Findings from Experiments of On-line Joint Reinforcement Learning of Semantic Parser and Dialogue Manager with real Users

Matthieu Riou, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre

ARTICLE

Findings from Experiments of On-line Joint Reinforcement Learning of Semantic Parser and Dialogue Manager with real Users

Matthieu Riou and Bassam Jabaian and Stéphane Huet and Fabrice Lefèvre

LIA, Avignon Universty
339 Chemin des Meinajaries, 84140 Avignon, France
name.surname@univ-avignon.fr

Abstract

Design of dialogue systems has witnessed many advances lately, yet acquiring huge set of data remains an hindrance to their fast development for a new task or language. Besides, training interactive systems with batch data is not satisfactory. On-line learning is pursued in this paper as a convenient way to alleviate these difficulties. After the system modules are initiated, a single process handles data collection, annotation and use in training algorithms. A new challenge is to control the cost of the on-line learning borne by the user. Our work focuses on learning the semantic parsing and dialogue management modules (speech recognition and synthesis offer ready-for-use solutions). In this context we investigate several variants of simultaneous learning which are tested in user trials. In our experiments, with varying merits, they can all achieve good performance with only a few hundreds of training dialogues and overstep a handcrafted system. The analysis of these experiments gives us some insights, discussed in the paper, into the difficulty for the system's trainers to establish a coherent and constant behavioural strategy to enable a fast and good-quality training phase.

1. Introduction

While end-to-end deep-learning-based dialogue systems represent a new avenue of research with promising, if not yet definitive, results (Wen et al. 2016 2017; Li et al. 2017), these models require a huge quantity of data to be trained efficiently. Hitherto, it is not clear how some initial (low cost) knowledge can be leveraged for a warm start of the system development followed by on-line training with users, as described in (Ferreira and Lefèvre 2013a; Su et al. 2016). Although some recent works have proposed end-to-end architectures (Dhingra et al. 2017; Wen et al. 2017; Shah et al. 2018; Zhao et al. 2019), they still rely on a prior huge data collection before they can reach usable performance.

So in our work, due to the absence of prior data, the presented system still relies on a classical architecture for goal-directed vocal interaction, with proven capabilities (Ultes et al. 2017). A cascade of modules deals with the audio information from the user downstream with progressive processing steps. First, words are deciphered from the audio signal (Automatic Speech Recognition, ASR). Then, the utterance meaning is derived (Semantic Parsing, SP) and can be combined with previously collected information, and its grounding status from the dialogue history (belief tracking). On top of it, a policy can make a decision about the following action to perform according to some global criteria (dialogue length, success in reaching the goal, etc.). Dialogue Management (DM) is then followed by operations conveying back the information

upstream to users: Natural Language Generation of the dialogue manager action (NLG), then speech synthesis.

The Hidden Information State (HIS) architecture (Young et al. 2009) offers such a global statistical framework to account for the relations between the data handled by the main modules of the system, allowing for a reinforcement learning of the DM policy. It can be implemented with sample-efficient learning algorithms (Daubigney et al. 2012) and can involve on-line learning through direct interactions with users (Ferreira and Lefèvre 2013bc 2015). More recently, on-line learning has been generalised to the input/output modules, SP and NLG, with protocols to control the cost of such operations during this human-in-the loop system development (as in (Ferreira et al. 2015 2016; Riou et al. 2017)). The work presented here is a first attempt to combine the on-line learning of SP and DM in a single phase of development. Not only it is expected to help speed-up and simplify the process, it is also likely to benefit from intertwined improvements of the modules.

In dialogue systems, SP extracts a list of semantic concept hypotheses from an input transcription of the user's query. This list is generally expressed as a sequence of Dialogue Acts (DAs) of the form *acttype(slot=value)* and transmitted to the Dialogue Manager (DM) to make the decision on the future action to perform. State-of-the-art SPs are based on probabilistic approaches and trained with various machine learning methods to tag the user input with these semantic concepts (Lefèvre and de Mori 2007; Hahn et al. 2011; Deoras and Sarikaya 2013). Dealing with supervised machine learning techniques requires a large amount of annotated data which are domain dependent and hardly available.

As a first example of methods to overcome this hindrance, a zero-shot learning algorithm for Semantic Utterance Classification was proposed (Dauphin et al. 2014). This method tries to find a sentence-wise link between categories and utterances in a semantic space. A neural network can be trained on a large amount of non-annotated and unstructured data to learn this semantic space. Building on this idea, Ferreira and Lefèvre (2015) presented a zero-shot learning method for SP (ZSSP) leveragining pre-trained word embeddings (Mikolov et al. 2013).

This approach and its extensions (Bapna et al. 2017; Rojas-Barahona et al. 2018) require neither annotated data nor in-context data and have been recently used for different dialogue system modules (Upadhyay et al. 2018; Zhao and Eskenazi 2018; Bapna et al. 2017). Indeed, the model is bootstrapped with only the ontological description of the target domain and generic word embedding features (learned from large and free general purpose data corpora). But to improve the module further with a light and controlled supervision from the users, an active learning strategy based on an adversarial bandit has also been introduced (Ferreira et al. 2016).

In the same line of ideas, made possible by the sample-efficient Reinforcement Learning (RL) algorithm KTD (Geist and Pietquin 2010a), an active learning scheme has been devised for the training of the DM (Ferreira and Lefèvre 2015). It uses reward shaping (Ng et al. 1999) to take into account local (turn-based) rewards from the user to offer a better control over the learning process and based speed its convergence up.

Stating that solutions exist for active on-line learning of both SP and DM modules, we now consider their simultaneous application to address the issue of the overall system's training. Several challenges must be dealt with in this context. In a modern dialogue system, in contrast to systems handcrafted by experts, many parameters must be learned: contextual mapping of the speakers' words with dialogue acts, as well as the policy of the dialogue manager which aims to choose the best next actions according to a dialogue situation (the history of the dialogue).

Besides, SP and DM intertwine (when one parameter is changed in the semantic parser, it impacts the behaviour of the dialogue manager, and vice versa). So, not only time-saving is foreseen in the joint learning of these two parts of a dialogue system, but also it is expected that good performance will rely on a coherent improvement of both modules.

In this article, a direct application of existing techniques (a bandit algorithm for SP and a RL/Q-learner for DM) is presented and tested; both modules remain separated and the parameters

of their on-line training are kept disjoint. Moreover, another possibility with shared parameters in a single Q-learner is introduced and evaluated. This latter presents the pros and cons of an integrated approach: one single unit makes the decision for the two learned modules and can act more coherently, but it has to be fed with larger inputs to base its decision. And in the case of a RL/Q-learner, it can greatly question its performance w.r.t. the training data size, as it will be shown in our experiments.

From a practical point of view, in this work we developed a system intended to be used in a neuroscience experiment. From inside an fMRI scanner, users interact with a robotic platform, vocally powered by the system, which is live-recorded and displayed inside a head-antenna. Users discuss with the system about an image and they tried jointly to elaborate on the message conveyed by the image (see Section 6 for further details on this task). The overall architecture of the system in the various configurations tested will be later detailed in Section 3. For sake of simplicity, we may note right away that the study is exclusively focused on training SP and DM modules, since well-performing solutions can be used directly off-the-shelf for speech recognition and synthesis.

2. Related work

Former works have investigated on-line learning for vocal interaction systems. More generally, this is a general trend in Machine Learning applied to many fields (such as robotics). This idea should not be confused with the general attempt to introduce human-in-the-loop in other NLP tasks, such as Machine Translation where it is used for post-editing (Koehn and Germann 2014), or even already ancient active learning (Tur et al. 2005).

Specifically, in spoken dialogue systems, early propositions have devised the theoretical foundations of on-line learning, e.g. (Pietquin et al. 2011). Indeed, training a system with direct interactions required a consequent reduction in need of training data, w.r.t. former solutions based on huge pre-collected datasets or user simulators. Other approaches see on-line learning only as a complementary step on top of traditional training (Shah et al. 2018; Hancock et al. 2019) and are more oriented toward user or task adaptation. However, for those willing to fully skip the initial training phase (and its cost) the main difficulty to break down is the slow learning curve of the reinforcement-based algorithms used in the system. Several approaches have helped alleviate this difficulty, such as Gaussian Process modelling (Gašic et al. 2013), reward shaping alone (Su et al. 2016), or combined with Kalman Temporal Differences (Ferreira and Lefèvre 2015), etc. These seminal works have been pursued in several other directions to enlarge the conditions upon which an on-line training could be carried out while ensuring the best ratio between the user's involvement (nature and cost of feedback, manual annotations, etc.) and the level of performance reached, e.g. (Wang and Swegles 2013; Li et al. 2016; Chen et al. 2017ba; Chang et al. 2017).

