Adding object detection skills to visual dialogue agents

Gabriele Bani, Davide Belli, Gautier Dagan, Alexander Geenen, Andrii Skliar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, and Raquel Fernández

Institute for Logic, Language and Computation
University of Amsterdam
P.O. Box 94242, 1090 GE Amsterdam, The Netherlands
Corresponding author: elia.bruni@gmail.com

Abstract. Our goal is to equip a dialogue agent that asks questions about a visual scene with object detection skills. We take the first steps in this direction within the GuessWhat?! game. We use Mask R-CNN object features as a replacement for ground-truth annotations in the Guesser module, achieving an accuracy of 57.92%. This proves that our system is a viable alternative to the original Guesser, which achieves an accuracy of 62.77% using ground-truth annotations, and thus should be considered an upper bound for our automated system. Crucially, we show that our system exploits the Mask R-CNN object features, in contrast to the original Guesser augmented with global, VGG features. Furthermore, by automating the object detection in GuessWhat?!, we open up a spectrum of opportunities, such as playing the game with new, non-annotated images and using the more granular visual features to condition the other modules of the game architecture.

Keywords: visual dialogue, object detection

1 Introduction

In recent years, there has been considerable progress in combining natural language processing and computer vision, with applications that span image captioning [11], visual question answering [1] and, more recently, visually grounded dialogue [2]. Despite such advancements, current models achieve a rather fragile alignment between vision and language—as shown, for example, by Shekhar et al. [13]—and are thus far from being able to effectively exploit the two modalities in tandem.

In this work, we make progress in this direction by equipping a visual dialogue agent that asks natural language questions about an image with automatic object localisation skills. In particular, we focus on the GuessWhat?! game [3], where the goal is to identify a target object in an image by asking a series of yes/no questions to an Oracle agent who is aware of the target.

The model we propose uses as backbone the original architecture by de Vries et al. [3], but with the crucial difference that the objects in the image are automatically localised. As object detection system, we use Mask R-CNN (MRCNN),

a detection algorithm which has been shown to obtain state-of-the-art performance on standard image detection tasks [8]. We show that an agent equipped with automatic object detection skills performs almost at the level of an agent that has direct access to ground-truth object locations and categories.

2 Guessing in the GuessWhat?! game

The architectures proposed so far to model agents able to play the GuessWhat?! game [3] split the Questioner agent into two sub-modules: a Question Generator, which poses new questions based on the visual input and the dialogue history (i.e., previous questions and answers), and a Guesser, whose task is to pick an object from a list of candidates once the dialogue is terminated. In this work, we focus on the Guesser component.

In all current GuessWhat?! models, the Guesser relies on ground-truth annotations. In particular, when the guessing phase begins, the Guesser receives a variable number of candidate objects, described by their coordinates and object categories. These annotations (as well as the images of the game) are taken from the MS COCO Dataset [11] (we refer to this as *ground-truth model*). In the present work, we make use of recent advances in object detection to propose a new model for the Guesser that does not require any ground-truth annotations.

Our starting point is the general architecture introduced by de Vries et al. [3]. An LSTM processes the dialogue d into a fixed sized, continuous vector. The objects in the image are represented by an 8-dimensional bounding box feature[1] [9,15] and their categories. The bounding box feature b and a dense embedding of the object category c are concatenated and processed by an MLP with shared weights for all objects producing a representation for each object. A dot product between the dialogue and object representation results in a score for each object o_i. All scores are normalised with a softmax function, as shown in equation (1). The model is trained with Cross Entropy Loss.

$$p(o_i) = \text{softmax}\Big(\text{MLP}([b_i, c_i]) \cdot \text{LSTM}(d)\Big) \tag{1}$$

A key limitation of this approach is that it cannot generalise beyond the cases where ground-truth annotations are provided. Furthermore, it exploits a limited part of the information, namely the bounding box and the object category but not the image itself. Although [3] experiment with visual features extracted from the whole image by adding the fully connected layer of VGG16 [14] to the dialogue representation (we refer to this as *global features*), this did not improve results. In our model, we obtain a representation for each object by using different visual features, thus replacing the annotated bounding boxes and object categories. Our aim is to investigate how far we can get in terms of task success with a fully automatic approach, which would make the Guesser able to generalise to new images.

