Visual TTR
Modelling Visual Question Answering in Type Theory with Records

Ronja Utescher
Bielefeld University
r.utescher@uni-bielefeld.de

April 1, 2019

Abstract

In this paper, I will describe a system that was developed for the task of Visual Question Answering. The system uses the rich type universe of Type Theory with Records (TTR) to model utterances about the image, the image itself, and classifications made relating the outcomes of these two tasks. At its most basic, the decision of whether any given predicate can be assigned to an object in the image is delegated to a CNN. Consequently, images can be taken as evidence for propositional judgments. The end result is a model whose application of perceptual classifiers to a given image is guided by the accompanying utterance.

1 Introduction

Visual question answering is a recent popular task in the field of computer vision. However, the extent to which formal linguistics is needed to solve the task has been a point of contention. This paper details an approach that utilizes both a rule-based approach to parsing utterances about an image and a deep neural model to supply perceptual meaning. TTR (Cooper and Ginzburg, 2015) offers a powerful semantic framework for modelling natural language. TTR has been used to model more coarse-grained linguistic phenomena, many of them related to dialogue. However, this paper is concerned with relatively basic phenomena. The challenge here is to model a multimodal world, namely a visual and linguistic one.
This project builds on a previous VQA model using TTR which is detailed in Matsson (2018)[1]. Both projects utilize pyTTR (Cooper, 2017), a python implementation of TTR. This previous implementation features a pipeline that includes object recognition in the form of You Only Look Once (YOLO, Redmon et al. (2015)), representation of the image and question in TTR and, subsequently, evaluation of the utterance with respect to the image. The TTR representation of the image consists of a record type that contains an individual variable and bounding box for every detected object, as well as the predicates that apply to them. Furthermore, it uses the predicate *loc* to link individual variables to their bounding boxes. This predicate simply signifies that the individual with this name is *located* at this position in the image. I refine the TTR modelling of the image and object classification and replace YOLO with a set of binary word classifiers. In Visual TTR, predicates do not need to be added explicitly to the TTR representation of the image. Instead, links between the image and the question are made where appropriate. For example, if a question contains a reference to a dog, the system will try to find suitable objects by running the dog classifier on every annotated entity in the image. If the classifier returns a sufficiently high score for any of the objects, these objects are considered instances of the *dog* predicate (type). These technical changes enable a change to the order in which the model performs its sub-tasks. Where the original system runs an object recognition algorithm on the image and translates the result to a TTR representation of the image, the question is now parsed first, and guides the perceptual classification part of the architecture.

[1] see `https://github.com/arildm/imagettr`

In section 3, I make recommendations for appropriate training data and classifier design. Based on the classifier score, likely candidate regions will be considered instances (or witnesses) of the predicate type. In the case of polar questions, this classified record of the image is a witness of the question type iff the answer to the question is *yes*.

Matsson (2018)	Visual TTR
object recognition	bounding box annotations
$\downarrow bounding\ boxes, predicates, entities$	$\downarrow bounding\ boxes, entities$
image type	image record
question parsing	question parsing
-	object classification
type check	type check
$\downarrow answer$	$\downarrow answer$

Figure 1: Comparison of ImageTTR and Visual TTR Pipelines

2 A Visual Universe of Types

In order to implement the visual classification in TTR, all information necessary for classification should be contained in the representation of the image. While it would be possible to include the entire image matrix, this model uses the path to the image for legibility reasons. This section provides an overview of the types (and types of types) that are used in the model. Basic Types are basic in the sense that they do not depend on other types and should be thought of as corresponding to basic ontological categories (Cooper and Ginzburg, 2015).

2.1 Basic Types

Image(path) The source of the image data.

Int Integers, used to describe the coordinates of the bounding boxes.

Ind Variables of type *Ind* are Montagovian individuals. In the record for a given image, every object is assigned an identifier or name. In the case of the examples in this paper, this name uses the object ids annotated in the corpus (see section 3.1).