Joint learning of semantic parsers and dialogue management modules has received a considerable increase in interest since the introduction of end-to-end approaches based on deep neural networks. But even before that, some attempts to jointly train several modules of dialogue systems have been carried out, e.g. joint learning of dialogue management and language generation such as in (Lemon 2011).

More recently, the development of neural technologies applied to SDS has led to solutions presented as being able to fully train the module pipeline (Wen et al. 2016). But it appears that in practice none of the proposed systems could reach good performance while considering all the modules in a single training phase. For instance, even if devised as fully compatible with the model, in (Wen et al. 2016) the semantic belief tracker is not trained with the remainder of the network, and pre-trained modules were used instead. Other works have studied such option (Yang et al. 2017; Padmakumar et al. 2017; Rastogi et al. 2018; Ye et al. 2019). But even if they are based on sound and efficient propositions they have in common to rely on a huge dataset or a user simulator to reach a good level of performance.

Also the existence of newly proposed pre-trained language generators, such as the Transformers-like BERT (Devlin et al. 2018) or GPT (Radford and Salimans 2018), represents an interesting option combined with transfer learning to bootstrap dialogue systems Wolf et al. (2019). But such models are not available for all languages (a French version of BERT (Le et al. 2019) was made available after the experiments reported here were done). Besides, how they can be applied to complex dialogue tasks, other than simple chit-chat, is still under investigation.

So the work presented here is a novel proposition that combines joint learning of the two main dialogue system components and direct learning with users. For the first time, it is shown that this approach is doable and can lead to good result in terms of performance. Yet, of course, many degrees of liberty exist in the way it is implemented and taken over by trainers (users involved during the on-line learning phase). From a first set of experiments we endeavour to derive some useful insights on practical realisation of the whole training process.

2.1 Outline

The remainder of this paper is organised as follows. After presenting the basis of the on-line learning versions of SP in Section 3 and DM in Section 4, we define the simultaneous on-line learning strategies in Section 5. Section 6 provides an experimental study with human evaluations of the proposed approaches, along with an analysis giving us some new insights on the practical implications of on-line learning. We conclude in Section 7.

3. On-line learning for zero-shot SP

This section describes how our SP module addresses two issues. Firstly, the zero-shot learning algorithm is able to infer semantic concepts from transcripts using ontological information but no annotated data. Secondly, an adversarial bandit enables an active learning strategy.

3.1 Zero-shot learning spoken language understanding

The SP model concerned by this study is the ZSSP model presented in (Ferreira et al. 2016) and illustrated in Figure 1. We recalled here only what is necessary to understand its further combination with DM. This model makes use of a semantic knowledge base K and a semantic feature space F. K contains collected examples of lexical chunks associated with each targeted Dialogue Act (DA). K is first populated with ontological information (from a back-end database mainly) exclusively, then is completed during the on-line learning process described hereafter. F is a word embedding representation learned with neural network algorithms on large un-annotated open domain data (Mikolov et al. 2013; Bian et al. 2014).

The ZSSP model builds a scored graph of hypotheses from user utterances. All possible contiguous chunks are considered in the graph and a dot product between the k-most similar vectors and their corresponding assignment coefficients in the K matrix is computed to attribute to each chunk a list of scored semantic hypotheses.

A best-path decoding is performed in order to find the best semantic tags hypothesis for the considered user utterance.

3.2 On-line adaptation based on an adversarial bandit algorithm

An on-line adaptation strategy (facilitated by the zero-shot approach) is adopted, as presented in (Ferreira et al. 2016) and briefly recalled here. In this approach, at each dialogue iteration, the system chooses an adaptation action $i_t \in \mathscr{I}$ and updates K according to the user feedback.

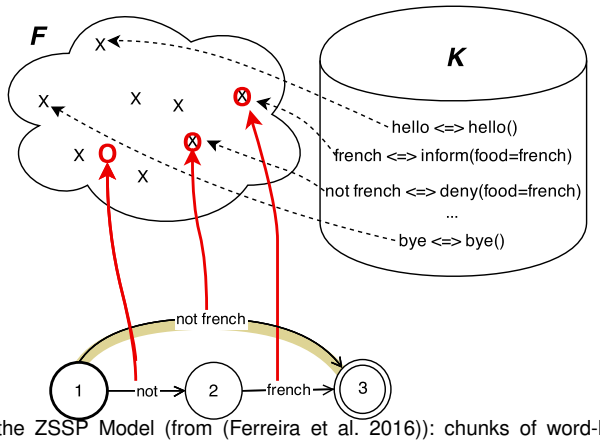

Figure 1. Basics of the ZSSP Model (from (Ferreira et al. 2016)): chunks of word-lattice ASR hypotheses are matched with known tuples (surface form, dialogue act, assignment coefficient) in database K, through an embedding lexical space F

.

The system gain $g(i_t)$, the user effort $\phi(i_t)$ and their combination in the loss function $l(i_t)$ for performing each action are defined and can be estimated during on-line training.

Three possible actions are considered:

- **Skip**: Skip the adaptation process for this turn ($\phi(\text{skip}) = 0$).
- **AskConfirm**: A yes/no question is presented to the user about the correctness of the selected DAs in the best semantic hypothesis. If the whole sentence is accepted, $\phi(\text{AskConfirm}) = 1$. Otherwise, $\phi(\text{AskConfirm})$ is equal to $1+$ the number of DA in the best semantic hypothesis (one yes/no confirmation request per DA).
- **AskAnnotation**: the user is asked to re-annotate the whole utterance. $\phi(\text{AskAnnotation}) = 1$ if the sentence is accepted straight away. Otherwise, the user will first inform the system about which chunks she wants to annotate ($+1$ per selected boundary), and then the system will sequentially ask for *acttype*, *slot* and *value* if necessary ($+1$ per interim question) for each DA.

An adversarial bandit algorithm is used in order to find $i_1, i_2, \ldots, i_t, \ldots$ so that for every t, the system minimises the loss $l(i_t)$. The loss function $l(i) \in [0, 1]$ is calculated as follows:

$$l(i) := \underbrace{\gamma\, g(i)}_{\text{system improvement}} + \underbrace{(1-\gamma)\frac{\phi(i)}{\phi_{max}}}_{\text{user effort}},$$

where $\gamma \in [0, 1]$ balances the importance of information improvement and user effort for the system and $\phi_{max} \in \mathbb{N}^*$ is the maximum number of exchanges between the system and the user (in a same turn/round). In this work, γ has been set to 0.5 for example.

The process is illustrated in Figure 2. From step at time t on the left, a semantic interpretation is proposed to the user who can annotate it. From the graph we can observe that an AskConfirm has been used in this case: the feedback is expressed in binary terms (correct "++" in green, incorrect "- -" in red). From these feedbacks the new entries ("yeah" and "downtown") are added to K and

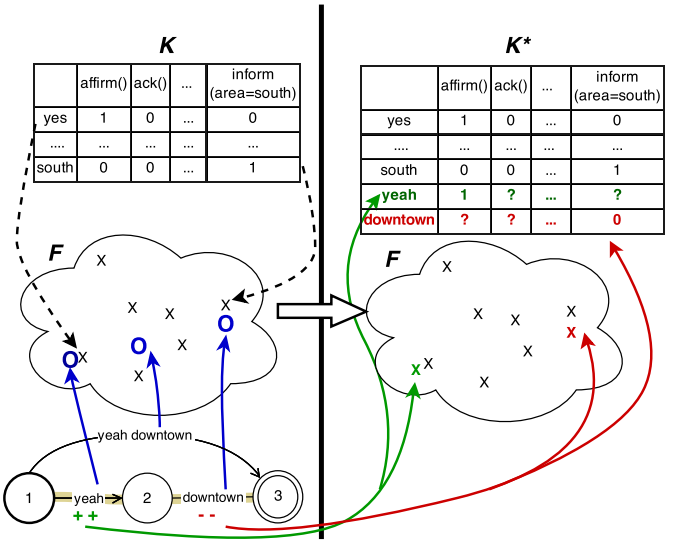

Figure 2. Adaptation process for the ZSSP Model (from (Ferreira et al. 2016)): iterative online application of the model to new user data complements K in a controlled way (w.r.t. annotation cost) and improves its coverage and quality. Here users' feedback is only binary, so assignment coefficients are only updated in K^* to (un)validate the couples (surface form, dialog act) tagged by the users

.

their parameters are set accordingly to the feedback values. It should be noted that in that case the information brought by the user is not always complete:

- "++" means that the proposed act is correct, but it does not mean that it is the only feasible interpretation; in that case, a positive weight associates the entry with the act in K, and uncertainty is recorded for the other possible associations. As such, a next use of this line could permit an exploration of other possible associations and gather returns from the user to specify the weight progressively.

