Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 137–145
Marseille, 20-25 June 2022

Every Time I Fire a Conversational Designer, the Performance of the Dialogue System Goes *Down*

Giancarlo A. Xompero⋆, Michele Mastromattei†, Samir Salman†, Cristina Giannone⋆, Andrea Favalli⋆, Raniero Romagnoli⋆, Fabio M. Zanzotto†
⋆Almawave SpA, Rome, Italy
{g.xompero, c.giannone, a.favalli, r.romagnoli}@almawave.it
†University of Rome Tor Vergata, Rome, Italy
{michele.mastromattei, fabio.massimo.zanzotto}@uniroma2.it

Abstract

Incorporating handwritten domain scripts into neural-based task-oriented dialogue systems may be an effective way to reduce the need for large sets of annotated dialogues. In this paper, we investigate how the use of domain scripts written by conversational designers affects the performance of neural-based dialogue systems. To support this investigation, we propose the Conversational-Logic-Injection-in-Neural-Network system (CLINN) where domain scripts are coded in semi-logical rules. By using CLINN, we evaluated semi-logical rules produced by a team of differently-skilled conversational designers. We experimented with the *Restaurant domain* of the MultiWOZ dataset. Results show that external knowledge is extremely important for reducing the need for annotated examples for conversational systems. In fact, rules from conversational designers used in CLINN significantly outperform a state-of-the-art neural-based dialogue system when trained with smaller sets of annotated dialogues.

Keywords: neural-based dialogue systems, task-oriented dialogue, handwritten rules, hybrid dialogue systems

1. Introduction

Is it possible that trainable end-to-end task-oriented dialogue systems need thousands of annotated examples to learn domain scripts, which conversational designers can partially write? Domain scripts have been crucial for customizing *traditional* dialogue systems (Bohus and Rudnicky, 2009) and are pivotal in neural dialogue systems. Indeed, these neural dialogue systems (Wen et al., 2017; Liu and Lane, 2018) handle domain scripts with two dedicated modules – the dialogue state tracker (DST) and the dialogue policy manager (DPM). The thousands of annotated examples are needed to approximate sufficiently the data distribution of the target domain (Evans and Grefenstette, 2018) when learning these DSTs and DPMs.

Annotating dialogues is a long process and, thus, reducing their need for training dialogue systems is a flourishing research area. Along this line of thinking, reinforcement learning is often used to gain explicit knowledge from *active* users (Zhao and Eskenazi, 2016; Williams et al., 2017; Jhunjhunwala et al., 2020). Even virtual active users, implemented in sort of adversarial networks (Liu and Lane, 2017), have been explored. Yet, an effective strategy is exploiting the domain knowledge of conversational designers by giving them a language to write domain scripts (Altszyler et al., 2021). In this way, even neural-based dialogue systems act as "humans that learn to perform the same tasks by reading a description" (Weller et al., 2020), that is, domain scripts written by conversational designers.

However, using high skilled conversational designers to manually develop domain scripts is in contrast with the mainstream in natural language processing (NLP), set with the famous Fred Jelinek's 1985 quote (Moore, 2005): *"Every time I fire a linguist, the performance of the speech recognizer goes up"*. Actually, the winning design pattern for NLP systems is combining low skilled annotators with machine learning algorithms, which extract implicit models from annotated corpora. High skilled rule writers, as conversational designers, are generally put aside.

In this paper, we aim to investigate whether handwritten domain scripts can alleviate the need for large sets of annotated dialogues in trainable end-to-end dialogue systems. The underlying question is where the investment should go when adapting dialogue systems: (1) annotating dialogues with low skilled annotators or (2) manually compiling domain scripts with conversational designers. As far as we know, this is the first study on how the quality of handwritten domain scripts affects the overall performance of end-to-end dialogue systems. To support this investigation, we propose the Conversational-Logic-Injection-in-Neural-Network system (CLINN). CLINN builds upon the Domain Aware Multi-Decoder (DAMD) network (Zhang et al., 2019), which is a state-of-the-art trainable end-to-end task-oriented dialogue system. CLINN includes a dedicated symbolic semi-logic language, in line with Jhunjhunwala et al. (2020), to allow the manual writing of rules for dialogue scripts for dialogue state tracker and dialogue policy manager of DAMD. We also use CLINN in order to investigate the quality of handwritten dialogue scripts produced by a team of differently-skilled conversational designers. We experimented with the *Restaurant domain* of the MultiWOZ dataset (Budzianowski et al., 2018). We used two different sets of dialogues to allow conversational designers to generate domain scripts. Results show that domain scripts injected are effective in situations in which training data are scarce and, moreover, experience in writing domain scripts is extremely important. In fact, CLINN, combined with DAMD, significantly outperforms DAMD when CLINN uses domain scripts of expert conversational designers.

