Beyond task success: A closer look at jointly learning to see, ask, and GuessWhat

Ravi Shekhar[†], **Aashish Venkatesh**[*], **Tim Baumgärtner**[*], **Elia Bruni**[*],
Barbara Plank[♥], **Raffaella Bernardi**[†] and **Raquel Fernández**[*]
[†]University of Trento, [*]University of Amsterdam,[♥]IT University of Copenhagen
bplank@itu.dk raffaella.bernardi@unitn.it raquel.fernandez@uva.nl

Abstract

We propose a grounded dialogue state encoder which addresses a foundational issue on how to integrate visual grounding with dialogue system components. As a test-bed, we focus on the *GuessWhat?!* game, a two-player game where the goal is to identify an object in a complex visual scene by asking a sequence of yes/no questions. Our visually-grounded encoder leverages synergies between guessing and asking questions, as it is trained jointly using multi-task learning. We further enrich our model via a cooperative learning regime. We show that the introduction of both the joint architecture and cooperative learning lead to accuracy improvements over the baseline system. We compare our approach to an alternative system which extends the baseline with reinforcement learning. Our in-depth analysis shows that the linguistic skills of the two models differ dramatically, despite approaching comparable performance levels. This points at the importance of analyzing the linguistic output of competing systems beyond numeric comparison solely based on task success.[1]

1 Introduction

Over the last few decades, substantial progress has been made in developing dialogue systems that address the abilities that need to be put to work during conversations: Understanding and generating natural language, planning actions, and tracking the information exchanged by the dialogue participants. The latter is particularly critical since, for communication to be effective, participants need to represent the state of the dialogue and the common ground established through the conversation (Stalnaker, 1978; Lewis, 1979; Clark, 1996).

In addition to the challenges above, dialogue is often situated in a perceptual environment. In this study, we develop a dialogue agent that builds a representation of the context and the dialogue state by integrating information from both the *visual* and *linguistic* modalities. We take the *GuessWhat?!* game (de Vries et al., 2017) as our test-bed, a two-player game where a Questioner faces the task of identifying a target object in a visual scene by asking a series of yes/no questions to an Oracle. We model the agent in the Questioner's role.

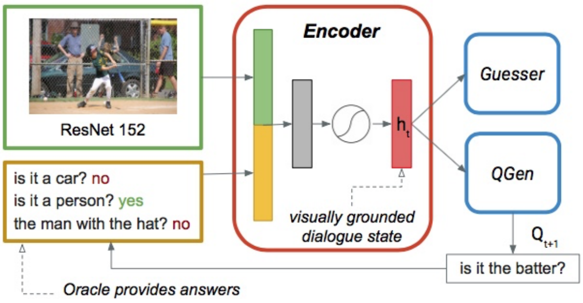


Figure 1: Our questioner model with a single visually grounded dialogue state encoder.

To model the Questioner, previous work relies on two independent models to learn to ask questions and to guess the target object, each equipped with its own encoder (de Vries et al., 2017; Strub et al., 2017; Zhu et al., 2017; Lee et al., 2017; Shekhar et al., 2018; Zhang et al., 2018). We propose an end-to-end architecture with a single *visually-grounded dialogue state encoder* (cf. Figure 1). Our system is trained jointly in a supervised learning setup, extended with a cooperative learning (CL) regime: By letting the model play the game with self-generated dialogues, the components of the Questioner agent learn to better perform the overall Questioner's task in a cooperative manner. Das et al. (2017b) have explored the use of CL to train two visual dialogue agents that receive joint rewards when they play a game successfully. To our knowledge, ours is the first approach where cooperative learning is applied to the internal com-

[1] Equal contribution by R. Shekhar and A. Venkatesh.

ponents of a grounded conversational agent.

Our cooperative learning regime can be seen as an interesting alternative to reinforcement learning (RL)—which was first applied to *GuessWhat?!* by Strub et al. (2017)—because it is entirely differentiable and computationally less expensive to train than RL. Little is known on how this learning approach compares to RL not only regarding task success, but also in terms of the quality of the linguistic output, a gap we seek to fill in this paper. In particular, our contributions are:[2]

- The introduction of a single visually-grounded dialogue state encoder jointly trained with the guesser and question generator modules to address a foundational question of how to integrate visual grounding with dialogue system components; this yields up to 9% improvement on task success.
- The effectiveness of cooperative learning, which yields an additional increase of 8.7% accuracy, while being easier to train than RL.
- A first in-depth study to compare cooperative learning to a state-of-the-art RL system. Our study shows that the linguistic skills of the models differ dramatically, despite approaching comparable task success levels. This underlines the importance of linguistic analysis to complement solely numeric evaluation.