- "- -" implies that the act should not be linked to this entry anymore and that the correct association remains to be found. Therefore, a null weight annihilates the correspondence between the new entry and the proposed act. The other parameters are set in a way to explore other possible associations during further appearances of the entry, until some positive confirmation, as in the previous case, could enforce a confirmed pairing.

4. On-line learning for RL dialogue manager

The dialogue manager used in this paper relies on the system presented in (Ferreira and Lefèvre 2015). It is based on a POMDP (Partially Observable Markov Decision Process) dialogue management framework, the Hidden Information State (HIS) (Young et al. 2009). In a nutshell, the system maintains a distribution over possible dialogue states (the belief state) and uses it to generate an adequate answer. An efficient Reinforcement Learning (RL) algorithm is used to train the system by maximising an expected cumulative discounted reward, with the help of the KTD framework and reward shaping.

4.1 Policy learning of the dialogue manager within the KTD framework

At each turn, the dialogue manager generates several possible answers, depending on its belief state. It generates all possible full dialogue acts and matches them to the 11 summary acts described in Table 2 using heuristic rules. Some summary acts can be deemed impossible to realise at some point if no conversion to a full act is possible; for instance, at least one entity must be selected to propose "Inform" (Gašić et al. 2009). In such a case, a fallback method is implemented which picks the next best possible dialogue act proposed by the policy (knowing that certain actions, e.g. "repeat", are always possible). Generally speaking, impossibility is due to missing data to convert summary acts into fully-specified acts, and therefore into system's text responses, and "Ask" is the only case where the heuristics uses another criterion (repetition) to deem it impossible.

Table 2. List of the summary acts used by the dialogue manager

Greet	Greet user
Bye	End the dialogue
BoldRQ	Bold query request
TentRQ	Tentative query request
Confirm	Confirm an ungrounded piece of information
FindAlt	Find alternative database entity
Split	Distinguish two hypotheses
Repeat	Repeat
Offer	Offer a database entity
Inform	Give info about current offer
QMore	Query if the user wants more information

To learn the policy, i.e. the mapping between situations and actions, an RL approach is used through the KTD learning algorithm (Geist and Pietquin 2010b). This algorithm is derived from a Kalman-based Temporal Differences (KTD) framework, originally devised to track the hidden state of a non-stationary dynamic system through indirect observations of this state. The KTD framework is used in the context of dialogue systems, since it has several advantages and desirable properties for the DM problem. Indeed, it is sample-efficient, it allows on-policy/off-policy learning through two algorithms (respectively KTD-Q and KTD-SARSA) which can both perform on-line and offline learning, it provides ways to deal with the "exploration/exploitation" dilemma using uncertainty on value estimates, it allows value tracking, and it supports linear and non-linear parametrisation.

To model the policy of our DM, a linear case was considered. In that respect, the Q-function has a parametric representation $\hat{Q}_\theta = \theta^T \phi(s,a)$ for each state s and action a. The feature vector $\phi(s,a)$ is a set of n basis functions to be designed by the practitioner and $\theta \in \mathbb{R}^n$ the parameter vector to be learned; the components of the parameter vector θ are the hidden variables which are modelled as a random vector. Such parameter vector is considered to evolve following a random walk through an evolution equation: $\theta_t = \theta_{t-1} + v_t$, with v_t a white noise of covariance matrix P_{v_t}. In order to train the parameters of the DM system with an off-policy learning, the KTD-Q algorithm is here employed (Geist and Pietquin 2010b).

4.2 Reward function for RL dialogue manager

At each turn, the policy selects a summary act to answer the user, then feedback is given by the users to score the response, compute a reward, and update the policy. The reward function usually relies on objective criteria such as the success (or not) of the entire dialogue and the number of

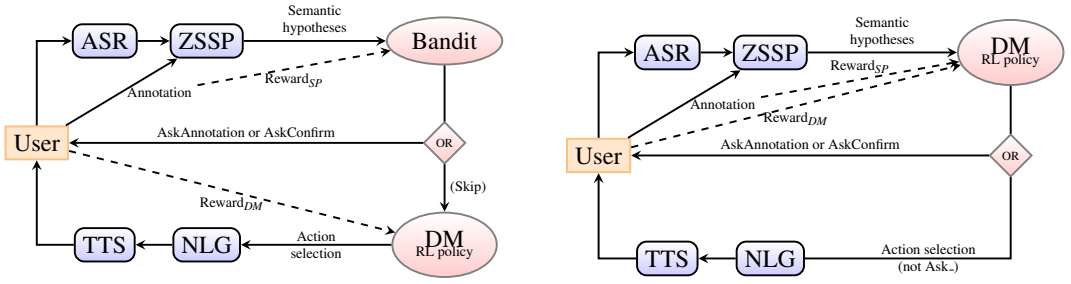

Figure 3. Configurations of BR *(left)* and RR *(right)* systems

turns. To take into account local rewards and further speed up the overall training of the system, we also considered reward shaping (Ng et al. 1999).

As described in (Ferreira and Lefèvre 2015), the reward function is here enhanced with a social reward:

$$R(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{social}(s_t, a_t, s_{t+1})$$

where R_{env} is an immediate reward function given by the environment (namely a penalty for each turn of the dialogue, aside for the last one which has a reward depending on the success or not of the dialogue), and R_{social} is a potential-based reward shaping function:

$$R_{social}(s_t, a_t, s_{t+1}) = \lambda \psi_{social}(s_{t+1}) - \psi_{social}(s_t)$$

with ψ_{social} a real-valued function (called the shaping potential function), defined here as a score given by the user at each turn. From the user's point of view, there are two different types of feedback. The global feedback is given by the user at the end of the dialogue, depending on whether the entire dialogue is a success or not. The social (or local) feedback $\psi_{social}(s_t)$ is provided at each turn t to score the last response only. At the end of the dialogue, the policy is updated according to the whole collected feedback.

In this work, $\lambda = 0.95$, the global feedback value is set to 20 in case of success, 0 otherwise. The penalty for each turn is set to -1 and the social feedback $\psi_{social}(s_t) \in \{-1, -0.5, 0, 0.5, 1\}$.

5. Joint on-line learning

In order to effectively learn the dialogue system on-line, the expert user needs to be able to both improve the SP model and the dialogue manager. Two different joint learning protocols are proposed to achieve it. Both protocols are illustrated in Figure 3.

The first one, referred to as **BR**[a] hereafter, directly juxtaposes the bandit to learn the ZSSP and the Q-learner RL approaches to learn the dialogue manager policy. An adversarial bandit algorithm (see Section 3) is applied for training ZSSP and a Q-learner (see Section 4) is used to learn the DM policy. The knowledge base of the ZSSP is updated after each dialogue turn (if the chosen action is not **Skip**) and the DM policy is updated at the end of each dialogue.

The second protocol, referred to as **RR**[b] hereafter, directly adds the ZSSP learning actions to the dialogue manager RL policy, and therefore combines the two learning processes into one single policy.

This variant of joint learning merges both policies in a single Q-learner. In that purpose, the DM summary state vector used by the policy was augmented with a ZSSP-related dimension. Let

[a]BR stands for **B**andit-SP and **R**L-DM learning protocol.

[b]RR means that the system is learned with an **R**L-SP and an **R**L-DM.

us note that only one dimension was added so as to limit the increase of the state size. At this point, it is handcrafted as no data is available to derive it optimally from raw observations.

This new dimension is evaluated from a set of quality indices of the annotations made by the ZSSP model. A 3-point scale is based upon five distinct features:

(1) **confidence** $[0, 1]$: confidence score of the semantic parser;

(2) **fertility** $[0, 1]$: ratio of concepts w.r.t. the utterance word length, since ZSSP tends to produce an over-segmentation of the incoming utterances with inserted concepts;

(3) **rare** $0, 1$: presence of rare concepts in the annotation. Rare concepts are "help", "repeat", "restart", "reqalts", "reqmore", "ack" or "thankyou", and are wrongly annotated in general;

(4) **known chunks** $[0, 1]$: ratio of annotated chunks available in the semantic knowledge base K among the total number of annotated chunks.

(5) **gap** $[0, \infty[$: the difference between the confidence scores of the 1-best and the 2-best annotations. Since those differences are very low (< 0.01), the natural logarithm is applied to break out the data in order to have more readable values.

From these features, the ZSSP-related DM state new dimension is computed as:

0 ***all clear***: rare $= 0$ and confidence ≤ 0.499 and fertility ≤ 0.4 and known chunks ≥ 0.5 and gap ≥ -5.5;

1 ***average condition***: rare $= 0$ and fertility ≤ 0.5 and known chunks ≥ 0.15 and gap ≥ -6.5 and (confidence > 0.499 or fertility > 0.4 or known chunks < 0.5 or gap < -5.5);

2 ***alarming***: rare $= 1$ or fertility > 0.5 or known chunks < 0.15 or gap < -6.5.