2. Background and Related Work

Task-oriented dialogue systems are gaining impressive attention in several real scenarios. However, when dialogue systems are evaluated in settings with real users (Laranjo et al., 2018), their underlying models show all their limitations.

A specific study has shown the limitations of traditional rule-based dialogue systems in the health domain (Miner et al., 2016) when evaluated by external research groups. Devising strategies to generate more effective dialogue systems is then a clear need.

Learning end-to-end dialogue systems seems to be the path to go, but huge annotated training sets are needed. Moreover, it is difficult to build up datasets in order to cover the expected distribution of dialogues in the target domain. It turns out that these datasets are quite sparse (Budzianowski et al., 2018; Kim et al., 2017). Alternative ways to help train neural-based dialogue systems are then gaining attention.

Reinforcement learning is often used to reduce the centrality of annotated datasets. A fairly interesting approach is using an Agenda-Based User Simulator (ABUS) (Liu and Lane, 2017; Schatzmann et al., 2007), which avoids introducing real humans into the learning loop. The advantage of using a user simulator is to get good performance without collecting data for supervised dialogue policy – an expensive and time-consuming process. The basic idea of an ABUS model is to build hand-crafted rules according to an agenda which is declared before the dialogue is started. ABUS has been used in different domains such as the movie domain (Li et al., 2016). Nevertheless, there is no standard automatic metric for evaluating these user simulators, as it is unclear to define how closely the simulator resembles real user behaviors. Indeed, although there are standards metrics to evaluate a user simulator under different aspects (Kobsa, 1994; Chin, 2001), there is no metric that actually correlates with the performance of a user simulator with human satisfaction (Shi et al., 2019).

A more direct way to introduce knowledge into neural-based dialogue systems is by injecting rules of domain scripts into dialogue state trackers and dialogue policy managers. This approach is the most general line of research of merging symbolic knowledge and neural networks. In the context of neural-based dialogue systems, this line is pursued by using constrained rules (Jhunjhunwala et al., 2020), logical rules to be used in inductive logic programming (Zhou et al., 2020) or declarative languages (Altszyler et al., 2021). These rules and models can be easily included in the existing dialogue state tracking models to guide the training and prediction phases without additional learning parameters (Hu et al., 2016; van Krieken et al., 2022). These models obtain the same advantage of the user simulator and in addition overcome the problem of the evaluation of the user-simulator itself. Indeed, the injected knowledge is, in different ways, rules governed by conversational designers.

However, there is not an extensive study on how conversational designers may affect the performance of the overall system by writing these additional rules for domain scripts.

3. Method and System

Our solution to inject knowledge of conversational designers into end-to-end dialogue systems is the Conversational-Logic-Injection-in-Neural-Network system (CLINN). CLINN is a rule-based dialogue state tracker which allows to use handwritten domain scripts. It is used in combination with the Domain Aware Multi-Decoder network (DAMD), which is a state-of-the-art end-to-end dialogue system. This section describes, firstly, DAMD and, then, CLINN.

3.1. Domain Aware Multi-Decoder Network

The Domain Aware Multi-Decoder network (DAMD) (Zhang et al., 2019) offers a great opportunity to inject external knowledge from handwritten domain scripts. In fact, DAMD produces symbolic representations of dialogue states at each turn of the dialogue. Dialogue states S_t at the time t are triples (R_t, B_t, A_t) where B_t is the belief state, A_t is the selected action, and R_t is the answer of the system given the action A_t. The symbolic representation of these states is based on *belief spans* (Lei et al., 2018). These belief spans are sequences of symbols expressing belief states, which are the inner parts of dialogue states.

Moreover, DAMD is a module-based neural network that, at a given time t, takes S_{t-1} and U_t as inputs, where U_t is the inserted user utterance, and produces S_t. DAMD consists of four seq-to-seq modules plus access to an external database (see Fig. 1). The four modules, which partially work as DST and DPM, behave as follows. The *context encoder* encodes the context of the turn (U_t, R_{t-1}) in a context vector c_t. The *belief span decoder* receives the previous belief span B_{t-1} and, combined with the context vector c_t, produces the belief span B_t of the current turn. This B_t is used to query the database DB and the answer DB_t is concatenated with B_t and U_t to form the internal state S_t of the