2 Related Work

Task-oriented dialogue systems The conventional architecture of task-oriented dialogue systems includes a pipeline of components, and the task of tracking the dialogue state is typically modelled as a partially-observable Markov decision process (Williams et al., 2013; Young et al., 2013; Kim et al., 2014) that operates on a symbolic dialogue state consisting of predefined variables. The use of symbolic representations to characterise the state of the dialogue has some advantages (e.g., ease of interfacing with knowledge bases), but it has also some key disadvantages: the variables to be tracked have to be defined in advance and the system needs to be trained on data annotated with explicit state configurations.

Given these limitations, there has been a shift towards neural end-to-end systems that learn their own representations. Early works focus on non-goal-oriented chatbots (Vinyals and Le, 2015; Sordoni et al., 2015; Serban et al., 2016; Li et al., 2016a,b). Bordes et al. (2017) propose a memory network to adapt an end-to-end system to task-oriented dialogue. Recent works combine conventional symbolic with neural approaches (Williams et al., 2017; Zhao and Eskenazi, 2016; Rastogi et al., 2018), but all focus on language-only dialogue. We propose a visually grounded task-oriented end-to-end dialogue system which, while maintaining the crucial aspect of the interaction of the various modules at play in a conversational agent, grounds them through vision.

Visual dialogue agents In recent years, researchers in computer vision have proposed tasks that combine visual processing with dialogue interaction. Pertinent datasets created by Das et al. (2017a) and de Vries et al. (2017) include *VisDial* and *GuessWhat?!*, respectively, where two participants ask and answer questions about an image. While impressive progress has been made in combining vision and language, current models make simplifications regarding the integration of these two modalities and their exploitation for task-related actions. For example, the models proposed for *VisDial* by Das et al. (2017a) concern an image guessing game where one agent does not see the target image (thus, no multimodal understanding) and is required to 'imagine' it by asking questions. The other agent does see the image, but only responds to questions without the need to perform additional actions.

In *GuessWhat?!*, the Questioner agent sees an image and asks questions to identify a target object in it. The Questioner's role hence involves a complex interaction of vision, language, and guessing actions. Most research to date has investigated approaches consisting of different models trained independently (de Vries et al., 2017; Strub et al., 2017; Zhu et al., 2017; Lee et al., 2017; Shekhar et al., 2018; Zhang et al., 2018). We propose the first multimodal dialogue agent for the *GuessWhat?!* task where all components of the Questioner agent are integrated into a joint architecture that has at its core a *visually-grounded dialogue state encoder* (cf. Figure 1).

Reinforcement learning for visual dialogue agents was introduced by Das et al. (2017b) for *VisDial* and by Strub et al. (2017) for *GuessWhat?!*. Our joint architecture allows us to explore a simpler

[2] Code and supplementary material are available at `https://vista-unitn-uva.github.io`.

solution based on cooperative learning between the agent's internal modules (see Section 5 for details).

3 Task and Data

The *GuessWhat?!* game (de Vries et al., 2017) is a simplified instance of a referential communication task where two players collaborate to identify a referent—a setting used extensively in human-human collaborative dialogue (Clark and Wilkes-Gibbs, 1986; Yule, 1997; Zarrieß et al., 2016).

The *GuessWhat?!* dataset[3] was collected via Amazon Mechanical Turk by de Vries et al. (2017). The task involves two human participants who see a real-world image, taken from the MS-COCO dataset (Lin et al., 2014). One of the participants (the Oracle) is assigned a target object in the image and the other participant (the Questioner) has to guess it by asking Yes/No questions to the Oracle. There are no time constraints to play the game. Once the Questioner is ready to make a guess, the list of candidate objects is provided and the game is considered successful if the Questioner picks the target object. The dataset consists of around 155k English dialogues about approximately 66k different images. Dialogues contain on average 5.2 questions-answer pairs.

4 Models

We focus on developing an agent who plays the role of the Questioner in *GuessWhat?!*.

4.1 Baseline model

As a baseline model (BL), we consider our own implementation of the best performing system put forward by de Vries et al. (2017). It consists of two independent models: a Question Generator (QGen) and a Guesser. For the sake of simplicity, QGen asks a fixed number of questions before the Guesser predicts the target object.