Under the RR protocol, the two ZSSP-annotation actions ("Askconfirm" and "Askannotation", see Section 2) are also included inside the list of summary actions that can be picked up by the dialogue policy. In such a case, the user is presented with the appropriate annotation window in the system's graphical interface and can correct the current annotation. Purely vocal interactions for this process are under study. Yet feasible, it remains a challenging task which could introduce errors of its own, so it seemed more appropriate to evaluate the whole process first with a graphical interface and no input errors. Once done, the turn is updated (i.e. the annotation process has taken the place of the normal user audio response) and the dialogue is pursued. Even though the policy might learn it by itself, we chose to inhibit two "Ask" actions in a row (they are tagged as impossible in the next turn), to avoid cycles of annotations. Finally, these two ZSSP-annotation actions have a specific social feedback: instead of -1, the feedback f_i uses the loss function $l(i)$ defined in Section 3 and rescaled to obtain a score in $[-1, 1]$:

$$f_i = (1 - l_i) \times 2 - 1 \quad .$$

Table 3. Example of a successful dialogue (translated from French).

S	So what about this picture?
U_asr	*I see a red apple.*
S	Yes. And what can you say about it?
U_asr	*It's upside-down.*
S	It seems to be upside-down. And it's humanised. Due to what?
U_asr	*Because of its eyes.*
S	So, it is red and has eyes. Does it make you think of someone? Or does it just look like a basic character?
U_asr	*It looks like spiderman.*
S	Indeed, I think this red apple looks like spiderman. It seems it has eyes. It is upside-down.
U_asr	*Can we talk about the message?*
S	I think this is related to GMOs.
U_asr	*Ah yes, maybe.*
S	And GMOs are genetically modified, like several super-heroes.
U_asr	*Yes, they are. Goodbye.*

6. Experimental study

6.1 Task description

Experiments presented in this paper concern a chit-chat dialogue system framed in a goal-oriented dialogue task (Riou et al. 2019)[c]. In this context, users discussed with the system about an image (out of a small predefined set of 6), and jointly tried to discover the message conveyed by the image, as described in (Chaminade 2017).

The experiments reported hereafter are a step towards a more global objective to develop a system for a neuroscience evaluation of human communication social skills. Inside an fMRI, users will interact with a robotic platform, vocally powered by the presented system, which is live-recorded and displayed inside the head-antenna. These experiments will be performed in French on a new task for which no data are yet available. Therefore, it contrasts with the situation where publicly available corpora can be used (Williams et al. 2013; Casanueva et al. 2017). Likewise, crowd-sourcing is not affordable to realise large-scale data collection, as after a few attempts it seems that no platform (Amazon Mechanical Turk or others) offers enough NLP-skilled workers with a good command of French.

In order to use a goal-oriented system for such a task, the principle which was followed was to construct, as the system's back-end, a database containing several hundreds of possible combinations of characteristics of the image, each associated with a hypothesis of the conveyed message. During her interaction with the system, it is expected that the user progressively provides elements from the image matching entities in the database. This makes the system select a small subset of possible entities from which it can pick both additional characteristics to inform the user with, and

[c]The datasets generated during the current study are available to the research community upon request to the authors.

ultimately a pre-defined message to utter as a plausible explanation for the image purpose. This allows the user to speak rather freely about the image for several tens of seconds before arguing briefly about the message. The discussion is expected to last around one minute at most.

The task-dependent knowledge base used in the experiments is derived from the neuroscience task description of fruit images (Chaminade 2017), as well as from a generic dialogue information task. The semantics of the domain is represented by 16 different act types, 9 slots and 51 values. The lexical forms used in the natural language generation module to render act types were manually elaborated. A total of around 150 surface forms are initially edited for the basic acts (e.g. `inform(fruit=$A)` $\rightarrow$ `'Right, that's an $A'`), with on average 2-3 variants per act that the system will pick out randomly. They are automatically combined to produce the possible more complex acts (e.g. `inform(fruit=$A, seems=$B, possesses=$C)` $\rightarrow$ `'That $A with $C seems rather $B'`). The final template database amounts to 27k distinct forms. An example of a representative dialogue for the task is given in Table 3.

Although we are fully aware that the experimental setup is rather distant from more classic tasks of slot-filling and database retrieval, the dialogue system was designed as a goal-oriented system and is eventually close to characteristics expected inside a dialogue platform for more ordinary tasks. Our dialogue system takes into account 13 dialogue acts and 9 slots, which are numbers comparable to what can be found in previous task-oriented systems (e.g. Budzianowski et al. (2018) mention between 6 and 16 act types and, between 3 and 15 slots used in seven domains).

6.2 User interface

To handle joint learning of SP and DM, a dedicated user interface has been developed. Basically, it is a web-based interface remotely connected to the dialogue system (see Figure 8). It starts with displaying an information content, then the user can initiate the dialogue and an image appears (randomly selected from the fruit database).

Before the dialogue starts, the user is prompted with a panel explaining the whole setup. Among the instructions the notion of successful dialogue is detailed:
"Once the conversation is over (either you or the system said "bye"), an evaluation board will appear at the page bottom. A **successful dialogue** must respect:

- at least two image's features have been mentioned (no matter who proposed them);
- the system has given its opinion on the message;
- no major failure occurred;
- and all this in less than a minute!

A new dialogue can be started after the board submission."

During the dialogue the user is given two possibilities to send feedbacks to the system:

- either through a pop-up window for SP annotations (see Figure 9); in that case the bandit algorithm for ZSSP adaptation, as described in 3, chooses when a pop-up is triggered and what kind of annotation is asked (only binary feedbacks or full manual annotation);
- or with social rewards for DM reinforcement (see Figure 10); in that case the user decides at each step to use it or not, and when needed to select the value of the reward (in $[-2, 2]$).

After the dialogue has been concluded by either participant, a survey is proposed to collect meta-data about the success of the interactions and ratings on various dimensions (see Figure 11 of Annex 1).

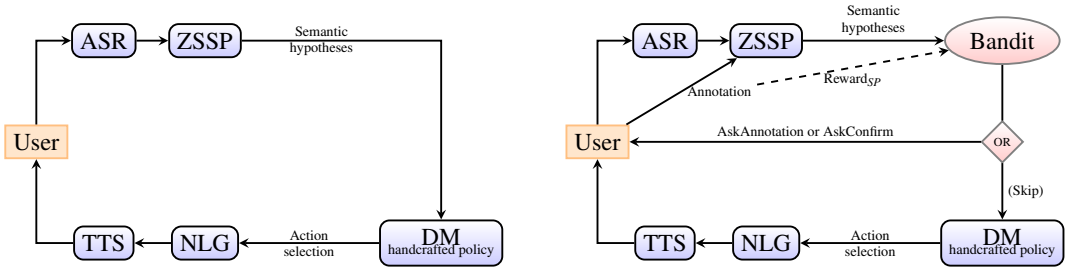

Figure 4. Configurations of ZH *(left)* and BH *(right)* systems

6.3 Results

The evaluation of the two joint learning approaches is presented here. Two complementary systems, representing the separate learning classical approach, are proposed in comparison with **BR** and **RR** (see Section 5): **ZH** is a baseline system without on-line learning using the initial ZSSP and a handcrafted dialogue manager policy[d], whereas the system **BH** combines the bandit on-line learning for ZSSP and the handcrafted dialogue manager policy. They are both illustrated in Figure 4.

For each system, a dialogue system expert user (with a good knowledge of the expected system behaviour and the target task) communicated with the system to train a model. Then a group of 13 (mostly) naive users has been recruited to test the models. They were totally unaware of which model they were testing, and models are presented in random order. Due to their availability some have tested several systems, by groups of 12 dialogues per trials. At the end of each session, the users were asked to rate on a scale of 0 (worst) to 5 (best) the understanding and generation qualities of the system. The amount of training dialogues as well as the number of test sets for each configuration are given in Table 5. Let us note that the number of dialogues (80) to train BH was reduced w.r.t. BR and RR (140), since BH relies on a handcrafted policy, fixed throughout the training phase. Figure 5 (a) shows that for BH the training average success rate is close to 100 % after only 60 dialogues.

Table 5. Evaluation of the different configurations of joint on-line learning

Model	**Train**	**Test**	**Success**	**Avg cum.**	**Sys. Underst. Rate**		**Sys. Gener. Rate**	
	(#dial)	**(#dial)**	**(%)**	**Reward**	Mean	SD	Mean	SD
ZH	0	142	29	-1.9	1.6	1.5	4.0	1.4
BH	80	96	70	7.0	3.2	1.4	4.6	0.7
BR	140	96	89	10.9	3.3	1.6	4.6	0.7
RR	140	96	65	4.4	2.9	1.3	3.8	1.1

The user trials of the training trials for each protocol are given in Table 5. For rates computed from the evaluations made by the 13 subjects, mean and standard deviation are provided (last four columns). The results show that the different configurations of the trained system display acceptable performance. The BR model trained with 140 dialogues shows the best success rate (89%) and significantly[a] over-performs all other models. Moreover, the ZH model leads to significantly[a] lower success values than all other models. The difference in performance between the ZH and the BH models (+41 points) shows the impact of the ZSSP adaptation on the overall success of

[d]The handcrafted policy used in the system is in line with the description given in sec. 4.4 of Young et al. (2009).