2.2 The Image in TTR

2.2.1 Segment & Region

The model utilizes segmented images, as are commonly provided with state-of-the-art image corpora like MS COCO (Lin et al., 2014) or Visual Genome (Krishna et al., 2016).

The segment contains the (x,y) coordinates of the bottom-left corner as well as the width and height of the bounding box, as well as the path to the image. Note that this constitutes all the visual information about the relevant part of the image.

The region contains a segment and its name, a variable of type *Ind*. The two fields in the *Region* type represent the segment *seg* and the name z of the object in question.

$$\begin{bmatrix} seg & : & \begin{bmatrix} x & : & Int \\ y & : & Int \\ w & : & Int \\ h & : & Int \\ path & : & Image \end{bmatrix} \\ z & : & Ind \end{bmatrix} \tag{1}$$

2.2.2 Scene

The Scene type consists of at least one *Ind* type and a corresponding *Region* type object. The Scene is the TTR representation of the entire image and contains the information of every object in the image. Note that the names of each object appear twice in this format. This has two purposes. One, the image itself also contains the individuals; two, the individual is now clearly linked to its segment.
When processing an image, a record/an instance of the *Scene* type is produced. In an image with only two objects, this could look like (2).

$$\left[\begin{array}{rcl} obj_0 & = & \left[\begin{array}{rcl} seg & = & \left[\begin{array}{rcl} x & = & 349 \\ y & = & 138 \\ w & = & 71 \\ h & = & 90 \\ path & = & image.jpg \end{array}\right] \\ z & = & a_{1032844} \end{array}\right] \\ obj_1 & = & \left[\begin{array}{rcl} seg & = & \left[\begin{array}{rcl} x & = & 3 \\ y & = & 146 \\ w & = & 204 \\ h & = & 90 \\ path & = & image.jpg \end{array}\right] \\ z & = & a_{1032847} \end{array}\right] \\ z_0 & = & a_{1032844} \\ z_1 & = & a_{1032847} \end{array}\right] \tag{2}$$

3 Visual Grounding

3.1 Training Data

Visual Genome (Krishna et al., 2016) is a densely annotated dataset of 108k images. The dataset contains several kinds of human-generated annotations such as region descriptions and question/answer pairs. However, the model described in this paper works solely with object annotations. The object annotations consist of a name and bounding box. The object names are extracted from region descriptions by crowd-workers. The format I used for preprocessing these annotations can be found in the repository released alongside Schlangen (2019).[2]

3.2 Object Classification

In contrast to Matsson (2018), the model described in this paper uses object classifiers. Conceptually, these represent the system's understanding of the perceptual meaning of object names. This means that a separate classifier must be trained for every word in the system's vocabulary. This particular implementation uses word classifiers with a architecture much like that described in (Schlangen et al., 2016)[3]. These classifiers are binary logistic regression classifiers based on vgg19 (Simonyan and Zisserman, 2014) features. The classifiers share a common base model that outputs the visual features, while the final layer is different for every word(-model).
Opting for these classifiers over the YOLO-model leads to more control over the vocabulary. YOLO uses the PASCAL VOC (Everingham et al., 2015) test set of twenty object categories. The perceptual classifiers also do not have a structural bias against infrequent categories in the training data.

[2] However, I trained classifiers on a subset of the roughly 3.8 million object annotations.

[3] Although the architectures are similar, the data and application of the models turn out quite differently. Compared to reference resolution task that the word classifiers from Schlangen et al. (2016) were used for, object naming is a comparatively simpler task.