QGen is implemented as an Recurrent Neural Network (RNN) with a transition function handled with Long-Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), on which a probabilistic sequence model is built with a Softmax classifier. At each time step in the dialogue, the model receives as input the raw image and the dialogue history and generates the next question one word at a time. The image is encoded by extracting its VGG-16 features (Simonyan and Zisserman, 2014). In our new joint architecture (described below in Section 4.2), we use ResNet152 (He et al., 2016) features instead of VGG, because they tend to yield better performance in image classification and are more efficient to compute. For the baseline model it turns out that the original VGG-16 features lead to better performance (41.8% accuracy for VGG-16 vs. 37.3% with ResNet152 features). While we use ResNet152 features in our models, we keep the original VGG-16 feature configuration as de Vries et al. (2017), which constitutes a stronger baseline.

The Guesser model exploits the annotations in the MS-COCO dataset (Lin et al., 2014) to represent candidate objects by their object category and their spatial coordinates. This yields better performance than using raw image features in this case, as reported by de Vries et al. (2017). The objects' categories and coordinates are passed through a Multi-Layer Perceptron (MLP) to get an embedding for each object. The Guesser also takes as input the dialogue history processed by its own dedicated LSTM. A dot product between the hidden state of the LSTM and each of the object embeddings returns a score for each candidate object.

The model playing the role of the Oracle is informed about the target object o_{target}. Like the Guesser, the Oracle does not have access to the raw image features. It receives as input embeddings of the target object's category, its spatial coordinates, and the current question asked by the Questioner, encoded by a dedicated LSTM. These three embeddings are concatenated and fed to an MLP that gives an answer (Yes or No).

4.2 Visually-grounded dialogue state encoder

In line with the baseline model, our Questioner agent includes two sub-modules, a QGen and a Guesser. As in the baseline, the Guesser guesses after a fixed number of questions, which is a parameter tuned on the validation set. Our agent architecture differs from the baseline model by de Vries et al.: Rather than operating independently, the language generation and guessing modules are connected through a common *grounded dialogue state encoder* (GDSE) which combines linguistic and visual information as a prior for the two modules. Given this representation, we will refer to our Questioner agent as GDSE.

As illustrated in Figure 1, the encoder receives as input representations of the visual and linguis-

[3] https://guesswhat.ai/download

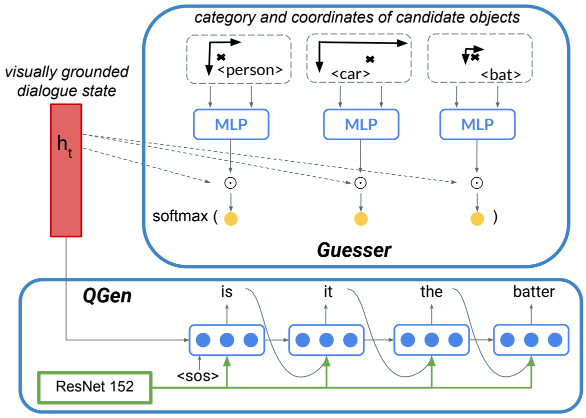


Figure 2: Question Generation and Guesser modules.

tic context. The visual representation consists of the second to last layer of ResNet152 trained on ImageNet. The linguistic representation is obtained by an LSTM (LSTM$_e$) which processes each new question-answer pair in the dialogue. At each question-answer QA_t, the last hidden state of LSTM$_e$ is concatenated with the image features I, passed through a linear layer and a *tanh* activation to result in the final layer h_t:

$$h_t = \tanh\left(W \cdot [\text{LSTM}_e(qa_{1:t-1});\ I]\right) \qquad (1)$$

where $[\cdot;\cdot]$ represents concatenation, $I \in \mathbb{R}^{2048\times1}$, LSTM$_e$ $\in \mathbb{R}^{1024\times1}$ and $W \in \mathbb{R}^{512\times3072}$ (identical to prior work except for tuning the ResNet-specific parameters). We refer to this final layer as the *dialogue state*, which is given as input to both QGen and Guesser.

As illustrated in Figure 2, our QGen and Guesser modules are like the corresponding modules by de Vries et al. (2017), except for the crucial fact that they receive as input the same grounded dialogue state representation. QGen employs an LSTM (LSTM$_q$) to generate the token sequence for each question conditioned on h_t, which is used to initialise the hidden state of LSTM$_q$. As input at every time step, QGen receives a dense embedding of the previously generated token w_{i-1} and the image features I:

$$p(w_i) = p(w_i | w_1, \ldots, w_{i-1}, h_t, I) \qquad (2)$$

We optimise QGen by minimising the Negative Log Likelihood (NLL) of the human dialogues and use the Adam optimiser (Kingma and Ba, 2015):

$$\mathcal{L}_Q = \sum_i -\log p(w_i) \qquad (3)$$

Thus, in our architecture the LSTM$_q$ of QGen in combination with the LSTM$_e$ of the Encoder form a sequence-to-sequence model (Sutskever et al., 2014), conditioned on the visual and linguistic context — in contrast to the baseline model, where question generation is performed by a single LSTM on its own.