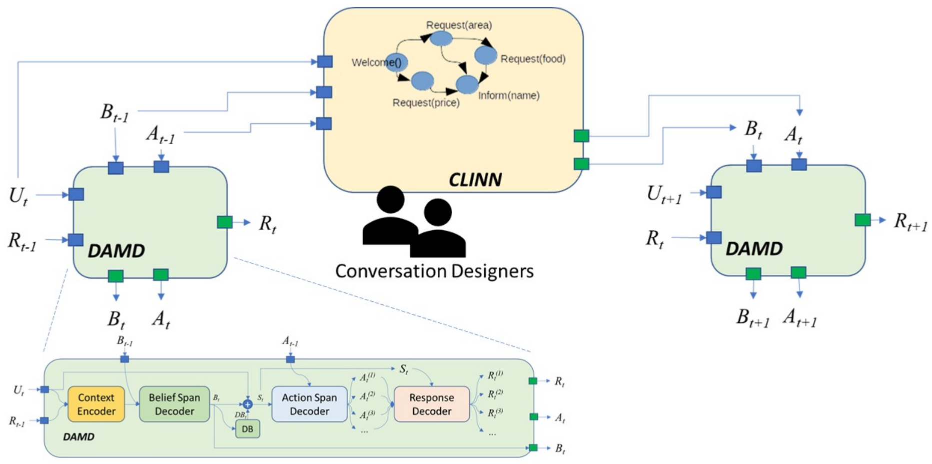

Figure 1: Injecting external handwritten domain scripts with the Conversational-Logic-Injection-in-Neural-Network (CLINN) and the architecture of the Domain Aware Multi-Decoder (DAMD) network.

turn. Then, the *action span decoder* produces the current action $A_t^{(i)}$ by taking into consideration the current internal state S_t and the previous action A_{t-1} . Finally, the *response decoder* emits the final response $R_t^{(i)}$ taking into consideration the current internal state S_t and the corresponding action $A_t^{(i)}$. In Zhang et al. (2019), multiple actions and multiple responses are produced to increase variability in dialogues and, for this reason, the framework is called multi-action data augmentation.

In this work, we decided to use a simplified version of the DAMD architecture, which receives in input the user's dialogue act U_t and the system action A_{t-1} instead of the system response R_{t-1}. Moreover, we removed the response decoder, leaving a simple action decoder that generates a single action A_t.

3.2. Developing Domain Scripts for Dialogue State Trackers

Building on DAMD, we propose a knowledge-based dialogue state tracker, that is, our Conversational-Logic-Injection-in-Neural-Network (CLINN). CLINN allows conversational designers to develop domain scripts by using symbolic transition rules. It is a fully operational dialogue state tracker, which can evolve by itself (CLINN-base) or work in cooperation with DAMD (CLINN+DAMD) by substituting dialogue states when its transition rules fire (see Fig. 1).

In the following, we describe the representation of dialogue states and of transition rules in a semi-logical language.

3.2.1. Dialogue States in a Semi-logical Language

As far as we aim to inject handwritten domain scripts into the dialogue system, we need to clarify how dialogue states are represented and how rules for domain scripts can be described. For these reasons, we express both dialogue states and rules in a logical form (as in Zhou et al. (2020)). By using this logical form, rules can be expressed by using logical constraints and variables. The shared representation of dialogue states (cf. Henderson et al. (2019)) is then presented in the following way:

$$S_t \quad = \quad \begin{array}{ll} \hline U_t & \textit{Inform(food(thai))} \\ & \textit{Request(name(\textbf{?}))} \\ \hline B_{t-1} & \textit{area(west)} \\ \hline A_{t-1} & \textit{Request(food(\textbf{?}))} \\ \hline \end{array}$$

In belief states B_{t-1}, logical facts are represented as *feature(value)*, for example, *area(west)*. Instead, in the case of user utterances U_t and system actions A_{t-1}, feature-value pairs are inserted into predicates representing dialogue acts and, then, are represented as *dialogue_act(feature(value))*, for example, *Inform(food(thai))*. Requested values are indicated with "**?**". This representation is needed to indicate dialogue acts hidden in user utterances and in system actions. The catalogue of dialogue acts we are using is presented in the Experimental section (Table 1).

3.2.2. Symbolic Transition Rules in a Semi-logical Language

Domain scripts consist of symbolic transition rules for controlling the dialogue state tracker. These transition rules are then expressed in a logic programming formalism, that is, horn clauses with variables. For the sake of simplicity, these rules are expressed as *preconditions* and *actions*. Preconditions are matched on the current dialogue states. If preconditions fire and variables are unified with current values, the result of the rule is to add or replace the bounded action in the next dialogue state. In the following, there are two examples of transition rules in Table 2. Given the above state S_t, transition rule R_1 fires. In fact, all its preconditions are satisfied and the variables X and Y are unified to the values *thai* and *west*, as *Inform(food(X))* in U_t is matched to *Inform(food(thai))* and *area(Y)* in B_{t-1} is matched to *area(west)*. As result of the application of

General-domain

bye, greet, reqmore, welcome

Restaurant-domain

inform, request, nooffer, recommend, select, offerbook, offerbooked, nobook

Additional User Dialogue Acts

Act	*Description*
getrecommend	asking for a recommendation
acceptance	accepting system's proposals
rejection	reject system's proposals
alternatives	asking for other restaurants

Table 1: General and Domain Specific Dialogue Acts from MultiWOZ, along with our additional dialogue acts.