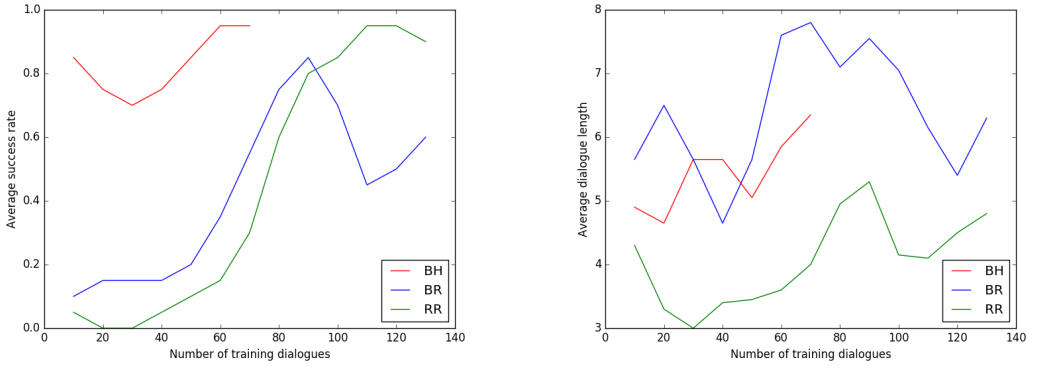

Figure 5. **(a)** Moving average success rates **(b)** Moving average dialogue lengths

the conversation, along with a better understanding (1.6 for ZH vs. 3.2 for BH). Dialogue success annotations have been manually screened after the experiments and no errors w.r.t. to the guidelines given to test users have been noticed.

The average cumulative reward for the test is directly correlated with the success rate and confirms the previous findings. Besides, due to a well-tuned template-based generation system, the system generation quality rate is high (≥ 3.8) for all configurations. The RR protocol offers a success rate smaller than BH and BR (65% for RR vs. 89% for BR). RR has been added to the study as it is expected to save some development time, as well as tuning cost in operation. But to achieve this, RR must deal with inputs from both tasks (SP and DM) and process them appropriately to balance the annotation decisions. Clearly the way RR is currently elaborated failed to address this requirement, as will be discussed in more details in Section 6.4.5.

6.4 Training analysis

In order to improve our training protocol and to get some insights from our experiments, the training logs were analysed. Only one training is presented for each model in the plots, but in reality several training attempts have been performed by 4 different experts.[e] After the training process, each expert was asked to describe his training strategy in term of dialogue complexity, utterance lengths and usage of feedbacks turns. Those feedbacks were also analysed in this study. While not statistically significant, these attempts allow us to get a better idea of the properties of on-line training process and the strategy implemented by the expert trainers. The experts were quite free in their training choices. For instance, they did not have restrictions on how they used the additional feedbacks.

6.4.1 Dialogue complexity: length

The dialogue complexity is dependent on various factors. Moreover these factors are not independent. Yet we propose to study the impact of dialogue complexity through two main factors: total length in number of turns and amount of exchanged concepts. They are presented separately, in

[a]Statistical significance was analysed with a two-tailed Welch's t-test. Results were considered statistically significant with a p-value < 0.001.

[e]In fact, for each condition's system, two expert users trained two different systems. Due to the difficulty to muster volunteers for the evaluation, only a subset of users was asked to evaluate the two versions of each condition's system. These results are not extensively presented in the paper to avoid interpretation complexity to readers. However, the results are very comparable for the two versions of each system.

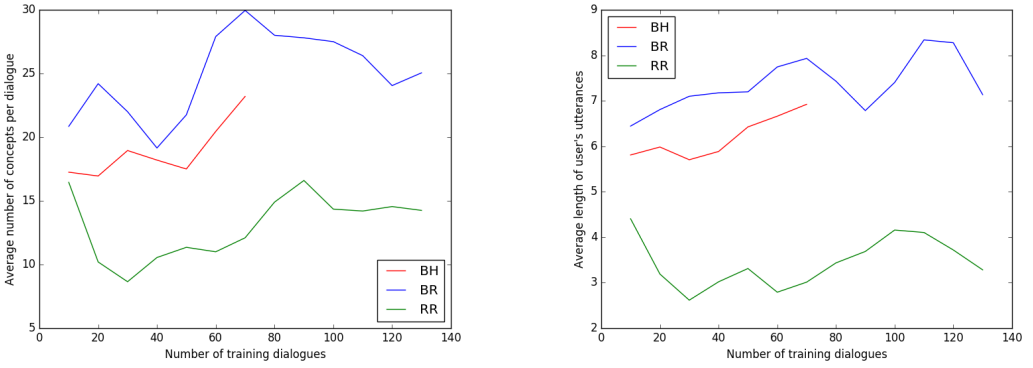

Figure 6. **(a)** Moving average number of concepts per dialogue **(b)** Moving average utterance lengths during the training phase for different models

this section and the next, as we posit that the former is better correlated to the evolution of the DM learning when the latter is strongly linked to the SP capacity. Regarding the dialogue manager policy, all experts agree that they preferred to first use simple dialogues to build an efficient dialogue manager policy. Then, when the system started to be usable (regular sequences of 2-3 successful dialogues), more sophisticated dialogues were involved in reaching another step in the system capability. These choices can be seen in the training logs. In Figure 5 (a) and (b), it can be observed that the success rate and the dialogue length tend to vary a lot during training:

- at the onset of the learning process, dialogues are shorter (with a minimum of 3 turns for RR and around 6 turns for BH and BR) while the success rate is increasing;
- then, after 40 to 60 dialogues, they become more complex, so that the process drifts towards a decrease of the reward and success rates, and a production of longer dialogues.

These two phases can appear periodically during training. After a phase of complex dialogues, which tends to unsettle the dialogue policy, experts can go back to more simple dialogues for a while. Let us note that with the BH model, the experts generate sophisticated dialogues faster, since this model does not require to learn a dialogue manager policy in parallel.

6.4.2 Dialogue complexity: concepts

To further analyse ZSSP learning, the total number of concepts exchanged per dialogue during the learning process has been considered. To count the exchanged concepts, each triplet of act, slot and value has been considered as one concept. The results are shown in Figure 6 (a). Experts start their training trying their ideal target dialogue. When this one does not succeed, they tend to focus on simpler dialogues with a limited number of concepts. Then, when the system improves, they also diversify the concepts they are using. RR presents fewer concepts per dialogue than BH and BR. This can be explained by the very low triggering level of ZSSP learning actions for this model, therefore the experts cannot efficiently extend the concepts used during a dialogue, and then tend to be conservative in their expressions (and not to introduce too much noise).

This turn complexity can also be observed with the average user's utterance lengths. The results are shown in Figure 6 (b). The experts tend to augment their utterance lengths while the system improves, with the exception of RR; in the latter case, the experts tend to have shorter and simpler utterances since ZSSP training is of lower quality.

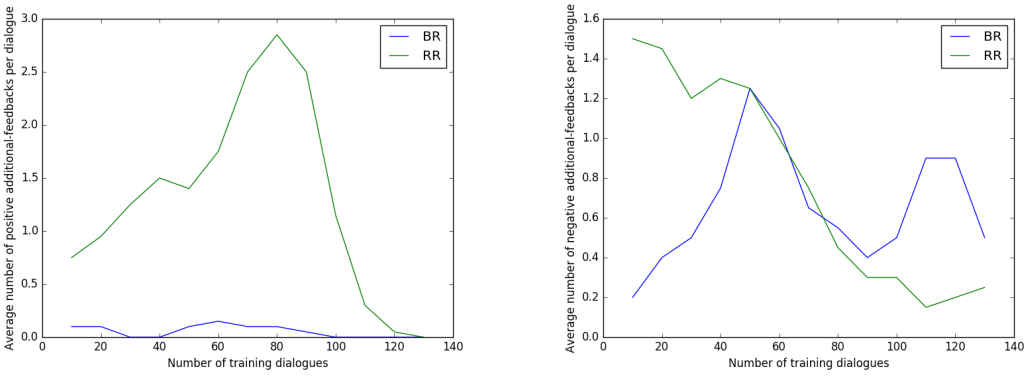

Figure 7. **(a)** Moving average number of additional positive feedbacks per dialogue **(b)** Moving average number of additional negative feedbacks per dialogue during the training phase for different models

6.4.3 Dialogue failures

In order to find possible improvements of the training process, the common causes of dialogue failure were investigated. 48 test dialogues were thoroughly analysed. A large difference of success rates between the models can be explained by the SP errors. An expert was asked to annotate the correct SP outputs for each user input in the test dialogues subset. In addition, an annotation was added when the SP error was due to erroneous speech recognition. Thus, an SP success score was computed as the ratio of the number of correct concepts over the total number of reference concepts plus the number of inserted concepts. ZH presents an SP success rate of 41% against 70% for BH, 60% for BR and 49% for RR. Those variations can explain the differences of success rates between the different models.