The Guesser consists of an MLP which is evaluated for each candidate object in the image. It takes the dense embedding of the category and the spatial information of the object to establish a representation $r_j \in \mathbb{R}^{512\times1}$ for each object. A score is calculated for each object by performing the dot product between the dialogue state h_t and the object representation. Finally, a softmax over the scores results in a probability distribution over the candidate objects:

$$p(o_j) = \frac{e^{h_t^T \cdot r_j}}{\sum_j e^{h_t^T \cdot r_j}} \qquad (4)$$

We pick the object with the highest probability and the game is successful if $o_{guess} = o_{target}$, where $o_{guess} = \arg\max_j p(o_j)$. As with QGen, we optimise the Guesser by minimising the NLL and again make use of Adam:

$$\mathcal{L}_G = -\log p(o_{target}) \qquad (5)$$

The resulting architecture is fully differentiable. In addition, the GDSE agent faces a multi-task optimisation problem: While the QGen optimises $\mathcal{L}_Q$ and the Guesser optimises $\mathcal{L}_G$, the parameters of the Encoder (W, LSTM$_e$) are optimised via both $\mathcal{L}_Q$ and $\mathcal{L}_G$. Hence, both tasks faced by the Questioner agent contribute to the optimisation of the dialogue state h_t, and thus to a more effective encoding of the input context.

5 Learning Approach

We first introduce the supervised learning approach used to train both BL and GDSE, then our cooperative learning regime, and finally the reinforcement learning approach we compare to.

5.1 Supervised learning

In the baseline model, the QGen and the Guesser modules are trained autonomously with supervised learning (SL): QGen is trained to replicate human questions and, independently, the Guesser is trained to predict the target object. Our new architecture with a common dialogue state encoder allows us to formulate these two tasks as a multi-task problem, with two different losses (Eq. 3 and

5 in Section 4.2). These two tasks are not equally difficult: While the Guesser has to learn the probability distribution of the set of possible objects in the image, QGen needs to fit the distribution of natural language words. Thus, QGen has a harder task to optimize and requires more parameters and training iterations. We address this issue by making the learning schedule task-dependent. We call this setup *modulo-n* training, where *n* indicates after how many epochs of QGen training the Guesser is updated together with QGen.

Using the validation set, we experimented with *n* from 5 to 15 and found that updating the Guesser every 7 epochs worked best. With this optimal configuration, we then train GDSE for 100 epochs (batch size of 1024, Adam, learning rate of 0.0001) and select the Questioner module best performing on the validation set (henceforth, GDSE-SL or simply SL).

5.2 Cooperative learning

Once the model has been trained with SL, new training data can be generated by letting the agent play new games. Given an image from the training set used in the SL phase, we generate a new training instance by randomly sampling a target object from all objects in the image. We then let our Questioner agent and the Oracle play the game with that object as target, and further train the common encoder using the generated dialogues by backpropagating the error with gradient descent through the Guesser. After training the Guesser and the encoder with generated dialogues, QGen needs to 'readapt' to the newly arranged encoder parameters. To achieve this, we re-train QGen on the human data with SL, but using the new encoder states. Also here, the error is backpropagated with gradient descent through the common encoder.

Regarding *modulo-n*, in this case QGen is updated at every n^{th} epoch, while the Guesser is updated at all other epochs; we experimented with *n* from 3-7 and set it to the optimal value of 5. The GDSE previously trained with SL is further trained with this cooperative learning regime for 100 epochs (batch size of 256, Adam, learning rate of 0.0001), and we select the Questioner module performing best on the validation set (henceforth, GDSE-CL or simply CL).

5.3 Reinforcement learning

Strub et al. (2017) proposed the first extension of BL (de Vries et al., 2017) with deep reinforcement learning (RL). They present an architecture for end-to-end training using an RL policy. First, the Oracle, Guesser, and QGen models are trained independently using supervised learning. Then, QGen is further trained using a policy gradient.

We use the publicly available code and pretrained model based on Sampling (Strub et al., 2017), which resulted in the closest performance to what was reported by the authors.[4] This is the RL model we use throughout the rest of the paper.