Preconditions		$\rightarrow$	*Action*	
$R_1 =$				
U_t	*Inform(food(X))* *Request(name(?))*			
B_{t-1}	*area(Y)*	$\rightarrow$	B_t	*area(Y)* *food(X)*
A_{t-1}	*Request(food(?))*			
$R_2 =$				
U_t	*Inform(food(X))* *Request(name(?))*			
B_t	*area(Y), food(X)*	$\rightarrow$	A_t	*Inform(address(Z))* *Request(price(?))*
A_{t-1}	*Request(food(?))*			
DB_t	*between(4,10)*			

Table 2: Two sample rules of a handwritten domain script.

this transition rule, *area(west)* and *food(thai)* are added to the belief or override existing beliefs. The application of the transition rule R_2 is similar, but its effect is on the next action of the system A_t.

During the writing of transition rules for domain scripts, we asked conversational designers to build up two types of rules: rules affecting only the belief B_t (*Belief rules*) and rules affecting only the action A_t (*Action rules*). Belief and action rules are then sorted in two separated lists, and the selection algorithm takes the first rule for each type whose constraints are satisfied.

Transition rules, as written by conversational designers, may be over-constrained and this fact may hinder its application in novel dialogues. For this reason, transition rules are applied in two ways:

- Fully constrained (*Full*) - all constraints are considered.
- Partially constrained (*Free*) - constraints on previous actions A_{t-1} are not considered for belief rules, and constraints on current beliefs B_t are not considered for action rules.

In the experimental section, we will analyze how this may affect the final performance of the dialogue system.

4. Experiments

Through these experiments, our aim is to investigate: (1) if conversational designers can reduce the need for annotated dialogues in neural-based dialogue systems, and (2) what is the relevance of the skill level of conversational designers for the final performance of learned neural-based dialogue systems.

The section is organized as follows. Section 4.1 shows the setting of our experiments by describing the general principles, the dialogue corpus, the production of transition rules, the evaluation of the coherence of transition rules produced by different conversational designers, and the metrics used to evaluate the dialogue systems. Section 4.2 analyzes the results.

4.1. Experimental Set-Up

4.1.1. General Principles and Dialogue Corpus

The general principle in our experiments involving neural networks is: performing repeated experiments and evaluating statistical significance of the difference among different configurations. Indeed, results of neural-based dialogue systems, as well as results of all experiments using neural networks, may vary a lot depending on the initial conditions. Different seeds given to random pseudo-generators can determine different initial conditions for learning. Therefore, we repeated each experiment involving DAMD for 6 times with 6 fixed seeds. Whenever relevant, we computed paired statistical significance analysis with other configurations.

Experiments are carried out on the *restaurant* domain of the widely-used MultiWOZ dataset (Budzianowski et al., 2018) extended by Lee et al. (2019), as in Zhang et al. (2019). The full dataset has been designed as a human-human task-oriented dialogue dataset collected via the Wizard-of-Oz framework. One participant is the system. The dataset contains conversations on several domains for tourism services (hotel, train, restaurant, taxi,...). Each domain has a set of dialogue acts in addition to general ones such as *greeting* or *bye*. These dialogue acts are used to describe interactions between users and the system. The restaurant domain of this dataset consists of 1200 dialogues for the training set, 61 dialogues for the testing set, and 50 dialogues for the validation set. To simulate data scarcity at different levels, we derived three additional training sets by randomly sampling the full training set. These additional training sets contain 150, 300 and 450 dialogues. Hence, results will be presented for a specific training set and

Group of conversational designers	Inter-designer agreement			
	Small set		Medium set	
	Belief rules	Action rules	Belief rules	Action rules
exps	0.52	0.36	0.49	0.43
exps vs. jrs	0.53	0.27	0.56	0.30
jrs	0.58	0.44	0.89	0.56

Table 3: Inter-designer agreement score within subgroups of conversational designers and between different subgroups computed on the different rule sets.

by using a learning curve with respect to the increasing number of dialogues.

Finally, we improved the restaurant dataset of MultiWOZ by annotating the 500 missing user's dialogue acts[1]. For some of these cases, we have also introduced some additional dialogue acts, which are, in our opinion, more suitable. Additional dialogue acts are listed and described in Table 1.

4.1.2. Designing and Evaluating Transition Rules for Domain Scripts

To investigate our two research questions, we built up a team of five conversational designers divided in two subgroups with different level of expertise: (1) a subgroup of two experts (*exps*), which have more than 15 years of experience in natural language processing and more than 5 years of experience in experimental and production of dialogue systems; (2) a subgroup of three juniors (*jrs*), which have less than 1 year of experience in NLP and no experience in dialogue systems production. The three junior conversational designers have been trained for a week. Clearly, it is extremely difficult to build larger groups of conversational designers. Indeed, conversational designers are experienced professionals, at least as opposed to low skilled dialogue annotators.

The procedure to write domain scripts is the following. Given a fixed set of annotated training dialogues, each conversational designer generates the set of transition rules for domain scripts in two steps: 1) s/he observes the set of annotated dialogues; 2) s/he generates a set of transition rules. We asked conversational designers to produce two separate sets of transition rules: the *Belief rules* and the *Action rules*. The two sets, respectively, act on belief state B_t and on system action A_t.

Conversational designers are exposed to two sets of annotated training dialogues: the *small set* and the *medium set*. The *small set* contains 5 dialogues. The *medium set* contains the small set plus 10 additional dialogues. Firstly, designers see the small set and produce the first set of transition rules and, only then, they see the medium set to produce the second set of transition rules.

To evaluate the difficulty of writing rules for domain scripts, we measured inter-designer agreement within subgroups and between subgroups. We defined the inter-designer agreement measure for each pair of designers as:

$$AGR = \frac{|R1 \cap R2|}{|R1 \cup R2|} \tag{1}$$

where $R1$ and $R2$ are the sets of rules produced by the first and second annotators, respectively. Inter-designer agreement for subgroups is averaged with respect to the pairs of designers as in the *Fleiss' kappa* for inter-annotator agreement.

4.1.3. Evaluation Metrics for Dialogue systems

The automatic evaluation of dialogue systems is, in general, a very difficult problem (Deriu et al., 2021). Yet, since a human evaluation is extremely expensive, we used metrics widely adopted to evaluate both actions and belief states of dialogue systems. These metrics, hereafter described, are: *Action-F1*, *Joint Goal*, *Slot Accuracy* and *Slot F1*. *Action-F1* is the micro-averaged F1-score of the predicted dialogue action a_t compared to the correct one $\hat{a_t}$. *Joint Goal* is defined as the fraction of dialogue turns for which the values v_i for all slots s_i of the belief state are predicted correctly. *Slot Accuracy* is defined as the fraction of slots values correctly predicted by the model over all slot values. *Slot F1* is defined as the micro-averaged F1-score of slot prediction.

4.1.4. Configurations and Meta-parameters

We experimented with four configurations: *DAMD*, *Fully-informed DAMD*, *CLINN-base*, and *CLINN+DAMD*. *DAMD* is the basic DAMD system tested in the configuration where inputs B_{t-1} and A_{t-1} at the step t are the actual B and A produced by the previous application of DAMD. *Fully-informed DAMD* is the basic DAMD system tested in the configuration where inputs B_{t-1} and A_{t-1} at the step t are taken from the ground truth. DAMD and *Fully-informed DAMD* represent the lower and upper bounds of our study, respectively. *CLINN-base* is a system that evolves only utilizing transition rules written by conversational designers. Finally, *CLINN+DAMD* is a combination of DAMD and the module that applies rules written by designers.

DAMD is mainly trained with almost the same hyper-parameters used in Zhang et al. (2019). Our version of DAMD has 3 encoders and 2 decoders based on single-layer bidirectional GRUs with hidden size of 100. Since our focus is only on the restaurant domain, DAMD relies on a vocabulary restricted to words of that domain.

[1] The dataset will be available upon request.