[1] This representation consists of the normalised width and height of the bounding box, as well as the lower left and the upper right bounding box coordinates.

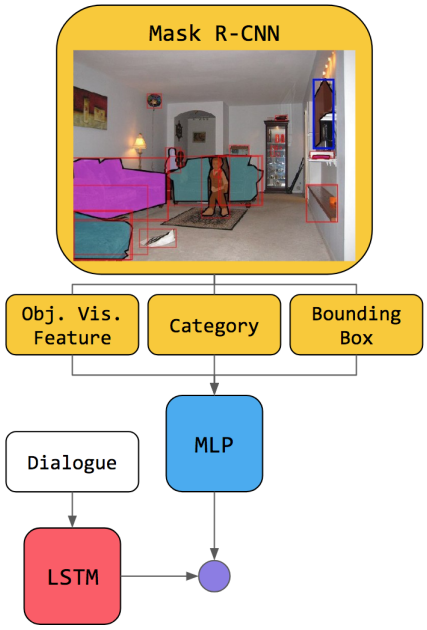

Fig. 1: Proposed Guesser Architecture with features from MRCNN.

3 Automatic candidate localisation

When natural language refers to the outside visual world, it does not only address it holistically, but it also focuses on specific objects and locations. Hence, if we want to ground language in vision, we should provide it with visual features that are at that level of granularity. This is particularly the case for the GuessWhat?! game, where the task is that of localising an object of interest. Previous work on GuessWhat?! relied always on global features instead, such as VGG and Resnet, which provide only a poor representation of the individual objects in the scene and we believe this is the reason why they do not have significant impact on the task [3].

On the other hand, recent advances in object detection allow for precise localisation of multiple objects in a visual scene. We leverage this progress by making use of MRCNN [8], the current state of the art in object detection on MS COCO [11]. We use the Detectron implementation [7] with ResNet-101-FPN [10] as a backbone.

MRCNN performs both object detection and semantic segmentation. It outputs (i) a bounding box for each detected object, (ii) a class probability for each bounding box, and (iii) a class probability for each pixel in the image. Based on the Fast/Faster R-CNN architecture [6,12], MRCNN has a Convolution Neural Network (CNN) at its core, processing the entire image. Given region proposals, the feature maps of the CNN for a region are processed by a pooling and multiple

fully connected layers, eventually branching into the three outputs mentioned above. We make use of the last hidden layer as 'visual features' (we refer to this as *object features*).

We explore different types of visual information to obtain object representations for the Guesser model with four different settings:

1. Ground-truth model with manual bounding boxes and object category annotations.
2. Ground-truth model as in 1, but augmented with global features from the first fully connected layer of VGG16 (FC6).
3. The predictions of MRCNN, replacing the ground-truth bounding box and object category.
4. The predictions of MRCNN, replacing the ground-truth bounding box and object category and adding the last hidden layer as visual features representing the object.

Figure 1 sketches the architecture of the last setting, i.e., using the information obtained through MRCNN to represent the object.

4 Training and evaluation

Performance in object detection is measured by evaluating the Intersection over Union (IoU), which compares two bounding boxes A and B, as shown in equation (2). A detection is considered successful if the IoU between the ground-truth and the predicted bounding box is at least 0.5.

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

4.1 Category filtering

Since we aim at replacing the manual annotations used in the ground-truth model, we need to label the MRCNN predictions in order to train our model. From the set of bounding boxes and categories, we need to select the prediction which matches the ground-truth annotation best. The simplest option is to select the proposal with the highest IoU ≥ 0.5 as the target. We also experiment with a refined condition where we first filter all proposals by category, i.e., we only consider those proposals where the category prediction matches the ground-truth category. We then select the proposal with the highest IoU as the target.