When looking at specific examples of SP errors (see Table 7), we observe that some of them are common causes of dialogue failures, such as:

- a false recognition of a "goodbye" act, often due to ASR errors, causing the dialogue to stop prematurely;
- SP errors creating false beliefs in the dialogue manager, leading to deviate from the image currently being discussed. Those errors are sometimes due to speech errors: about 10% for ZH, 25% for BH, 28% for BR and 27% for RR. But these differences are to be linked to the SP success rate: a better SP implies fewer errors, so the proportion of SP errors caused by ASR errors increases.

Moreover, negation utterances are still diversely handled by the SP systems. For example "this is not a lemon" is not understood in ZH and RR models, but usually is in BH and BR. However, "this is not a lemon, this is an apple" remains difficult to address for all models.

6.4.4 Use of turn feedbacks for DM training

The additional feedbacks allow the experts to locally reward or penalise specific system responses, in complement to the standard overall reward (task success penalised by length). In (Ferreira and Lefèvre 2015), it has been shown in a simulated environment that negative feedbacks can better guide the dialogue training than positive ones.

During interviews with the experts after the trainings, they indicate that they seem instinctively more motivated to penalise wrong responses than to reward correct ones, which is a good thing

Table 7. Examples of impacting SP errors

ASR (French)	il y a des bras des dieux et des jambes
ASR (English)	there are arms gods and legs
	There is an ASR error due to the phonetic proximity of dieux *(gods) and* yeux *(eyes).*
SP	inform(looks_like=superhero,possesses=legs,possesses=arms)
	The SP misinterpreted gods *with the concept of* superhero.
ASR (French)	le fruit a l'air d'avoir baissé les bras
ASR (English)	the fruit seems to have given up
	The expression baisser les bras *(to give up) can be translated literally into* lower the arms.
SP	inform(possede=bras), reqalts()
	Idioms are often difficult to interpret, leading to SP errors. Furthermore, the SP sometimes tends to add acts such as reqalts() *or* ack() *(especially for ZH and RR).*

as a previous study showed in simulations that they were more profitable to the learning process (Ferreira and Lefèvre 2015). However, some experts report they insisted on positive rewards in order to force the system to retain good behaviours. Figure 7 shows the uses of both types of feedbacks: negative feedbacks are regularly used for BR and RR (BH policy is handcrafted and does not imply user feedbacks). On the contrary, the use of positive feedbacks is more diverse. BR used almost none while RR used more positive than negative feedbacks. While emitting a lot of positive feedbacks requires more efforts from the expert, it also increases the stability of right choices in the dialogue policy.

6.4.5 Comparison of joint learning protocols

The main motivation for the RR approach is to rely on a single algorithm. It is expected to save some development time, as well as tuning cost in operation. Yet the two tasks (SP and DM) have distinct representations and a single MDP fails to address both in a same process.

From the training logs, it can be observed that low performance of RR seems to be related to the very low triggering level of the ZSSP learning actions after the exploration steps during RR w.r.t. the use of the bandit in BH and BR. As a consequence, in RR, the SP learning clearly lagged behind the DM learning, although at onset it is of paramount importance. To remedy this, the policy state space could be modified to take better account of the situations which should lead to ZSSP actions, while preserving its capacities of discrimination for the dialogue actions. Despite our endeavours to add some dedicated indicators in the summary state vectors (see Section 5), it failed to influence enough the value function and policy. But introducing more SP-related features in the Q-learning process is delicate as increasing the size of the state representation vector will impact the sample efficiency of the training algorithm. So in this regard new RL algorithms could be investigated to address the modelisation issue, while maintaining the learning efficiency in the context of very few data.

7. Conclusion

After proposing methods to interactively train both semantic parsing and dialogue management on-line, this paper proposed and evaluated ways to combine them in a joint learning process. Experiments have been carried out in real conditions and are therefore scarce. Yet it has been

possible to show that simultaneous learning can be operated, and that after roughly a hundred dialogues the performance of the various configurations tested were generally good enough compared to a handcrafted system. A combination of Bandit for SP and RL/Q-learning for DM oversteps an integrated approach using only RL/Q-learning. While more desirable in terms of development complexity, the latter suffers from merging into a single framework two decision processes based on different ground features.

In any case both configurations offer some insights on the characteristics of the on-line learning process. For instance, in our experiments the experts tend to focus first on simple dialogues to increase success rates, then they try to produce more complex dialogues to both improve the dialogue manager policy and the semantic parser. Our experimental study shows that the use of additional feedbacks at the turn level helps the training. Even if both negative and positive feedbacks can be used, negative ones are commonly preferred to guide the learning process. We also observed that at the end of the training process, most dialogue failures are due to ASR and SP errors. A competitive ASR system remains critical to train an efficient dialogue system.

Based on these results, merging the resulting policies between trials is the next challenge, so as to be able to stack training data coming from different users and save even more time to the system developers. Another research lead is to allow experts to annotate back their data above the 1-turn current limit. But it would become a post-processing technique, for which the benefit-cost ratio must be evaluated.

Acknowledgements. This work has been partially supported by grants ANR-16-CONV-0002 (ILCB), ANR-11-LABX-0036 (BLRI).

References

Bapna, A., **Tür, G.**, **Hakkani-Tür, D.**, **and Heck, L.** 2017. Towards zero-shot frame semantic parsing for domain scaling. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2017, pp. 2476–2480, Stockholm, SW. ISCA.

Bian, J., **Gao, B.**, **and Liu, Y.** 2014. Knowledge-Powered Deep Learning for Word Embedding. In **T. Calders et al.**, editor, ECML PKDD, pp. 132–148. Springer-Verlag Berlin Heidelberg.

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. arXiv.org.

Casanueva, I., **Budzianowski, P.**, **Su, P.-H.**, **Mrkšić, N.**, **Wen, T.-H.**, **Ultes, S.**, **Rojas-Barahona, L.**, **Young, S.**, **and Gašić, M.** 2017. A Benchmarking Environment for Reinforcement Learning Based Task Oriented Dialogue Management. arXiv.org.

Chaminade, T. 2017. An experimental approach to study the physiology of natural social interactions. Interaction Studies, 18(2):254–275.

Chang, C., **Yang, R.**, **Chen, L.**, **Zhou, X.**, **and Yu, K.** 2017. Affordable On-line Dialogue Policy Learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, pp. 2190–2199.

Chen, L., **Yang, R.**, **Chang, C.**, **Ye, Z.**, **Zhou, X.**, **and Yu, K.** 2017a. On-line Dialogue Policy Learning with Companion Teaching. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2:198–204.

Chen, L., **Zhou, X.**, **Chang, C.**, **Yang, R.**, **and Yu, K.** 2017b. Agent-Aware Dropout DQN for Safe and Efficient On-line Dialogue Policy Learning. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2454–2464.

Daubigney, L., **Geist, M.**, **Chandramohan, S.**, **and Pietquin, O.** 2012. A Comprehensive Reinforcement Learning Framework Framework for Dialogue Management Optimization. Journal of Selected Topics in Signal Processing, 6(8):891–902.

Dauphin, Y. N., **Tur, G.**, **Hakkani-Tur, D.**, **and Heck, L.** 2014. Zero-Shot Learning for Semantic Utterance Classification. arXiv.org.

Deoras, A. and Sarikaya, R. 2013. Deep Belief Network based Semantic Taggers for Spoken Language Understanding. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, , INTERSPEECH 2013, Lyon, France.

Devlin, J., **Chang, M.-W.**, **Lee, K.**, **and Toutanova, K.** 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.org.

Dhingra, B., **Li, L.**, **Li, X.**, **Gao, J.**, **Chen, Y.-N.**, **Ahmed, F.**, **and Deng, L.** 2017. Towards End-to-End Reinforcement

Learning of Dialogue Agents for Information Access. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pp. 484–495, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ferreira, E., **Jabaian, B.**, **and Lefèvre, F.** 2015. Online adaptative zero-shot learning spoken language understanding using word-embedding. In Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, pp. 5321–5325, South Brisbane, Queensland, Australia. IEEE.

Ferreira, E. and Lefèvre, F. 2013a. Expert-based reward shaping and exploration scheme for boosting policy learning of dialogue management. In Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013, pp. 108–113, Olomouc, Czech Republic. IEEE.

