation in a system utterance (Turn 17). The dialogue nicely shows how the system is able to resolve a conflict between the belief of the conversational relation and the belief of the conversational object (i.e., CamRestaurants#area = CamHotels#area vs. CamRestaurants#area = "west").

1:	Sys	hello()
2:	User	inform(CamHotels#type="placetostay", CamHotels#kind="hotel", CamHotels#area="north", CamHotels#stars="2")
3:	Sys	inform(CamHotels#kind="hotel", CamHotels#name="lovell lodge", CamHotels#stars="2", CamHotels#area="north")
4:	User	confirm(CamHotels#kind="hotel")
5:	Sys	inform(CamHotels#name="lovell lodge", CamHotels#kind="hotel")
6:	User	inform(CamRestaurants#type="restaurant", CamRestaurants#food="chinese", CamRestaurants#pricerange="expensive")
7:	Sys	request(CamRestaurants#area)
8:	User	inform(CamRestaurants#area=CamHotels#area)
9:	Sys	request(CamRestaurants#area)
10:	User	inform(CamRestaurants#area=CamHotels#area)
11:	Sys	request(CamRestaurants#area)
12:	User	inform(CamRestaurants#area=CamHotels#area)
13:	Sys	request(CamRestaurants#area)
14:	User	inform(CamRestaurants#area="north")
15:	Sys	inform(CamRestaurants#food="chinese", CamRestaurants#pricerange="expensive", CamRestaurants#name="hakka", CamRestaurants#area="north")
16:	User	bye()
17:	Sys	bye()

Figure 9: An example dialogue using the MDDM (Experiment 1, Env. 1, $r = 0.7$) about finding a hotel first (blue) followed by a restaurant (green). As the system is not able to process user input addressing a relation it keeps requesting the same information (Turns 7-12: CamRestaurants#area) over and over until the user finally utters the actual value of the area (Turn 14: CamRestaurants#area="north") which then leads to a successful dialogue.