5.4 Experimental details

We use the same train (70%), validation (15%), and test (15%) splits as de Vries et al. (2017). The test set contains new images not seen during training. We use two experimental setups for the number of questions to be asked by the question generator, motivated by prior work: 5 questions (5Q) following de Vries et al. (2017), and 8 questions (8Q) as in Strub et al. (2017). As noted in Section 3, on average, there are 5.2 questions per dialogue in the *GuessWhat?!* data set.

For evaluation, we report task success in terms of accuracy (Strub et al., 2017). To neutralize the effect of random sampling in training CL, we trained the model 3 times. RL is tested 3 times with sampling. We report means and standard deviation (for some tables these are provided in the supplementary material; see footnote 2).

6 Results

Table 1 reports the results for all models. There are several take-aways.

Grounded joint architecture First of all, our visually-grounded dialogue state encoder is effective. GDSE-SL outperforms the baseline by de Vries et al. (2017) significantly in both setups (absolute accuracy improvements of 6.6% and 9%). To evaluate the impact of the multi-task learning aspect, we did an ablation study and used the encoder-decoder architecture to train the QGen and Guesser modules independently. With such a decoupled training we obtain lower results:

[4] Their result of 53.3% accuracy published in Strub et al. (2017) is obsolete, as stated on their GitHub page (`https://github.com/GuessWhatGame/guesswhat`) where they report 56.5% for sampling and 58.4% for greedy search. By running their code, we could only replicate their results with sampling, obtaining 56%, while greedy and beam search resulted in similar or worse performance. Our analysis showed that greedy and beam search have the additional disadvantage of learning a smaller vocabulary.

Model	5Q	8Q
Baseline	41.2	40.7
GDSE-SL	47.8	49.7
GDSE-CL	53.7 (±.83)	58.4 (±.12)
RL	56.2 (±.24)	56.3 (±.05)

Table 1: Test set accuracy for each model (for setups with 5 and 8 questions). GDSE-SL is our grounded supervised learning system, GDSE-CL the cooperative learning setup, and RL the results we obtain with the reinforcement learning system by Strub et al. (2017).

44% and 43.7% accuracy for 5Q and 8Q, respectively. Hence, the multi-task component brings an increase of up to 6% over the baseline.[5]

Cooperative learning and RL The introduction of the cooperative learning approach results in a clear improvement over GDSE-SL: +8.7% (8Q: from 49.7 to 58.4) and +5.9% (with 5Q). Despite its simplicity, our GDSE-CL model achieves a task success rate which is comparable to RL: In the 8Q setup, GDSE-CL reaches an average accuracy of 58.4 versus 56.3 for RL, giving CL a slight edge in this setup (+2.1%), while in the 5Q setup RL is slightly better (+2.5%). Overall, the accuracy of the CL and RL models is close. The interesting question is how the linguistic skills and strategy of these two models differ, to which we turn in the next section.

We compared to Strub et al. (2017), but RL has also been put forward by Zhang et al. (2018), who report 60.7% accuracy (5Q). This result is close to our highest GDSE-CL result (60.8 ±0.51, when optimized for 10Q).[6] Their RL system integrates several partial reward functions to increase coherence, which is an interesting aspect. Yet their code is not publicly available. We leave the comparison to Zhang et al. (2018) and adding RL to GDSE to future work.

7 Analysis

In this section, we present a range of analyses that aim to shed light on the performance of the models. They are carried out on the test set data using the 8Q setting, which yields better results than the 5Q setting for the GDSE models and RL. Given that there is only a small difference in accuracy for the baseline with 5Q and 8Q, for comparability we analyse dialogues with 8Q also for BL.

7.1 Quantitative analysis of linguistic output

We analyse the language produced by the Questioner agent with respect to three factors: (1) lexical diversity, measured as type/token ratio over all games, (2) question diversity, measured as the percentage of unique questions over all games, and (3) the number of games with questions repeated verbatim. We compute these factors on the test set for the models and for the human data (H).

As shown in Table 2, the linguistic output of SL & CL is closer to the language used by humans: Our agent is able to produce a much richer and less repetitive output than both BL and RL. In particular, it learns to use a more diverse vocabulary, generates more unique questions, and repeats questions within the same dialogue at a much lower rate than the baseline and RL: 93.5% of the games played by BL contain at least one verbatim question repetition, for RL this happens in 96.47% of the cases, whereas for SL and CL this is for only 55.8% and 52.19% of the games, respectively.