Ferreira, E. and Lefèvre, F. 2013b. On the use of social signal for reward shaping in reinforcement learning for dialogue management. In Proceedings of the 17th Workshop on the semantics and pragmatics of dialogue, {SemDial}, Amsterdam, NL.

Ferreira, E. and Lefèvre, F. 2013c. Social signal and user adaptation in reinforcement learning-based dialogue management. In 2nd Workshop on Machine Learning for Interactive Systems - Bridging the Gap Between Perception, Action and Communication, MLIS-IJCAI 2013, pp. 61–69, Beijing, China. {ACM} Press.

Ferreira, E. and Lefèvre, F. 2015. Reinforcement-learning based dialogue system for human-robot interactions with socially-inspired rewards. Computer Speech & Language, Special issue on Speech and Language for Interactive Robots, 34(1):256–274.

Ferreira, E., **Masson, A. R.**, **Jabaian, B.**, **and Lefèvre, F.** 2016. Adversarial bandit for online interactive active learning of zero-shot spoken language understanding. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, pp. 6155–6159, Shanghai, China.

Gašic, M., **Breslin, C.**, **Henderson, M.**, **Kim, D.**, **Szummer, M.**, **Thomson, B.**, **Tsiakoulis, P.**, **and Young, S.** 2013. On-line Policy Optimisation of Bayesian Spoken Dialogue System via Human Interaction. In Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP.

Gašić, M., **Lefèvre, F.**, **Jurčíček, F.**, **Keizer, S.**, **Mairesse, F.**, **Thomson, B.**, **Yu, K.**, **and Young, S.** 2009. Back-off action selection in summary space-based POMDP dialogue systems. In **IEEE**, editor, Proceedings of 2009 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU, pp. 456–461, Merano, Italy. IEEE.

Geist, M. and Pietquin, O. 2010a. Kalman Temporal Differences. Journal of Artificial Intelligence Research, 39:483–532.

Geist, M. and Pietquin, O. 2010b. Managing Uncertainty within Value Function Approximation in Reinforcement Learning. In Active Learning and Experimental Design workshop (collocated with AISTATS 2010).

Hahn, S., **Dinarelli, M.**, **Raymond, C.**, **Lefèvre, F.**, **Lehnen, P.**, **de Mori, R.**, **Moschitti, A.**, **Ney, H.**, **and Riccardi, G.** 2011. Comparing Stochastic Approaches to Spoken Language Understanding in Multiple Languages. IEEE Transactions on Audio, Speech, and Language Processing, 19(6):1569–1583.

Hancock, B., **Bordes, A.**, **Mazaré, P.-E.**, **and Weston, J.** 2019. Learning from Dialogue after Deployment: Feed Yourself, Chatbot! arXiv.org.

Koehn, P. and Germann, U. 2014. The Impact of Machine Translation Quality on Human Post-editing. In Proceedings of Workshop on Humans and Computer, pp. 38–46, Gothenburg, Sweden. Association for Computational Linguistics.

Le, H., **Vial, L.**, **Frej, J.**, **Segonne, V.**, **Coavoux, M.**, **Lecouteux, B.**, **Allauzen, A.**, **Crabbé, B.**, **Besacier, L.**, **and Schwab, D.** 2019. FlauBERT: Unsupervised Language Model Pre-training for French. arXiv.org.

Lefèvre, F. and de Mori, R. 2007. Unsupervised state clustering for stochastic dialog management. In Proceedings of 2007 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU, pp. 550–555, Kyoto, Japan. IEEE.

Lemon, O. 2011. Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation. Computer Speech and Language, 25(2):210–221.

Li, J., **Miller, A. H.**, **Chopra, S.**, **Ranzato, M.**, **and Weston, J.** 2016. Dialogue Learning With Human-In-The-Loop. arXiv.org.

Li, X., **Chen, Y.-N.**, **Li, L.**, **Gao, J.**, **and Celikyilmaz, A.** 2017. End-to-End Task-Completion Neural Dialogue Systems. arXiv.org.

Mikolov, T., **Chen, K.**, **Corrado, G.**, **and Dean, J.** 2013. Efficient Estimation of Word Representations in Vector Space. arXiv.org.

Ng, A., **Harada, D.**, **and Russell, S.** 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In Proceedings of the Sixteenth International Conference on Machine Learning, ICML '99, pp. 278–287.

Padmakumar, A., **Thomason, J.**, **and Mooney, R. J.** 2017. Integrated Learning of Dialog Strategies and Semantic Parsing. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pp. 547–557.

Pietquin, O., **Geist, M.**, **and Chandramohan, S.** 2011. Sample Efficient On-line Learning of Optimal Dialogue Policies with Kalman Temporal Differences. In Proceedings of the 2011 International Joint Conference on Artificial Intelligence, IJCAI, Barcelona (Spain).

Radford, A. and Salimans, T. 2018. Improving Language Understanding by Generative Pre-Training. OpenAI Preprint, pp. 1–12.

Rastogi, A., **Gupta, R.**, **and Hakkani-Tur, D.** 2018. Multi-task Learning for Joint Language Understanding and Dialogue State Tracking. In Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, pp. 376–384, Melbourne, Australia. Association for Computational Linguistics.

Riou, M., **Jabaian, B.**, **Huet, S.**, **Chaminade, T.**, **and Lefèvre, F.** 2017. Integration and evaluation of social competences such as humor in an artificial interactive agent. In Proceedings of the 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents - ISIAA 2017. ACM Press.

Riou, M., **Jabaian, B.**, **Huet, S.**, **and Lefèvre, F.** 2019. Lilia, A Showcase for Fast Bootstrap of Conversation-Like Dialogues Based on a Goal-Oriented System. In **Martin-Vide, C.**, editor, Statistical Language and Speech Processing, 13, Lubjana, Slovenia. Springer Nature Switzerland AG.

Rojas-Barahona, L. M., **Ultes, S.**, **Budzianowski, P.**, **Casanueva, I.**, **Gasic, M.**, **Tseng, B.-H.**, **and Young, S.** 2018. Nearly Zero-Shot Learning for Semantic Decoding in Spoken Dialogue Systems. arXiv.org.

Shah, P., **Hakkani-T, D.**, **and Liu, B.** 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In Proceedings of NAACL-HLT 2018, pp. 41–51. Association for Computational Linguistics.

Su, P.-H., **Gasic, M.**, **Mrksic, N.**, **Rojas-Barahona, L.**, **Ultes, S.**, **Vandyke, D.**, **Wen, T.-H.**, **and Young, S.** 2016. On-line Active Reward Learning for Policy Optimisation in Spoken Dialogue Systems. In 54th Annual Meeting of the Association for Computational Linguistics, pp. 2431–2441, Berlin, Germany.

Tur, G., **Hakkani-Tür, D.**, **and Schapire, R. E.** 2005. Combining active and semi-supervised learning for spoken language understanding. Speech Communication.

Ultes, S., **Rojas Barahona, L. M.**, **Su, P.-H.**, **Vandyke, D.**, **Kim, D.**, **Casanueva, I.**, **Budzianowski Pawełand Mrkšić, N.**, **Wen, T.-H.**, **Gasic, M.**, **and Young, S.** 2017. PyDial: A Multi-domain Statistical Dialogue System Toolkit. In Proceedings of ACL 2017, System Demonstrations, pp. 73–78. Association for Computational Linguistics.

Upadhyay, S., **Faruqui, M.**, **Tür, G.**, **Hakkani-Tür, D.**, **and Heck, L.** 2018. (ALMOST) ZERO-SHOT CROSS-LINGUAL SPOKEN LANGUAGE UNDERSTANDING. In 2018 {IEEE} International Conference on Acoustics, Speech and Signal Processing ({ICASSP}).

Wang, F. and Swegles, K. 2013. Modeling user behavior online for disambiguating user input in a spoken dialogue system. Speech Communication, 55(1):84–98.

Wen, T.-H., **Miao, Y.**, **Blunsom, P.**, **and Young, S.** 2017. Latent Intention Dialogue Models. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 10, Sydney, Australia. PMLR.org.

Wen, T.-H., **Vandyke, D.**, **Mrkšić, N.**, **Gašić, M.**, **M. Rojas-Barahona, L.**, **Su, P.-H.**, **Ultes, S.**, **Young, S.**, **Mrksic, N.**, **Gasic, M.**, **Rojas-Barahona, L. M.**, **Su, P.-H.**, **Ultes, S.**, **Young, S.**, **Wen**, **Wen, T.-H.**, **Vandyke, D.**, **Mrksic, N.**, **Gasic, M.**, **Rojas-Barahona, L. M.**, **Su, P.-H.**, **Ultes, S.**, **and Young, S.** 2016. 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, EACL 2016, Address: Valencia, Spain.