4 Classification in Visual TTR

4.1 Perceptual Segments

For every predicate, there is a corresponding Basic Type that maps from visual data (in this case, a *seg* record) to a basic perceptual type. It is here that pyTTR invokes the classifier. For example, the conditions for being a *DogSeg* are (1) be a *seg*, (2) get a higher-than-threshold score from the dog classifier (see (3)). This is not yet applicable to the question - it represents the perceptually basic type of *looking like a dog*.

$$\left[\begin{array}{lll} seg & : & \left[\begin{array}{rcl} x & : & Int \\ y & : & Int \\ w & : & Int \\ h & : & Int \\ path & : & Image \end{array}\right] \end{array}\right] : DogSeg\ \ if\ clsfr(seg) > \ threshold \tag{3}$$

4.2 Predicates

Classification, one of the major cornerstones of the model, has the power to add regions to the witness cache of a given predicate. In order to add entities to the witness cache, potential candidate regions are queried. If the result of the query is positive, the region's record is considered a witness of the predicate (see (4)).

$$\left[\begin{array}{lll} seg & = & \left[\begin{array}{rcl} x & = & 10 \\ y & = & 9 \\ w & = & 473 \\ h & = & 300 \\ path & = & image.jpg \end{array}\right] \\ z & = & a_{1032844} \end{array}\right] : \left[\begin{array}{lll} c & : & dog(z) \end{array}\right] \tag{4}$$

4.3 Objects as Witnesses of a PType

In the previous section, I show how regions of the image can be identified as evidence for a certain predicate. However, this alone does not cover any TTR parse of a question. To illustrate, think of a basic example - *Is there a dog?*. This should be modelled like so:

$$\left[\begin{array}{lll} z & : & Ind \\ c & : & \langle \lambda v : Ind.dog(v), z\rangle \end{array}\right] \tag{5}$$

If the system has already classified one of the *obj*s as being of type *c : dog(z)* and there exists a corresponding *z* in the image record, the system will come to the conclusion that the image is in fact a witness for the question type:

$$\left[\begin{array}{lll} obj_2 & = & \left[\begin{array}{lll} seg & = & \left[\begin{array}{rcl} x & = & 10 \\ y & = & 9 \\ w & = & 473 \\ h & = & 300 \\ path & = & image.jpg \end{array}\right] \\ z & = & a_{1032844} \end{array}\right] \\ z_2 & = & a_{1032844} \end{array}\right] : \left[\begin{array}{lll} z & : & Ind \\ c & : & \langle \lambda v : Ind.dog(v), z\rangle \end{array}\right] \tag{6}$$

As shown in (6), classification in Visual TTR is a type judgment. Iff the answer to the question is *yes*, the image is a Record of the Record Type of the question. For example, the picture is an instance of the *kind of situations where there is a dog*. The surface representation of the image does not change. However, type judgments were made - on the basis of the question, the perceptual classifiers and the image. Figure 2 (above) provides a visualization of the information that the model uses to make a determination about the predicate type. The image, the object bounding boxes, and the scores produced by the classifier.

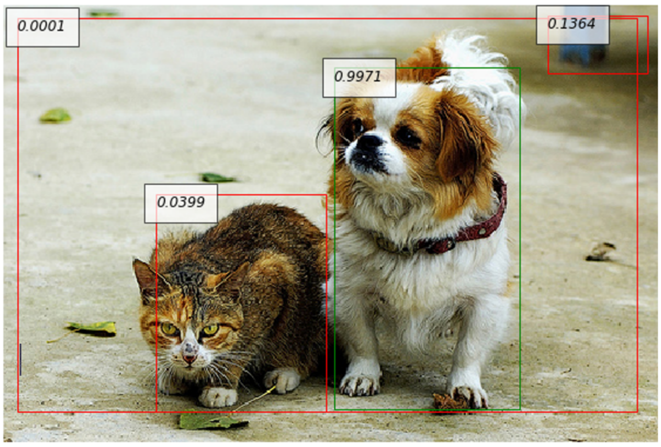

Figure 2: (Above) The image, with bounding boxes and the scores of the *dog*-classifier.