Model	Designed Domain Script			Metrics			
	Size	Type	Designer Group	Action F1	Joint Goal	Slot Acc	Slot F1
CLINN-base	Small	Full	exps	14.1	29.5	92.65	56.7
			jrs	8.9	22.8	92	49.7
		Free	exps	14.6	38.8	93.8	66.9
			jrs	8.9	25.2	92.4	56.2
	Medium	Full	exps	18.8	28.6	92.8	58.8
			jrs	8.9	25.2	92.4	56.2
		Free	exps	19.6	44.9	94.8	72.6
			jrs	19.7	45.7	95.2	75.1
Fully-informed-DAMD				44.8	72.2	98.4	92.9
DAMD				43.7	56.7	97.1	87.6
CLINN+DAMD	Small	Full	exps	43.3	$57.8^{\dagger\diamond}$	$97.2^{\diamond\diamond}$	$87.9^{\dagger\diamond}$
			jrs	43.7	$57.2^{\dagger\dagger\dagger}$	$97.2^{\diamond\diamond\diamond}$	$87.9^{\star\star\star}$
		Free	exps	43.3	$61.5^{\star\star}$	$97.5^{\star\star}$	$88.9^{\star\star}$
			jrs	43.5	$61.3^{\star\star\star}$	$97.4^{\star\diamond\diamond}$	$88.8^{\star\star\star}$
	Medium	Full	exps	$44.9^{\star\star}$	$59.6^{\star\diamond}$	$97.3^{\star\diamond}$	$88.3^{\star\star}$
			jrs	43.5	57.1	97.2	87.8
		Free	exps	$44.8^{\dagger\dagger}$	$63.3^{\star\dagger}$	$97.6^{\star\star}$	$89.7^{\star\star}$
			jrs	$44.6^{\dagger\star\dagger}$	$61.6^{\diamond\diamond\diamond}$	$97.5^{\dagger\dagger\dagger}$	$89.3^{\dagger\dagger\dagger}$

Table 4: Results over the test set for CLINN and DAMD systems. Results of CLINN-base and CLINN+DAMD are averaged on the target group of conversational designers (exps or jrs). DAMD results are the average of 6 runs over a training set of 300 examples. CLINN+DAMD is trained as DAMD for each member of the target group. Symbols $\dagger$,$\diamond$ and $\star$ indicate that the difference between the result of one member of the Designer group and the result of DAMD is statistically significant with a confidence level of, respectively, 90%, 95%, and 99% with the sign test.

4.2. Results and Discussion

Results from the experiments are relevant both for industrial practice and for research. In this section, firstly, we analyze the relative quality and agreement level of the conversational designers. Secondly, we investigate the quality of the produced rule sets for domain scripts used in CLINN. Finally, we describe the limitations of our study.

Our first observation is that writing rules for domain scripts is not easy. Agreement is low in writing these rules and seems to decrease with expertise level (see Table 3). Indeed, nearly all inter-designer agreements are lower than 0.60. The only outlier is 0.89 of the Belief rules for the Medium set of annotated dialogues. Action rules are more difficult to define than Belief rules, as the agreement on Action rules is generally lower than the one on Belief rules. Moreover, agreement in subgroups with the same level of expertise is higher than agreement between subgroups (0.27 for the Small set and 0.30 for the Medium set). Experience of conversational designers generally increases the level of disagreement: the jrs subgroup has a higher agreement with respect to exps subgroup. Reading more annotated dialogues helps jrs to be more convinced on the same set of rules. The agreement on Belief rules surges from 0.58 of the Small set to 0.89 of the Medium set. The agreement on Action rules increases from 0.44 to 0.56. This is not true for the exps subgroup. Then, experts seem to use their knowledge combined with the one derived from observed dialogues whereas juniors seem to be more influenced by annotated dialogues they see.

To understand the performance improvement obtained with handwritten domain scripts, we report results of two configurations of the fully neural-based dialogue system: DAMD, which is our baseline, and Fully-informed-DAMD, which gives the upper-bound that can be obtained by using only annotated dialogues (see Table 4). These two configurations are useful to understand if transition rules are effective or not. For example, there is a very small space of improvement in metrics like Action F1 – 1.05 difference in mean – and Slot Accuracy – 1.20 difference in mean. Moreover, Fully-informed DAMD outperforms DAMD with statistical significance for all the metrics.

It seems to be clear that only handwritten domain scripts are not sufficient to build up a dialogue system that can efficiently handle testing dialogues. Results from CLINN-base are not satisfactory. There is no CLINN-base configuration whose result is in between the baseline and the upper bound of neural dialogue systems, that is, DAMD and Fully-informed-DAMD. An integration between handwritten domain scripts and neural-based dialogue systems is desired.

The injection of handwritten domain scripts into neural dialogue systems is effective and useful. In fact, all the CLINN+DAMD configurations outperform the DAMD system in Joint Goal, Slot Accuracy, and Slot F1 metrics. The difference, except for some cases, is statistically significant (see Table 4). In the case of Action F1, CLINN+DAMD significantly outperforms DAMD for the majority of Medium rule set configurations, while the Small rule set seems not very effective for this metric according to the lack of significant improvements in performance. Moreover, rules designed by experts achieve overall better scores than other configurations (44.9 and 44.8), while juniors' ones have statistically significant better scores only for Free rule type (44.6).

Using handwritten domain scripts with less constraints seems to be the way to go when used in combination with DAMD. The configuration *Free* is better than the

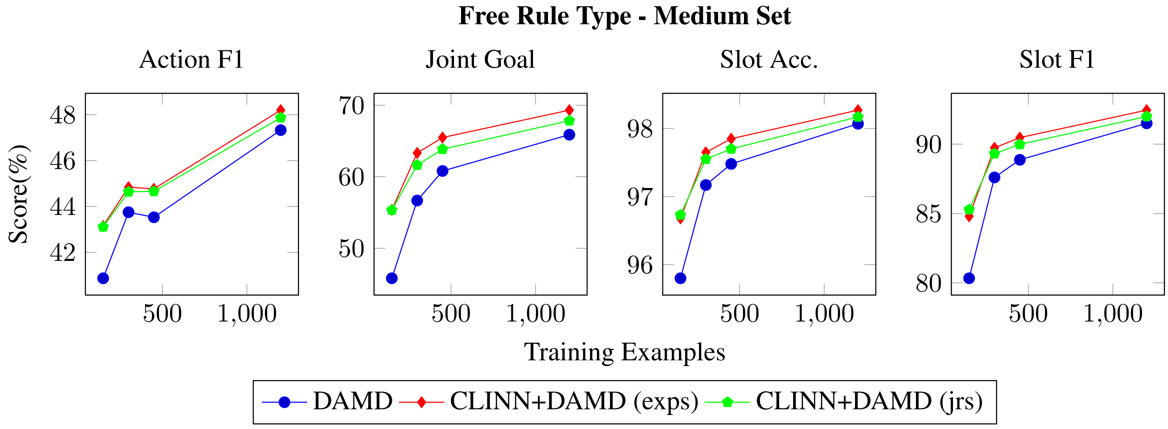

Figure 2: Trend of models' performances when increasing the number of dialogue examples used to train DAMD. The plots show average results of the configurations where CLINN uses the *Medium* set of the *Free* Rule type.

configuration *Full* for quite all the metrics in both Small and Medium settings (see Table 4). Indeed, using less constraints causes an overall improvement in performance. For example, an improvement of 3.7 in Joint Goal is obtained in the case of Medium rule set of exps (63.3), and additional 4.5 average improvement is obtained for the same rule set written by jrs (61.6). Similar performance gains in Joint Goal are also obtained with Small rule sets of both groups of conversational designers, where rules of exps and jrs obtain respectively 3.7 and 4.1 average improvement. In addition, most metrics show better performances using rules with less constraints. Similar effects are also observed in CLINN-base configurations, suggesting that using over-constrained rules with DAMD seems to be less effective.

Experience in writing transition rules is important. CLINN+DAMD using domain scripts of experts significantly outperforms DAMD in most configurations, except for Action F1 metric where results with Small rule sets are slightly worse than DAMD evolving alone. Moreover, experts are able to gain more effective rules by reading additional dialogues. The difference in Joint Goal between Small-Free and Medium-Free is higher for experts (61.5 to 63.3) than for juniors (61.3 to 61.6).

Finally, using conversational designers to build up transition rules seems to be better than using effort in annotating additional dialogues. In fact, adding training examples to DAMD does not clearly outperform CLINN+DAMD (Figure 2) in most metrics. DAMD with 1,200 examples behaves similarly to CLINN+DAMD with rules of experts that uses 450 training examples. This version of DAMD is even close to CLINN+DAMD with rules of experts using only 300 training examples. This is a very important observation, as it suggests a clear view for where to invest time and efforts.

There are, of course, some limitations in this study to acknowledge. Firstly, actions produced by DAMD and CLINN do not contain values of informed slots, preventing belief state trackers from accessing possible additional information that should be tracked in the next turn. Secondly, when CLINN produces only the belief state B_t, the action A_t, generated by DAMD and which will be forwarded to the next turn, is generated according to DAMD's B_t; this is due to the architecture of DAMD that prevents the replacement of the hidden representation of DAMD's B_t with the CLINN's symbolic B_t. However, these limitations do not falsify our previous conclusions.

5. Conclusion

Merging pre-existing explicit knowledge and learning from examples is one of the most important research lines in studies in learning neural networks and, in general, in machine learning. Yet, there is not a clear understanding on how the quality of teachers affects the results of final systems.

In this paper, we carried out a study on how rules provided by conversational designers affect the performance of neural-based dialogue systems. We firstly collected different sets of rules derived from task-oriented dialogue systems implemented by differently-skilled conversational designers; then we combined them with a neural-based dialogue system by applying these rules to situations in dialogue for which they are appropriate. Our results are an important indication as we showed that designers can significantly reduce the sets of annotated dialogue examples, especially in the case of more experienced designers. Moreover, we gained some insights about how different skills of designers affect dialogue systems designing and, hence, their performances. Therefore, as a general contribution, our study showed that, in contrast with the main stream in natural language processing, companies developing dialogue systems should invest more in experienced conversational designers and less in extensive dialogue collection and annotation.

Acknowledgments

We would like to thank Valentina Bellomaria and Caterina Masotti for their valuable help and discussions.

6. Bibliographical References

Altszyler, E., Brusco, P., Basiou, N., Byrnes, J., and Vergyri, D. (2021). Zero-shot multi-domain dialog state tracking using prescriptive rules. In Artur S. d'Avila Garcez et al., editors, *Proceedings of the 15th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 1st International Joint Conference on Learning & Reasoning (IJCLR 2021), Virtual conference, October 25-27, 2021*, volume 2986 of *CEUR Workshop Proceedings*, pages 57–66. CEUR-WS.org.

Bohus, D. and Rudnicky, A. I. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech and Language*, 23(3):332–361.

Budzianowski, P., Wen, T. H., Tseng, B. H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018). MultiWoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*.

Chin, D. N. (2001). Empirical evaluation of user models and user-adapted systems. *User modeling and user-adapted interaction*, 11(1):181–194.

Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., and Cieliebak, M. (2021). Survey on evaluation methods for dialogue systems. *Artif. Intell. Rev.*, 54(1):755–810, jan.

Evans, R. and Grefenstette, E. (2018). Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:65–170.

Henderson, M., Vulic, I., Gerz, D., Casanueva, I., Budzianowski, P., Coope, S., Spithourakis, G., Wen, T. H., Mrkšic, N., and Su, P. H. (2019). Training neural response selection for task-oriented dialogue systems. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 5392–5404, 6.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany, August. Association for Computational Linguistics.

Jhunjhunwala, M., Bryant, C., Shah, P., and Park, M. (2020). Multi-Action Dialog Policy Learning with Interactive Human Teaching. In *sigdial*.

Kim, S., D'Haro, L. F., Banchs, R. E., Williams, J. D., and Henderson, M. (2017). The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer.

Kobsa, A. (1994). User modeling and user-adapted interaction. In *Conference companion on Human factors in computing systems*, pages 415–416.

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y., and Coiera, E. (2018). Conversational agents in healthcare: A systematic review.

Lee, S., Zhu, Q., Takanobu, R., Zhang, Z., Zhang, Y., Li, X., Li, J., Peng, B., Li, X., Huang, M., and Gao, J. (2019). ConvLab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69, Florence, Italy, July. Association for Computational Linguistics.

Lei, W., Jin, X., Ren, Z., He, X., Kan, M. Y., and Yin, D. (2018). Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 1437–1447. Association for Computational Linguistics.

Li, X., Lipton, Z. C., Dhingra, B., Li, L., Gao, J., and Chen, Y.-N. (2016). A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Liu, B. and Lane, I. (2017). Iterative Policy Learning in End-to-End Trainable Task-Oriented Neural Dialog Models. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2018-Janua(2):5715–5719, 9.

Liu, B. and Lane, I. (2018). End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73, New Orleans, Louisiana, USA, June. Association for Computational Linguistics.

Miner, A. S., Milstein, A., Schueller, S., Hegde, R., Mangurian, C., and Linos, E. (2016). Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health. *JAMA Internal Medicine*, 176(5):619–625, 5.

Moore, R. K. (2005). Results from a survey of attendees at ASRU 1997 and 2003. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 117–120. ISCA.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., and Young, S. (2007). Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers on XX - NAACL '07*, pages 149–152, Morristown, NJ, USA. Association for Computational Linguistics.

Shi, W., Qian, K., Wang, X., and Yu, Z. (2019). How to build user simulators to train RL-based dialog systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* *and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1990–2000, Hong Kong, China, November. Association for Computational Linguistics.

van Krieken, E., Acar, E., and van Harmelen, F. (2022). Analyzing differentiable fuzzy logic operators. *Artif. Intell.*, 302(C), jan.

Weller, O., Lourie, N., Gardner, M., and Peters, M. (2020). Learning from Task Descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wen, T.-H., Vandyke, D., Mrkšić, N., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain, April. Association for Computational Linguistics.

Williams, J. D., Asadi, K., and Zweig, G. (2017). Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 665–677.

Zhang, Y., Ou, Z., and Yu, Z. (2019). Task-Oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9604–9611, 4.

Zhao, T. and Eskenazi, M. (2016). Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhou, Z., Beirami, A., Crook, P., Shah, P., Subba, R., and Geramifard, A. (2020). Resource Constrained Dialog Policy Learning via Differentiable Inductive Logic Programming. In *Proceedigns of CoLing*.