4.2 Single- vs. multi-target

The procedure described above makes available one single bounding box as the target, namely the one with the highest IoU. However, since an IoU ≥ 0.5 is considered a correct detection in the object detection literature, we can relax this constraint and consider any prediction with IoU ≥ 0.5 as correct, even if it is not the bounding box with the highest IoU. We evaluate the model trained with the single-target setup, but with the multi-target evaluation procedure.

Table 1: Guessing accuracy with different settings introduced in Section 3. Settings 1 and 2 correspond to the ground-truth model without and with added global features. Settings 3 and 4 show results with single-target training using MRCNN bounding box and category predictions. For both settings we report the influence of category filtering. Setting 4 additionally uses object features

Setting	BBox & Cat	Filter	Features	Accuracy
1	Ground-Truth	-	-	62.19%
2	Ground-Truth	-	Global	62.77%
3	MRCNN	-	-	40.10%
3	MRCNN	Category	-	42.45%
4	MRCNN	-	Object	50.47%
4	MRCNN	Category	Object	53.40%

5 Results and discussion

5.1 Accuracy results

We train and evaluate all the models as described in the previous section. Table 1 reports accuracy results on task success when a single target with the highest IoU is considered. The first two rows report results for the two ground-truth models, while the rest of the table shows results for the conditions where the MRCNN object detection pipeline is used.

The first ground-truth model uses the manual bounding box and object category annotations to make a prediction (setting 1) and confirms results originally reported in [3]. Further, we also report results on a ground-truth model with global features added (setting 2). In contrast to [3], adding global features to the model does not decrease performance. This might be due to the lower level features we use in our experiments (first fully connected layer of VGG16, FC6), while [3] use higher level features (last fully connected of VGG16, FC8). However, the usage of these features also does not improve results. We therefore conclude that global features are not helpful enough.

Coming to our proposed model, we first report results on setting 3, which uses the MRCNN predictions of the bounding box and the object category, but not the object features. For this experiment, the performance of setting 1 has to be considered as its exact upper bound, because MRCNN predictions are replacing the ground-truth annotations and there are no added object features. The gap between the models is significant, with a 20% point difference. However, note that also the task becomes inherently harder as the average number of candidate objects jumps from 8.1 objects in the ground truth setting to 15.6 objects in the automated detection setting. Applying the category filter improved the results 2% points.

Next, we look at setting 4, which includes the object features. Remarkably, this leads to a big performance gain. The models obtain about 10% points more on the task than their counterparts without object features. This result is clearly

Table 2: Guessing accuracy for single- and multi-target evaluation. Results achieved with the best performing model from Table 1

Single-target	53.40%	Multi-target	57.92%

showing that a more fine-grained visual representation can be exploited by the Guesser in combination with the linguistic input to make its predictions. Again applying the category filter improves results about 3% points.

Table 2 shows accuracy results with the single- and multi-target setup for the best performing model in the single-target setup (MRCNN predictions with object features and category filtering). While the results obtained evaluating on a single-target are promising, when we relax the single-target constraint and allow multi-target evaluation we close the gap on the ground-truth even further, reaching 57.92% accuracy.

5.2 MS COCO and GuessWhat?! splits

For all conducted experiments we use MRCNN, which uses the MS COCO train and validation splits. However, these splits do not align with the GuessWhat?! splits. In the GuessWhat?! dataset the training set also contains MS COCO validation images, and vice versa. This could possibly compromise our results, since MRCNN is trained on the original MS COCO training split. Therefore, we reduced the GuessWhat?! splits, keeping only those games which align with the MS COCO split. This results in about 67% of the training data and 32% of validation and test data. The Guesser model with ground-truth information achieves 61.0% on the test set, whereas our proposed model with MRCNN features, category filter, and object features achieves 52.25%. When evaluated on the multi-target setting, the model achieves 56.37%. Since these results do not deviate significantly from using the full GuessWhat?! set, we conclude that performance improvements are not gained form using MS COCO train images in the GuessWhat?! validation and test splits.