Williams, J., **Raux, A.**, **Ramachandran, D.**, **and Black, A.** 2013. The Dialog State Tracking Challenge. In Proceedings of the SIGDIAL 2013 Conference, 404–413, Metz, France. Association for Computational Linguistics.

Wolf, T., **Sanh, V.**, **Chaumond, J.**, **and Delangue, C.** 2019. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. arXiv.org.

Yang, X., **Chen, Y.-N.**, **Hakkani-Tur, D.**, **Crook, P.**, **Li, X.**, **Gao, J.**, **and Deng, L.** 2017. End-to-end joint learning of natural language understanding and dialogue manager. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5690–5694. IEEE.

Ye, H., **Li, W.**, **and Wang, L.** 2019. Jointly Learning Semantic Parser and Natural Language Generator via Dual Information Maximization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL, 2090–2101, Florence, Italy. Association for Computational Linguistics.

Young, S., **Gašic, M.**, **Keizer, S.**, **Schatzmann, J.**, **Thomson, B.**, **and Yu, K.** 2009. The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. Computer Speech and Language, 24(2):150–174.

Zhao, T. and Eskenazi, M. 2018. Zero-Shot Dialog Generation with Cross-Domain Latent Actions. In Proceedings of the SIGDIAL 2018 Conference, Melbourne, Australia. Association for Computational Linguistics.

Zhao, T., **Xie, K.**, **and Eskenazi, M.** 2019. Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models. arXiv.org.

Annex 1: User interface

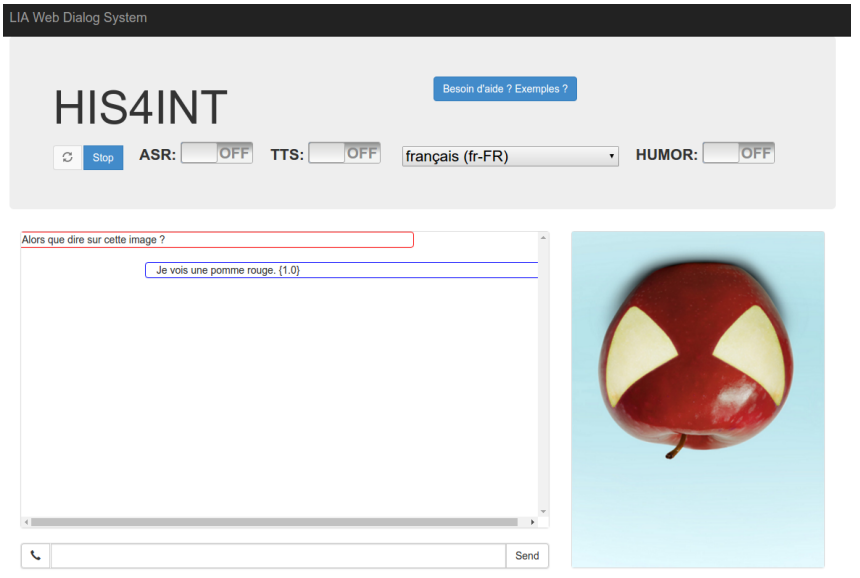

Figure 8. User interface: the dialogue window

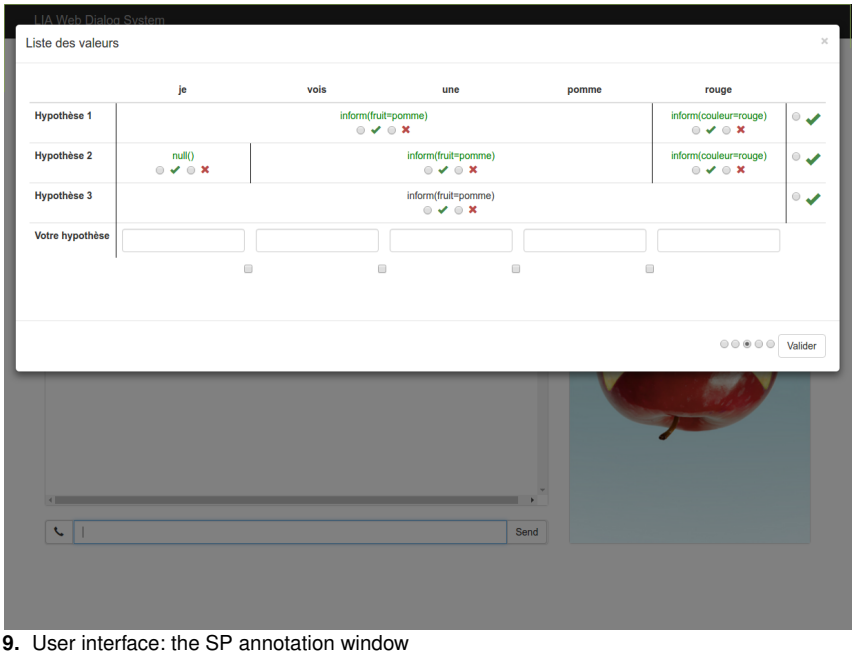

Figure 9. User interface: the SP annotation window

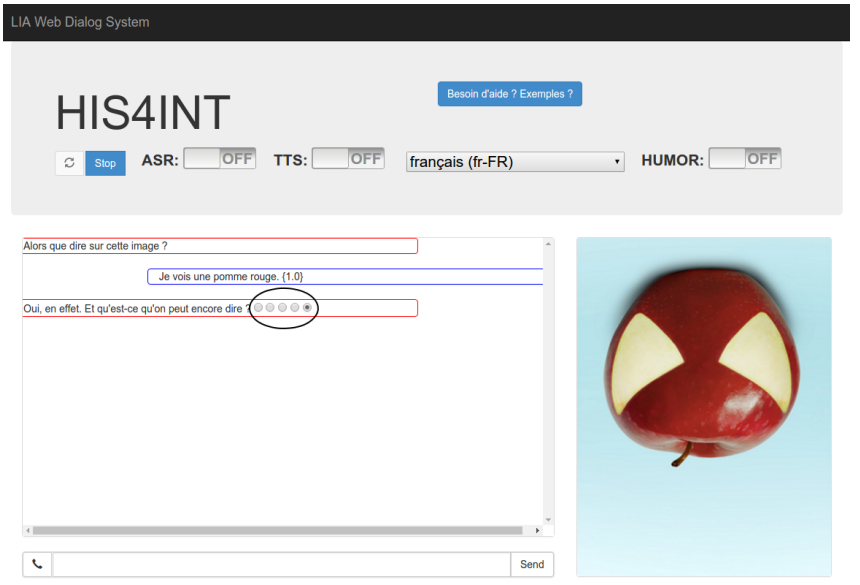

Figure 10. User interface: turn feedback for social reward of DM

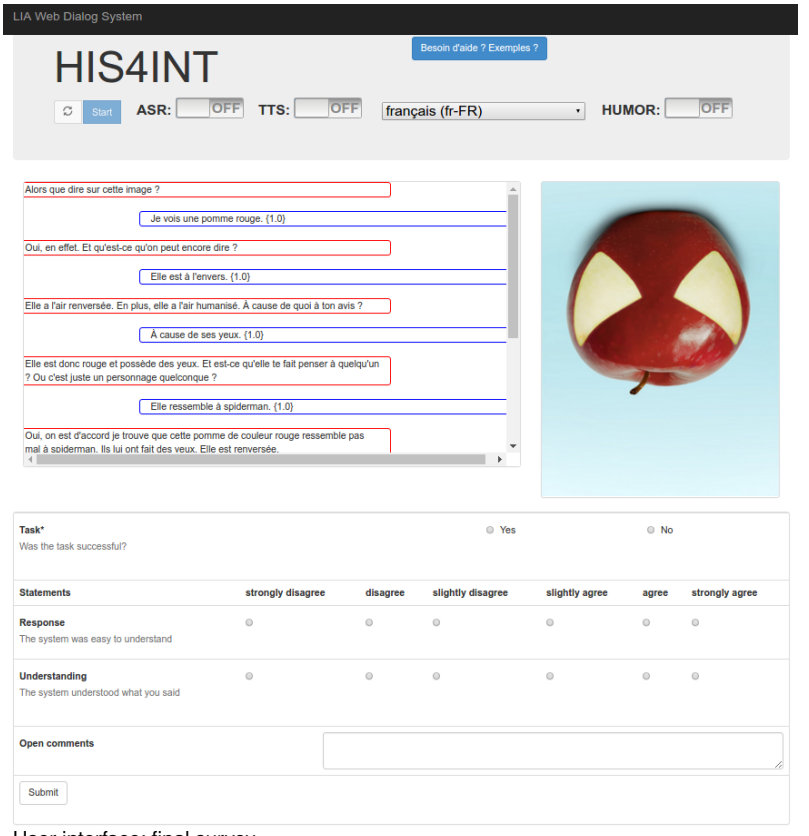

Figure 11. User interface: final survey