	Lexical diversity	Question diversity	% Games with repeated Q's
BL	0.030	1.60	93.50
SL	0.101	13.61	55.80
CL	0.115 (±.02)	14.15 (±3.0)	52.19 (±4.7)
RL	0.073 (±.00)	1.04 (±.03)	96.47 (±.04)
H	0.731	47.89	—

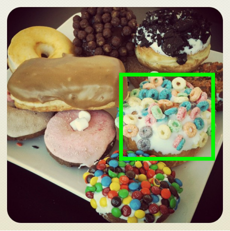
Table 2: Statistics of the linguistic output of all models with the 8Q setting and of humans (H) in all test games.

7.2 Dialogue strategy

To further understand the variety of questions asked by the agents, we classify questions into different types. We distinguish between questions that aim at getting the category of the target object (ENTITY questions, e.g., *'is it a vehicle?'*) and questions about properties of the queried objects (ATTRIBUTE questions, e.g., *'is it square?'* or *'are they standing?'*). Within ATTRIBUTE questions, we make a distinction between color, shape, size, texture, location, and action questions. Within ENTITY questions, we distinguish questions whose focus is an object category or a super-category (see the supplementary material for example questions). The classification is done by manually extracting keywords for each question type from the human di-

[5] While de Vries et al. (2017) originally report an accuracy of 46.8%, this result was later revised to 40.8%, as clarified on their GitHub page. Our own implementation of the baseline system achieves an accuracy of 41.2%.

[6] Since our aim is to compare to the best setup for BL (5Q) and RL (8Q), we do not report our results with 10Q in Table 1.



Humans	[*success*]	**GDSE-CL**	[*success*]	**RL**	[*failure*]
1. does it have cereal on it?	no	1. is it a donut?	yes	1. is it food?	yes
2. does it have pink frosting?	no	2. is it on the left?	no	2. is it a donut?	yes
3. does it have cookies?	no	3. is it on the right?	yes	3. is it in left?	no
4. is it a donut?	yes	4. is it the whole donut?	yes	4. is it in top?	yes
5. does it have maple frosting?	no	5. does it have pink icing?	no	5. is it in top?	yes
6. is there candy?	yes	6. is it touching the donut with the sprinkles?	yes	6. is it in top?	yes
7. is it the doughnut with M&Ms?	no	7. does it have chocolate icing?	no	7. is it in top?	yes
8. is it dark brown?	no	8. is it the third donut?	yes	8. is it top?	yes

Figure 3: Game example where GDSE-CL succeeds and RL fails at guessing the target object (green box).

alogues, and then applying an automatic heuristic that assigns a class to a question given the presence of the relevant keywords.[7] This procedure allows us to classify 91.41% of the questions asked by humans. The coverage is higher for the questions asked by the models: 98.88% (BL), 94.72% (SL), 94.11% (CL) and 99.51 % (RL).[8]

The statistics are shown in Table 3. We use Kullback-Leibler (KL) divergence to measure how the output of each model differs from the human distribution of fine-grained question classes. The baseline's output has the highest degree of divergence: For instance, the BL model does never ask any SHAPE or TEXTURE questions, and hardly any SIZE questions. The output of the RL model also differs substantially from the human dialogues: It asks a very large number of LOCATION questions (74.8% vs. 40% for humans). Our model, in contrast, generates question types that resemble the human distribution more closely.

Question type	BL	SL	CL	RL	H
ENTITY	**49.00**	**48.07**	**46.51**	**23.99**	**38.11**
SUPER-CAT	19.6	12.38	12.58	14.00	14.51
OBJECT	29.4	35.70	33.92	9.99	23.61
ATTRIBUTE	**49.88**	**46.64**	**47.60**	**75.52**	**53.29**
COLOR	2.75	13.00	12.51	0.12	15.50
SHAPE	0.00	0.01	0.02	0.003	0.30
SIZE	0.02	0.33	0.39	0.024	1.38
TEXTURE	0.00	0.13	0.15	0.013	0.89
LOCATION	47.25	37.09	38.54	74.80	40.00
ACTION	1.34	7.97	7.60	0.66	7.59
Not classified	**1.12**	**5.28**	**5.90**	**0.49**	**8.60**
KL (wrt human)	0.953	0.042	0.038	0.396	0.0

Table 3: Percentage of questions per question type in all the test set games played by humans (H) and the models with the 8Q setting, and KL divergence from human distribution of fine-grained question types.

We also analyse the structure of the dialogues in terms of the sequences of question types asked. As expected, both humans and models almost always start with an ENTITY question (around 97% for BL, SL and CL, 98.7% for RL, and 78.48% for humans), in particular a SUPER-CATEGORY (around 70% for BL, SL and CL, 84% for RL, and 52.32% for humans). In some cases, humans start by asking questions directly about an attribute that may easily distinguish an object from others, while this is very uncommon for models. Figure 3 shows an example: The human dialogue begins with an ATTRIBUTE question (*'does it have cereal on it?'*), which in this case is not very effective and leads to a change in strategy at turn 4. The CL model starts by asking an OBJECT question (*'is it a donut?'*) while the RL model begins with a more generic SUPER-CATEGORY question (*'is it food?'*).

We check how the answer to a given question type affects the type of the follow-up question. In principle, we expect to find that question types that are answered positively will be followed by more specific questions. This is indeed what we observe in the human dialogues, as shown in Table 4. For example, when a SUPER-CATEGORY question is answered positively, humans follow up with an OBJECT or ATTRIBUTE question 89.56% of the time. This trend is mirrored by all models.

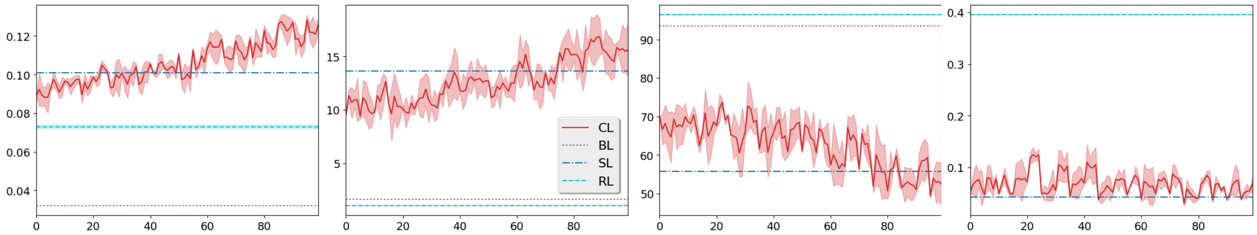
Question type shift	BL	SL	CL	RL	H
SUPER-CAT → OBJ/ATT	89.05	92.61	89.75	95.63	89.56
OBJECT → ATTRIBUTE	67.87	60.92	65.06	99.46	88.70

Table 4: Proportion of question type shift vs. no type shift in consecutive questions $Q_t \longrightarrow Q_{t+1}$ where Q_t has received a Yes answer.

Overall, the models also learn the strategy to move from an OBJECT to an ATTRIBUTE question when an OBJECT question receives a Yes answer. The BL, SL, and CL models do this to a lesser extent than humans, while the RL model systematically transitions to attributes (in 99.46% of cases), using

[7] A question may be tagged with several attribute classes if keywords of different types are present. E.g., *"Is it the white one on the left?"* is classified as both COLOR and LOCATION.

[8] In the supplementary material we provide details on the question classification procedure: the lists of keywords by class, the procedure used to obtain these lists, as well as the pseudo-code of the heuristics used to classify the questions.



(a) Lexical diversity (b) Question diversity (c) % Games w/ repeated Q's (d) KL-distance from human

Figure 4: Evolution of linguistic factors over 100 training epochs for our GDSE-CL model. Note: lexical and question diversity of the human data fall outside the range in (a) / (b). The same is the case with KL for BL in (d).

mostly LOCATION questions as pointed out above. For example (Figure 3), after receiving an affirmative answer to the OBJECT question 'is it a donut?' both CL and RL shift to a LOCATION question. Once location is established, CL moves on to other attributes while RL keeps asking the same LOCATION question, which leads to failure. Further illustrative examples are given in the supplementary material.

7.3 Analysis of the CL learning process

In order to better understand the effect of the cooperative learning regime, we trace the evolution of linguistic factors identified above over the CL epochs. As illustrated in Figure 4 (a) and (b), through the epochs the CL model learns to use a richer vocabulary and more diverse questions, moving away from the levels achieved by BL and RL, overpassing SL and moving toward humans.

The CL model progressively produces fewer repeated questions within a dialogue, improving over SL in the last few epochs, cf. Figure 4 (c). Finally, (d) illustrates the effect of modulo-n training: As the model is trained on generated dialogues, its linguistic output drifts away from the human distribution of question types; every 5^{th} epoch QGen is trained via supervision, which brings the model's behaviour closer back to human linguistic style and helps decrease the drift.

8 Conclusion

We present a new visually-grounded joint Questioner agent for goal-oriented dialogue. First, we show that our architecture archives 6–9% accuracy improvements over the *GuessWhat?!* baseline system (de Vries et al., 2017). This way, we address a foundational limitation of previous approaches that model guessing and questioning separately.

Second, our joint architecture allows us to propose a two-phase cooperative learning approach (CL), which further improves accuracy. It results in our overall best model and reaches state-of-the-art results (cf. Section 6). We compare CL to the system proposed by Strub et al. (2017) which extends the baseline with reinforcement learning (RL). We find that the two approaches (CL and RL) achieve overall relatively similar task success rates. However, evaluating on task success is only one side of the coin. Finally and most importantly, we propose to pursue an in-depth analysis of the quality of the dialogues by visual conversational agents, which is an aspect often neglected in the literature. We analyze the linguistic output of the two models across three factors (lexical diversity, question diversity, and repetitions) and find them to differ substantially. The CL model uses a richer vocabulary and inventory of questions, and produces fewer repeated questions than RL. In contrast, RL highly relies on asking location questions, which might be explained by a higher reliance on spatial and object-type information explicitly given to the Guesser and Oracle models. Limiting rewards to task success or other rewards not connected to the language proficiency does not stimulate the model to learn rich linguistic skills, since a reduced vocabulary and simple linguistic structures may be an efficient strategy to succeed at the game.

Overall, the presence of repeated questions remains an important weakness of all models, resulting in unnatural dialogues. This shows that there is still a considerable gap to human-like conversational agents. Looking beyond task success can provide a good basis for extensions of current architectures, e.g., Shekhar et al. (2018) add a decision-making component that decides when to stop asking questions which results in less repetitive and more human-like dialogues. Our joint architecture could easily be extended with such a component.

Acknowledgements

The work carried out by the Amsterdam team was partially funded by the Netherlands Organisation for Scientific Research (NWO) under VIDI grant nr. 276-89-008, *Asymmetry in Conversation*. We thank the University of Trento for generously funding a research visit by Barbara Plank to CIMeC that led to part of the work presented in this paper. In addition, we kindly acknowledge the support of NVIDIA Corporation with the donation to the University of Trento of the GPUs used in our research.

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal oriented dialog. In *Proceedings of ICLR*.

Herbert H Clark. 1996. *Using Language*. Cambridge University Press.

Herbert H Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017a. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. 2017b. Learning cooperative visual dialog agents with deep reinforcement learning. In *International Conference on Computer Vision (ICCV)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dongho Kim, Catherine Breslin, Pirros Tsiakoulis, M Gašić, Matthew Henderson, and Steve Young. 2014. Inverse reinforcement learning for micro-turn management. In *Fifteenth Annual Conference of the International Speech Communication Association*.

D. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Sang-Woo Lee, Yujung Heo, and Byoung-Tak Zhang. 2017. Answerer in questioner's mind for goal-oriented visual dialogue. In *NIPS Workshop on Visually-Grounded Interaction and Language (ViGIL)*.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8(1):339–359.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-2016*, pages 110–119.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of EMNLP*.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, Dollar, P., and C. L. Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of ECCV (European Conference on Computer Vision)*.

Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of SIGdial*.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Ravi Shekhar, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernández. 2018. Ask no more: Deciding when to guess in referential visual dialogue. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1218–1233.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL-HLT*, pages 196–205.

Robert Stalnaker. 1978. Assertion. In P. Cole, editor, *Pragmatics*, volume 9 of *Syntax and Semantics*. New York Academic Press.

Florian Strub, Harm de Vries, Jeremie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Joint Conference on Artificial Intelligence*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! Visual object discovery through multi-modal dialogue. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jason Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of ACL-2017*. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

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

George Yule. 1997. *Referential communication tasks*. Routledge.

Sina Zarrieß, Julian Hough, Casey Kennington, Ramesh Manuvinakurike, David DeVault, Raquel Fernández, and David Schlangen. 2016. Pentoref: A corpus of spoken references in task-oriented dialogues. In *10th edition of the Language Resources and Evaluation Conference*.

Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, Jianfeng Lu, and Anton van den Hengel. 2018. Goal-oriented visual question generation via intermediate rewards. In *Proceedings of the European Conference of Computer Vision (ECCV)*.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state traching and management using deep reinforcement learning. In *Proceedings of SIGDIAL-2016*.

Yan Zhu, Shaoting Zhang, and Dimitris Metaxas. 2017. Interactive reinforcement learning for object grounding via self-talking. In *NIPS Workshop on Visually-Grounded Interaction and Language (ViGIL)*.