5.3 Visualising predictions during the dialogue

To gain a qualitative understanding of how the Guesser model behaves, we visualise the predictions incrementally made by the model as the dialogue proceeds.

Figure 2 shows the visualisations for a sample game obtained with the best performing model (setting 4 with category filtering). In general, the Guesser is very accurate in making sense of the dialogue and changing its prediction accordingly, even for complex sentences subordinated to previous questions. The Guesser is especially good at understanding class and position of objects, thanks to the information included in the object features.

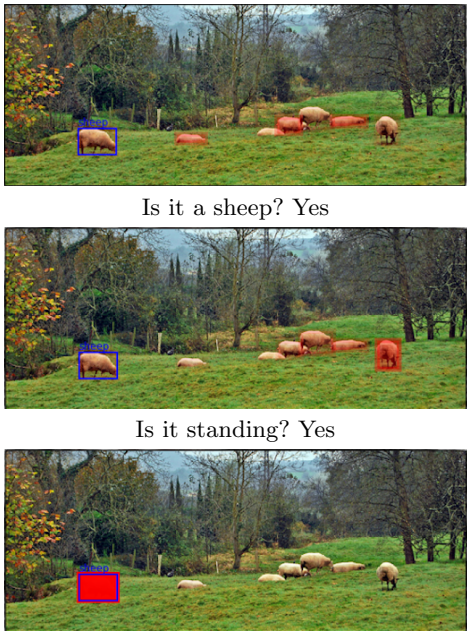

Fig. 2: Change in object probability over the course of the dialogue. The blue bounding box shows the target object. We colour every bounding box with intensity proportional to the prediction probability of it being the target.

6 Conclusion and future work

We have shown that using features from an object detection network is useful for the task of guessing a referent in the GuessWhat?! game. Our results indicate that the Guesser agent exploits the provided MRCNN visual features, in contrast to the baseline model where adding VGG features does not lead to improvements. This suggests that the MRCNN visual features are more informative than features used for object classification such as VGG. This might be due to the fact that VGG is trained on Imagenet [5], where usually a single object has to be classified. While the task MRCNN is trained on requires the localisation and classification of multiple objects. This setting is much closer to the task faced by the Guesser in the GuessWhat?! game.

With the proposed model, the agent can be scaled to any visual input as no ground-truth annotation of the objects is required. Although we achieve slightly inferior task performance than with annotations, the results are promising, especially considering that the automatically obtained visual information is noisy.

Furthermore, these results also open many new opportunities. Since the object detection is autonomous and end-to-end from the image, the features could also be utilised during the dialogue. For example, the Question Generator can

be conditioned on them and maintain a belief state over the candidate objects. This provides more fine-grained grounding of the agent as well as the explicit opportunity to ask more discriminative questions. In future work, we also plan to test the use of MRCNN features for the Oracle agent (which in the model by [3] uses ground-truth annotations) and systematically compare the results to the visual features leveraged by [4].

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
2. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual dialog. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2 (2017)
3. De Vries, H., Strub, F., Chandar, S., Pietquin, O., Larochelle, H., Courville, A.C.: Guesswhat?! visual object discovery through multi-modal dialogue. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
4. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. In: Advances in Neural Information Processing Systems. pp. 6594–6604 (2017)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
6. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
7. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. `https://github.com/facebookresearch/detectron` (2018)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
9. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4555–4564 (2016)
10. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
11. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European conference on computer vision. pp. 740–755 (2014)
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
13. Shekhar, R., Pezzelle, S., Klimovich, Y., Herbelot, A., Nabi, M., Sangineto, E., Bernardi, R.: Foil it! find one mismatch between image and language caption. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 255–265 (2017)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

15. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: Proceedings of the European conference on computer vision. pp. 69–85 (2016)