(Right) The formal representation of the image, with bounding boxes (*x,y,w,h*), the *path*, and the object ids (*z*).

$$\left[\begin{array}{lcl}
obj_0 & = & \left[\begin{array}{lcl} seg & = & \left[\begin{array}{rcl} x & = & 252 \\ y & = & 47 \\ w & = & 142 \\ h & = & 260 \\ path & = & 2401247.jpg \end{array}\right] \\ z & = & a_{401400} \end{array}\right] \\
z_0 & = & a_{401400} \\
obj_1 & = & \left[\begin{array}{lcl} seg & = & \left[\begin{array}{rcl} x & = & 116 \\ y & = & 143 \\ w & = & 130 \\ h & = & 166 \\ path & = & 2401247.jpg \end{array}\right] \\ z & = & a_{401401} \end{array}\right] \\
z_1 & = & a_{401401} \\
obj_2 & = & \left[\begin{array}{lcl} seg & = & \left[\begin{array}{rcl} x & = & 10 \\ y & = & 9 \\ w & = & 473 \\ h & = & 300 \\ path & = & 2401247.jpg \end{array}\right] \\ z & = & a_{401403} \end{array}\right] \\
z_2 & = & a_{401403} \\
obj_3 & = & \left[\begin{array}{lcl} seg & = & \left[\begin{array}{rcl} x & = & 415 \\ y & = & 7 \\ w & = & 76 \\ h & = & 44 \\ path & = & 2401247.jpg \end{array}\right] \\ z & = & a_{401416} \end{array}\right] \\
z_3 & = & a_{401416}
\end{array}\right]$$

5 Conclusions

While the system described in this paper is not yet a full-fledged Q&A system, it shows that TTR is a suitable formalism for the task of building and querying an understanding of an image. In order to reliably measure the effectiveness of the model, a proper training set is necessary. For example, this could mean the significant expansion of its grammar so that is covers a Visual Q&A dataset such as VQA v2 (Goyal et al., 2017). The expansion of the grammar is desirable also because being able to model more semantic nuance (e.g. background/foreground) is one of the major benefits of using TTR in the first place.

In this paper, I pay particular attention to the formal core of this system. A necessary aspect of such a model that I have glossed over is the parser. There is no off-the-shelf English to TTR parser, so the model does require the person implementing it to write a grammar. This has the disadvantage of limiting the model's coverage (and having to write a grammar). The advantage of a custom grammar is that it is possible to model domain-specific semantic phenomena. For example, *Is there a cat?* and *Is this a cat?* evoke the same classifier. The former applies the classifier to the whole image, while the latter applies it to all objects in the image.

A further upside to the model proposed here is transparency. It is possible for a human observer to examine the judgments that the model made when trying to answer a question. This is not very exciting in the *dog* example, but should become useful for questions that require multiple perceptual type judgments.

Another avenue for further work on the model would be to implement it in an agent-based system as described in Matsson (2018). In such a setting, judgments made about an image can become persistent additions to the agent's knowledge base.

References

Cooper, R. (2017). PyTTR. https://github.com/GU-CLASP/pyttr.

Cooper, R. and J. Ginzburg (2015). *Type Theory with Records for Natural Language Semantics**, Chapter 12, pp. 375–407. John Wiley & Sons, Ltd.

Everingham, M., S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision 111*(1), 98–136.

Goyal, Y., T. Khot, D. Summers-Stay, D. Batra, and D. Parikh (2017). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Krishna, R., Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Lin, T., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: common objects in context. *CoRR abs/1405.0312*.

Matsson, A. (2018). Implementing perceptual semantics in type theory with records). Master's thesis, University of Gothenburg.

Redmon, J., S. K. Divvala, R. B. Girshick, and A. Farhadi (2015). You only look once: Unified, real-time object detection. *CoRR abs/1506.02640*.

Schlangen, D. (2019). Natural language semantics with pictures: Some language vision datasets and potential uses for computational semantics.

Schlangen, D., S. Zarriess, and C. Kennington (2016). Resolving references to objects in photographs using the words-as-classifiers model.

Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition.