Heriot-Watt University

Heriot-Watt University
Research Gateway

Comparing dialogue strategies for learning grounded language from human tutors

Yu, Yanchao; Lemon, Oliver; Eshghi, Arash

Published in:
JerSem

Publication date:
2016

Document Version
Peer reviewed version

Link to publication in Heriot-Watt University Research Portal

Citation for published version (APA):
Yu, Y., Lemon, O., & Eshghi, A. (2016). Comparing dialogue strategies for learning grounded language from human tutors. In J. Hunter, M. Simons, & M. Stone (Eds.), JerSem: Proceedings of the 20th Workshop on the Semantics and Pragmatics of Dialogue (pp. 44-54). (SemDial Proceedings). Rutgers University.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Comparing dialogue strategies for learning grounded language from human tutors

Yanchao Yu
Interaction Lab
Heriot-Watt University
y.yu@hw.ac.uk

Oliver Lemon
Interaction Lab
Heriot-Watt University
o.lemon@hw.ac.uk

Arash Eshghi
Interaction Lab
Heriot-Watt University
a.eshghi@hw.ac.uk

Abstract

We address the problem of interactively learning perceptually grounded word meanings in a multimodal dialogue system. Human tutors can correct, question, and confirm the statements of a dialogue agent which is trying to interactively learn the meanings of perceptual words, e.g. colours and shapes. We show that different learner and tutor dialogue strategies lead to different learning rates, accuracy of learned meanings, and effort/costs for human tutors. For example, we show that a learner which can handle corrections in dialogue, and its own uncertainty about what it sees, can learn meanings that are as accurate as a fully-supervised learner, but with less cost/effort to the human tutor.

1 Introduction

Identifying, classifying and talking about objects or events in the surrounding environment are key capabilities for intelligent, goal-driven systems that interact with other agents and the external world (e.g. smart phones, robots, and other automated systems), as well as for image search/retrieval systems. To this end, there has recently been a surge of interest and significant progress made on a variety of related tasks, including generation of Natural Language (NL) descriptions of images, or identifying images based on NL descriptions (Karpathy and Li, 2015; Bruni et al., 2014; Socher et al., 2014). Another strand of work has focused on learning to generate object descriptions and object classification based on low level concepts/features (such as colour, shape and material), enabling systems to identify and describe novel, unseen images (Farhadi et al., 2009; Silberer and Lapata, 2014; Sun et al., 2013).

Our goal is to build *interactive* systems that can learn grounded word meanings relating to their perceptions of real-world objects – this is different from previous work such as e.g. (Roy, 2002), that learn groundings from descriptions without any interaction, and more recent work using Deep Learning methods (e.g. (Socher et al., 2014)).

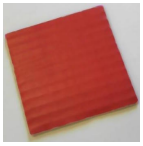
Dialogue	**Image**	**Final semantics**
S: Is this a green square? T: No it's red S: Thanks.		$\begin{bmatrix} x_{=o1} & : & e \\ p2 & : & red(x) \\ p3 & : & square(x) \end{bmatrix}$
T: What can you see? S: something orange. What is it? T: A circle. S: Thanks.		$\begin{bmatrix} x1_{=o2} & : & e \\ p & : & circle(x1) \\ p1 & : & orange(x1) \\ p2 & : & see(sys, x1) \end{bmatrix}$

Figure 1: Interactively agreed semantic contents

Most machine learning approaches to this type of problem rely on training data of high quantity with no possibility of online error correction. Furthermore, they are unsuitable for robots and multimodal systems that need to continuously, and incrementally learn from the environment, and may encounter objects they haven't seen in training data. These limitations are likely to be alleviated if systems can learn concepts, as and when needed, from situated dialogue with humans. Interaction with a human tutor also enables systems to take initiative and seek the particular information they need or lack by e.g. asking questions with the highest information gain (see e.g. (Skocaj et al., 2011), and Fig. 1). For example, a robot could ask questions to learn the colour of a "square" or to request to be presented with more "red" things to improve its performance on the concept (see e.g. Fig. 1). Furthermore, such systems could allow for meaning negotiation in the form of clarification interactions with the tutor.

This setting means that the system must be *trainable from little data, compositional, adaptive,*

and able to handle natural human dialogue with all its glorious context-sensitivity and messiness – for instance so that it can learn visual concepts suitable for specific tasks/domains, or even those specific to a particular user. Interactive systems that learn continuously, and over the long run from humans need to do so *incrementally*, *quickly*, and *with minimal effort/cost to human tutors*.

In this paper, we first outline an implemented dialogue system that integrates an incremental, semantic grammar framework, especially suited to dialogue processing – Dynamic Syntax and Type Theory with Records (DS-TTR[1] (Kempson et al., 2001; Eshghi et al., 2012)) with visual classifiers which are learned during the interaction, and which provide perceptual grounding for the basic semantic atoms in the semantic representations (Record Types in TTR) produced by the parser (see Fig. 1, Fig. 2 and section 3). In effect, the dialogue with the tutor continuously provides semantic information about objects in the scene which is then fed to online classifiers in the form of training instances. Conversely, the system can utilise the grammar and its existing knowledge about the world, encoded in the meanings it has already learned, to make reference to and formulate questions about the different attributes of an object identified in the visual scene.[2].

We then go on to use this system, in interaction with a simulated human tutor, to test hypotheses about how the accuracy of learned meanings, learning rates over time, and the overall cost/effort for the human tutor is affected by different dialogue policies and capabilities.

2 Related work

In this section, we will present an overview of vision and language processing systems, as well as multi-modal systems that learn to associate them. We compare them along two main dimensions: *Visual Classification methods: offline vs. online* and *the kinds of representation learned/used*.

Online vs. Offline Learning. A number of implemented systems have shown good performance on classification as well as NL-description of novel physical objects and their attributes, either using offline methods as in (Farhadi et al., 2009; Lampert et al., 2014; Socher et al., 2013; Kong et al., 2013), or through an incremental learning process, where the system's parameters are updated after each training example is presented to the system (Furao and Hasegawa, 2006; Zheng et al., 2013; Kristan and Leonardis, 2014). For the interactive learning task presented here, only the latter is appropriate, as the system is expected to learn from its interactions with a human tutor over a period of time. Shen & Hasegawa (2006) propose the SOINN-SVM model that re-trains linear SVM classifiers with data points that are clustered together with all the examples seen so far. The clustering is done incrementally, but the system needs to keep all the examples so far in memory. Kristian & Leonardis (2014), on the other hand, propose the oKDE model that continuously learns categorical knowledge about visual attributes as probability distributions over the categories (e.g. colours). However, when learning from scratch, it is unrealistic to predefine these concept groups (e.g. that red, blue, and green are colours). Systems need to learn for themselves that, e.g. colour is grounded in a specific sub-space of an object's features. For the visual classifiers, we therefore assume no such category groupings here, and instead learn individual binary classifiers for each visual attribute (see section 3.1 for details).

Distributional vs. Logical Representations. Learning to ground natural language in perception is one of the fundamental problems in Artificial Intelligence. There are two main strands of work that address this problem: (1) those that learn distributional representations using Deep Learning methods: this often works by projecting vector representations from different modalities (e.g. vision and language) into the same space in order to be able to retrieve one from the other (Socher et al., 2014; Karpathy and Li, 2015; Silberer and Lapata, 2014); (2) those that attempt to ground symbolic logical forms, obtained through semantic parsing (Tellex et al., 2014; Kollar et al., 2013; Matuszek et al., 2014) in classifiers of various entities types/events/relations in a segment of an image or a video. Perhaps one advantage of the latter over the former method, is that it is strictly compositional, i.e. the contribution of the meaning of an individual word, or semantic atom, to the whole representation is clear, whereas this is hard to say about the distributional models. As noted, our work also uses the latter methodology, though it is

[1] Download from `http://dylan.sourceforge.net`

[2] Here we assume that the words being grounded are in the lexicon, i.e. that their syntactic and semantic type are known: we leave the problem of grammar induction to one side here, though see (Eshghi et al., 2013)

dialogue, rather than sentence semantics that we care about. Most similar to our work is probably that of Kennington & Schlangen (2015) who learn a mapping between individual words - rather than logical atoms - and low-level visual features (e.g. colour-values) directly. The system is compositional, yet does not use a grammar (the compositions are defined by hand). Further, the groundings are learned from pairings of object references in NL and images rather than from dialogue.

What sets our approach apart from others is: a) that we use a domain-general, incremental semantic grammar with principled mechanisms for parsing and generation; b) Given the DS model of dialogue (Eshghi et al., 2015), representations are constructed jointly and interactively by the tutor and system over the course of several turns (see Fig. 1); c) perception and NL-semantics are modelled in a single logical formalism (TTR); d) we effectively induce an ontology of atomic types in TTR, which can be combined in arbitrarily complex ways for generation of complex descriptions of arbitrarily complex visual scenes (see e.g. (Dobnik et al., 2012) and compare this with (Kennington and Schlangen, 2015), who do not use a grammar and therefore do not have logical structure over grounded meanings).

3 System Architecture

We have developed a system to support an attribute-based object learning process through natural, incremental spoken dialogue interaction. The architecture of the system is shown in Fig. 2. The system has two main modules: a vision module for visual feature extraction and classification; and a dialogue system module using DS-TTR. Below we describe these components individually and then explain how they interact.

3.1 Attribute-based Classifiers used

Yu et. al (2015a; 2015b) point out that neither multi-label classification models nor 'zero-shot' learning models show acceptable performance on attribute-based learning tasks. Here, we instead use Logistic Regression SVM classifiers with Stochastic Gradient Descent (SGD) (Zhang, 2004) to incrementally learn attribute predictions.

All classifiers will output attribute-based label sets and corresponding probabilities for novel unseen images by predicting binary label vectors. We build visual feature representations to learn classifiers for particular attributes, as explained below.

3.1.1 Visual Feature Representation

In contrast with previous work (Yu et al., 2015a; Yu et al., 2015b), to reduce feature noise through the learning process, we simply extract a a 1280-dimensional feature vector consisting of only two base feature categories, i.e. the colour space for colour attributes, and a 'bag of visual words' for the object shapes/class (as shown in Fig. 2).

Colour descriptors, consisting of HSV colour space values, are extracted for each pixel and then are quantized to a 16×4×4 HSV matrix. These descriptors inside the bounding box are binned into individual histograms. Meanwhile, a bag of visual words is built in PHOW descriptors using a visual dictionary (that is pre-defined with a handmade image set). These visual words are calculated using 2x2 blocks, a 4-pixel step size, and quantized into 1024 k-means centres.

3.2 Dynamic Syntax and Type Theory with Records

Dynamic Syntax (DS) a is a word-by-word incremental semantic parser/generator, based around the Dynamic Syntax (DS) grammar framework (Cann et al., 2005) especially suited to the fragmentary and highly contextual nature of dialogue. In DS, dialogue is modelled as the interactive and incremental construction of contextual and semantic representations (Eshghi et al., 2015). The contextual representations afforded by DS are of the fine-grained semantic content that is jointly negotiated/agreed upon by the interlocutors, as a result of processing questions and answers, clarification requests, corrections, acceptances, etc. We cannot go into any further detail here due to lack of space, but proceed to briefly describe Type Theory with Records, the formalism in which the DS contextual/semantic representations are couched.

Type Theory with Records (TTR) is an extension of standard type theory shown to be useful in semantics and dialogue modelling (Cooper, 2005; Ginzburg, 2012). TTR is particularly well-suited to our problem here as it allows information from various modalities, including vision and language, to be represented within a single semantic framework (see e.g. Larsson (2013); Dobnik et al. (2012) who use it to model the semantics of spatial language and perceptual classification).

In TTR, logical forms are specified as *record*

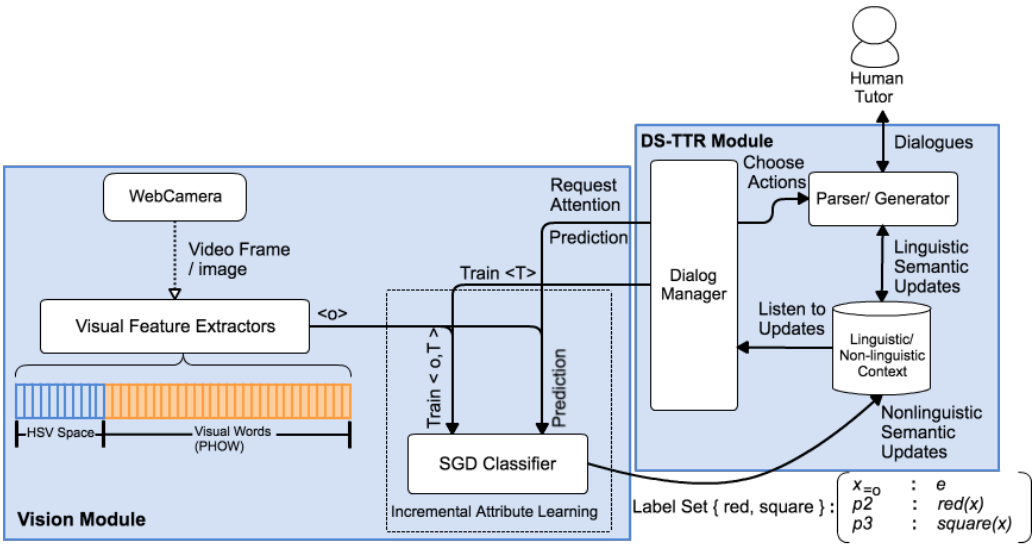


Figure 2: Architecture of the teachable system

types (RTs), which are sequences of *fields* of the form $[l : T]$ containing a label l and a type T. RTs can be witnessed (i.e. judged true) by *records* of that type, where a record is a sequence of label-object pairs $[l = v]$. We say that $[l = v]$ is of type $[l : T]$ just in case v is of type T. Importantly for us here, TTR has a subtyping relation, in terms of which inference is defined; but it also allows semantic information to be incrementally specified, i.e. record types can be indefinitely extended with more information/constraints. This is a key feature since it allows the system to encode *partial* knowledge about objects, and for this knowledge to be extended in a principled way, as and when it becomes available.

For further detail on TTR, see Cooper (2005) and Dobnik et al. (2012) among others.

3.3 Integration

Fig. 2 shows how the various parts of the system interact. At any point in time, the system has access to an ontology of (object) types and attributes encoded as a set of TTR Record Types, whose individual atomic symbols, such as 'red' or 'square' are grounded in the set of classifiers trained so far.

Given a set of individuated objects in a scene, encoded as a TTR Record, the system can utilise its existing ontology to output some maximal set of Record Types characterising these objects (see e.g. Fig. 1). Since these representations are shared by the DS-TTR module, they provide a direct interface between perceptual classification and semantic processing in dialogue: they can be used directly at any point to generate utterances, or ask questions about the objects.

On the other hand, the DS-TTR parser incrementally produces Record Types (RT), representing the meaning jointly established by the tutor and the system so far. In this domain, this is ultimately one or more type judgements, i.e. that some scene/image/object is judged to be of a particular type, e.g. in Fig. 1 that the individuated object, $o1$ is a red square. These jointly negotiated type judgements then go on to provide training instances for the classifiers. In general, the training instances are of the form, $\langle O, T \rangle$, where O is an image/scene segment (an object or TTR Record), and T, a record type. T is then decomposed into its constituent atomic types $T_1 \ldots T_n$, s.t. $\bigwedge T_i = T$. The judgements $O : T_i$ are then used directly to train the classifier that grounds the T_i.

4 Experiments and Results

In general, in real-world problems, there are a variety of dialogue behaviours that human tutors might adopt to teach the learner with novel knowledge, and these might lead to different reactions from the learner/system as well as different outcomes for the recognition performance of the learned concepts/meanings, effort from the tutor and trade-offs between these. Moreover, a learner with different capabilities (described below) can also affect these performances through dialogue. Our goal in this paper is therefore to explore the effects of these dialogue behaviours and capabilities on the overall performance of the learning agent by measuring the trade-off between recog-

nition performance and tutoring cost.

4.1 Design

Before explaining the experiment configurations, there are several notions that need to be defined in terms of basic dialogue capabilities, tutor behaviours, and learner dialogue capabilities –

Basic Dialogue Capabilities: The following capabilities are explored for both the tutor and the learner (see examples in Fig. 3):

- **Listening**: this only refers to a *learner*, while the *tutor* is making a statement about a specific object/attribute;
- **Statement**: the ability for both *learners* and *tutors* to describe attributes of an object, e.g. "this is a red square" or "this is red";
- **Correction**: the ability to process corrections only from the *tutor*, e.g. "no, this is green" or "no, this is a circle';
- **Implicit/explicit confirmation**: the ability to process confirmations from the *tutor*, e.g. "Yes, it's a square";
- **Question-answering**: the ability to answer questions from both the *tutor* and the *learner*, e.g. "T: what is this? S: this is a red square.";
- **Question-asking**: the ability to ask WH or polar questions requesting correct information, e.g. "what colour is this?" or "is this a red square?".

Tutor Behaviours: Following previous work (Skočaj et al., 2009), we generally identify tutor behaviours based on how he/she treats the learner into two groups: **1) *Tutor-Driven (TD)***: The tutor always gives available information about a particular object, i.e. supervised learning (always providing labels), by directly making statements (e.g. "this is a square" or "this is a red square"). This means that the whole learning process is an unidirectional interaction only handled by the tutor. In this case, the learner only needs to listen and update its learning models (i.e. the visual classifiers) upon what information the tutor presented. **2) *Tutor-Corrected (TC)***: while the learner is describing or asking something about the object, the tutor only asks WH questions and corrects mistakes of the learner, and otherwise confirms correct statements (e.g. "T: what is this? L: this is a red square. T: yes/no, it is a green square" in Fig. 3). In contrast to the TD behaviour, the learner performs more actively to get involved with the learning process with its own predictions/knowledge. It will update its classifiers only when the tutor provides answers or confirms.

According to the previous work from Skočaj et. al. (2009), both tutor strategies are frequently adopted in a perceptual learning process, which may lead to different levels of learner involvement. They assumed that the tutor can always perform well through the entire learning process. However, this may be extremely idealised for real-world problems, in which human tutors may not always supply all their knowledge when informing about a visual object. In this paper, we therefore also take the following situations into account:

- ***"Good-Tutor" (GT)***: the tutor always gives all the labels for each image, always corrects all the mistakes of the learner, and always confirms correct statements by the learner.
- ***"Lazy-Tutor" (LT)***: this tutor only gives one of the correct labels at a time (e.g. "it's red" or "it's a square"), and only corrects one mistake at a time. It always confirms when asked to. This tutor is more similar to what we can expect from real human behaviour when teaching robots than the Good Tutor.

Learner Dialogue Capabilities: In real-world learning tasks, a learner might be required to consider several additional capabilities, which may enable it to respond to tutor behaviours in a more natural way, especially with a *Lazy-Tutor* situation. Moreover, these capabilities are also likely to help improve the overall performance of the learner at the end of learning process, i.e. achieving a better trade-off between the performance of object/attribute recognition and the cost for the tutor. In this paper, we use the dialogue system outlined above to carry out experiments with three dialogue capabilities with two levels: **1) Uncertainty (+UC/-UC)**: determines whether the learner takes into account, in its dialogue behaviour, its own subjective confidence about the attributes of the presented object. The confidence is the score assigned by any of its attribute classifiers of the object being a positive instance of an attribute (e.g. 'red') - see below for how a confidence threshold is used here. In condition +UC, the agent will not ask a question if it is confident about the answer, and it will hedge the answer to a tutor question if it is not confident, e.g. "T: What is this? L: is this a red square?". In -UC, the learner

<table>
<tr><th></th><th>TD</th><th>TC (-UC)</th><th>TC (+UC)</th></tr>
<tr><td>Good(Ideal) Tutor</td><td>T: this is red.
L: okay.
T: this is a square.
L: okay.
---------- or ----------
T: this is a red square.
L: okay.</td><td>T: what is this?
L: this is a red square.
T: no, this is a green square.
L: okay.</td><td>T: what is this?
L: is this a red square?
T: no, this is a green square.
L: okay.</td></tr>
<tr><td rowspan="3">Lazy(natural) Tutor</td><td colspan="3">Without Knowledge-Demanding (-KD)</td></tr>
<tr><td>T: this is red.
L: okay.</td><td>T: what is this?
L: this is a red square.
T: yes, it is a square.
L: okay.</td><td>T: what is this?
L: is this a red square?
T: yes , this is a square.
L: okay.</td></tr>
<tr><td colspan="3">With Knowledge-Demanding (+KD)</td></tr>
<tr><td></td><td>T: this is red.
L: okay. what shape is it?
T: this is a square.
L: okay.</td><td>T: what is this?
L: this is a red square.
T: no, this is a circle.
L: okay. Is the colour correct?
T: yes.
L: okay.</td><td>T: what is this?
L: is this a green circle?
T: no, this is a square.
L: okay. Is the colour correct?
T: no, this is red.
L: okay.</td></tr>
</table>

Figure 3: Example dialogues in different conditions (TD = tutor-driven, TC = tutor-corrected, -UC= no learner uncertainty, +UC= learner uncertainty)

is confident and always takes itself to know the attributes of the given object (as given by its currently trained classifiers), and behaves according to that assumption. **2) Knowledge-Demanding (+KD/-KD)**: this determines whether the learner can request further details/information about objects, which may be useful when interacting with a "Lazy" Tutor (described above). In condition +KD, the learner is able to request more information by asking extra questions (see Fig. 3 e.g. "what (colour/shape) is it? or "is the colour correct?". Otherwise, the learner with -KD will only update the classifiers based on the information provided.

Confidence Threshold: To determine when and how the agent properly copes with its attribute-based predictions, we use confidence-score thresholds. It consists of two values, a base threshold (e.g. 0.5) and a positive threshold (e.g. 0.9).

If the confidences of all classifiers are under the base threshold (i.e. the learner has no attribute label that it is confident about), the agent will ask for information directly from the tutor via questions (e.g. "L: what is this?").

On the other hand, if one or more classifiers score above the base threshold, then the positive threshold is used to judge to what extent the agent trusts its prediction or not. If the confidence score of a classifier is between the positive and base thresholds, the learner is not very confident about its knowledge, and will check with the tutor, e.g. "L: is this red?". However, if the confidence score of a classifier is above the positive threshold, the learner is confident enough in its knowledge not to bother verifying it with the tutor. This will lead to less effort needed from the tutor as the learner becomes more confident about its knowledge.

However, since a learner with high confidence will not ask for assistance from the tutor, a low positive threshold may reduce the opportunities that allow the tutor to correct the learner's mistakes. With an additional experiment (*note*: we will not explain it here due of lack of space), we determined a 0.5 base threshold and a 0.9 positive threshold as the most appropriate values for an interactive learning process - i.e. this preserved good classifier recognition while not requiring much effort from the tutor. In (Yu et al., 2016) we show how these thresholds can be optimised.

4.2 Experimental Setup

We carried out a set of experiments to investigate the effects of these dialogue policies on an interactive learning process with a tutor. We compare different behaviours and capabilities with two baseline policies without corrections (NC), in which the learner cannot process corrections but only confirmations from the tutor. This means that the learner can update its classifiers only when its own predictions are correct. There are several settings related to these experiments below:

Table 1: Recognition Score Table

	Yes	LowYes	LowNo	No
Yes	1	0.5	-0.5	-1
No	-1	-0.5	0.5	1

Table 2: Tutoring Cost Table

C_{inf}	C_{yes}	C_{crt}	C_{ign}	C_{turn}
1	0.25	1	0	0.15

Tutor Simulation and Policy: To run our experiment on a large-scale, we have hand-crafted an *Interactive Tutoring Simulator*, which simulates the behaviour of a human tutor[3]. The tutor policy is set up based on different tutor-based behaviours and situations as mentioned above.

Evaluation and Cross-validation: To evaluate the performance of the system in each condition, we performed a 100-fold cross validation with 500 images for training and 100 for testing within a handmade object set[4]. For each training instance, the learning system interacts with the simulated tutor. We define a **Learning Step** as comprised of 25 such dialogues. At the end of each learning step, the system is tested using the test set. The values used for the Tutoring Cost and the Recognition Score at each learning step correspond to averages across the 100 folds.

4.3 Evaluation Metrics

To test how the different dialogue capabilities and strategies affect the language learning process, we follow metrics proposed by Skočaj et al.(2009), that consist of two main evaluation measures, i.e. *Recognition Scores* and *Tutoring Costs*. We tweak the details below to reflect our own dialogue system configurations.

Recognition score: This is a metric measuring the overall accuracy of the learned word meanings / classifiers, which "rewards successful classifications (i.e. true positives and true negatives) and penalizes incorrect predictions (i.e. false positives and false negatives)" (Skočaj et al., 2009) [5]. As the proposed system considers both correctness of predicted labels and prediction confidence on learning tasks, the measure will also take the true labels with lower confidence into account, as shown in Table 1; "LowYes" means that the system made positive predictions but with lower confidence. In this case, the system can generate a polar question for requesting tutor feedback. "LowNo" is similar to "LowYes", but only works on negative predictions.

Cost: The cost measure reflects the effort needed by a human tutor in interacting with the system. Skocaj et. al. (2009) point out that a comprehensive teachable system should learn as autonomously as possible, rather than involving the human tutor too frequently. There are several possible costs that the tutor might incur, see Table 2: C_{inf} refers to the cost of the tutor providing information on a single attribute concept (e.g. "this is red" or "this is a square"), and we set this cost as 1; C_{yes} is the cost of a simple confirmation (like "yes", "right") and set it to be 0.25; C_{crt} is the cost of correction for a single concept (e.g. "no, it is blue" or "no, it is a circle") and is also set to be 1. Moreover, the number of dialogue turns from the tutor was also taken into account in measuring total cost: each single turn costs 0.15 in this experiment. These values are based on the intuition that it is just as much effort for the Tutor to provide a concept as to correct one, and that confirmation has a smaller cost, while each turn also requires a small effort from the Tutor.

Performance Score As mentioned above, an efficient Learner dialogue policy should consider both classification accuracy (Recognition score) and tutor effort (Cost). We thus defined an integrated measure – the *Performance Score* (S_{perf}) – that we use to compare the general performance across different dialogue policies and capabilities:

$$S_{perf} = \frac{S_{recog}}{C_{tutor}}$$

i.e. the ratio of Recognition Score achieved by the Learner to the effort/Cost required by the Tutor. We seek dialogue strategies that balance these metrics.

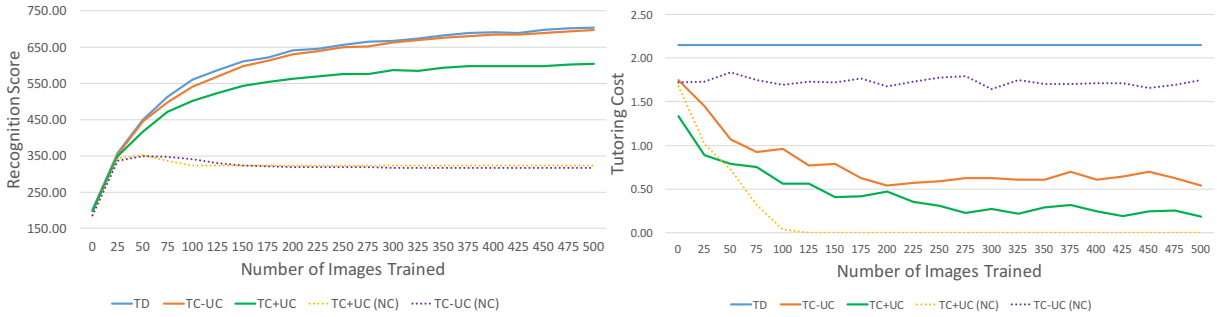
4.4 Results

We first investigate the improvement of learning performance over time for different learner policies and capabilities with an ideal tutoring situation (*Good* Tutor) (see Fig. 4). We compared both tutor policies (TD and TC) with correspond-

[3]The experiment involves hundreds of dialogues, so running this experiment with real human tutors has proven too costly at this juncture, though we plan to do this for a full evaluation of our system in the future.

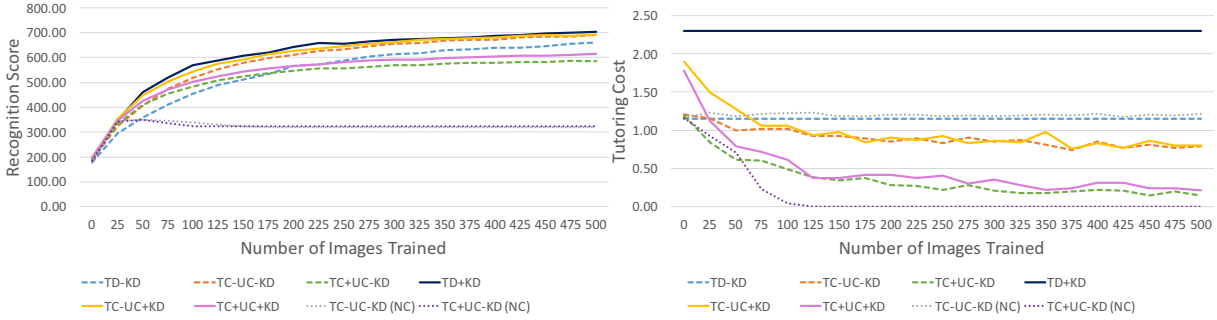
[4]All data from this paper will be made freely available.

[5]we use recognition score instead of accuracy because it better handles uncertainty predictions than accuracy, which could be more similar to a human-like learning task.



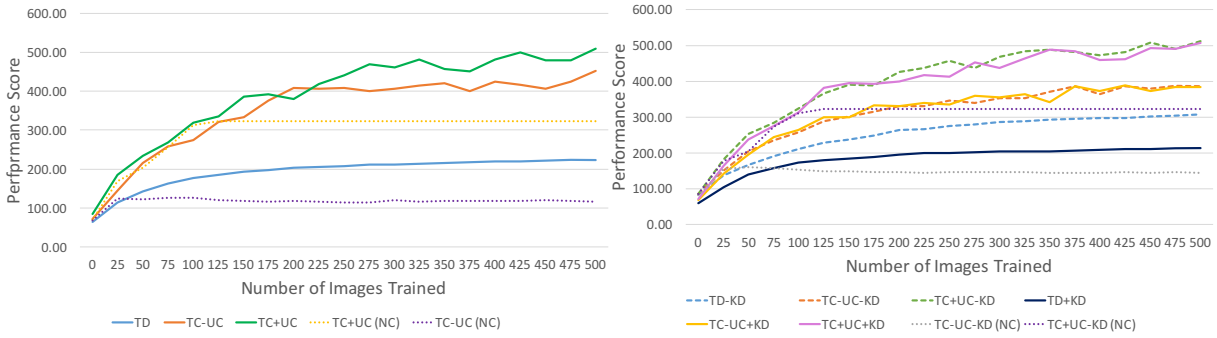
(a) Recognition Score (b) Tutoring Cost

Figure 4: Evolution of Learning Performance in the *Good Tutor* Condition (TD = tutor-driven, TC = tutor-corrected, -UC= no learner uncertainty, +UC= learner uncertainty, NC= no corrections)



(a) Recognition Score (b) Tutoring Cost

Figure 5: Evolution of Learning Performance in the *Lazy Tutor* Condition (TD = tutor-driven, TC = tutor-corrected, UC= learner uncertainty, NC= no corrections, KD= Knowledge-demanding)



(a) Recognition/Cost Ratio for "Good tutor" (b) Recognition/Cost Ratio for "Lazy Tutor"

Figure 6: Learner policy Performance with both Tutor types TD = tutor-driven, TC = tutor-corrected, UC= learner uncertainty, NC= no corrections, KD= Knowledge-demanding)

ing learner strategies and capabilities (+/-UC and NC) in terms of Recognition Score and Tutoring Cost. (Note that in the Good Tutor case, +/-KD has no effect).

Here we see that the Tutor-Driven (TD, blue line) and Tutor-Corrected without Uncertainty (TC-UC, red line) conditions gain the highest Recognition scores, while conditions without the Learner ability to process tutor corrections (NC) perform badly, as expected. In terms of Tutoring Cost though, we see that TD has a high cost while TC-UC has quite low cost. Interestingly, TC+UC (Tutor-corrected, with Uncertainty, green line), has a lower cost than both of these conditions, while still achieving a high Recognition score. This is because the Learner which is aware of its uncertainty about classifier outputs requires fewer corrections from the Tutor, while the classi-

fiers still become more accurate over time.

Similar to Fig. 4, Figs. 5a, b show the Recognition Score and Tutoring Cost respectively for the same learner strategies, but with a more natural tutoring situation (*Lazy* Tutor), and where the learner can be Knowledge-Demanding (+/-KD). In addition, Fig. 6 shows the overall performance of different learner strategies (i.e. the trade-offs between the recognition score and the tutoring cost) in the Good and Lazy Tutor situations separately.

Here, in the Good-Tutor condition, the TC-UC policy (orange line) shows better overall performance than TD (blue line) because of its lower tutoring cost. In addition, though the Uncertain Learner (TC+UC, green line) policy performs slightly worse on recognition score (this might be due to insufficient error detection and recovery), it also reduces the tutoring cost through time. Hence, this policy achieves better performance than the others in the final results (see Fig. 6a).

In terms of the *Lazy-Tutor* condition, both the TD and TC-UC policies, without Knowledge-demand (-KD), show slightly worse recognition performance than they did under the Good Tutor policy, because the learner does not gain as much knowledge from the tutor in each learning step. Whilst both policies cost much less than before for the same reason, they show better performance in the final results (as compared between Figures 6a and 6b). By contrast, as a situation with two incorrect predictions rarely occurs with the TC+UC-KD policy (for only about 20 out of 500 images), the *Lazy-Tutor* policy will not affect Recognition Score or Tutoring Cost very much for the TC+UC policy (see Fig. 5a, b). Therefore, its final performance shows a similar tendency as under the Good Tutor condition.

Moreover, the results in Figure 5 also show that a Knowledge-Demanding (+KD) learner policy may always improve recognition performance (Fig. 5a). For the Lazy Tutor condition, the conditions TC+UC+KD (pink line) and TC+UC-KD (green dotted line) have the best overall performance (Fig. 6b).

Since our ultimate goal here is to create a full dialogue system that can learn accurate concepts (word meanings) with little effort from human tutors, these results would lead us to choose a dialogue system that can can handle corrections – i.e. some variant of the Tutor Corrected system. The results show that, depending on the relative weight between Recognition Score and Tutor Cost, an optimal Learner Dialogue Policy could, for example, use TC-UC(NC) for the first 50 or 60 images, and then switch to TC+UC. We investigate such dynamic policies and their optimisation in a later study using Reinforcement Learning methods (Yu et al., 2016).

5 Conclusion

We have developed a multimodal dialogue interface to explore the effectiveness of situated dialogue with a human tutor for learning perceptually-grounded word meanings. The system integrates semantic representations from an incremental semantic parser/generator, DS-TTR, with attribute classification models that ground the semantic representations.

We compared the system's performance (its Recognition Score and Tutor Cost) under several different dialogue policies for interactive language grounding, on a hand-made dataset of simple objects. Overall, we see that dialogue interaction is important for teachable agents as it reduces the effort required from the human tutor. The fully supervised cases (TD) have a high cost for the Tutor, and equivalent final recognition performance can be reached with less effort when using a Tutor-Corrected (TC) dialogue policy where the Learner can process corrections in dialogue. Final Recognition performance is slightly less good with learners which take their own uncertainty into account (TC+UC), but they require much less effort from Tutors, resulting in better overall performance.

Ongoing work explores full Learner dialogue policies (i.e. turn-based decisions about what to say next) and their optimisation using Reinforcement Learning methods (Rieser and Lemon, 2011; Yu et al., 2016).

Acknowledgements

This research is supported by the EPSRC, under grant number EP/M01553X/1 (BABBLE project[6]), and by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688147 (MuMMER[7]).

[6] https://sites.google.com/site/hwinteractionlab/babble

[7] http://mummer-project.eu/

References

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1–47).

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

Robin Cooper. 2005. Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Simon Dobnik, Robin Cooper, and Staffan Larsson. 2012. Modelling language, action, and perception in type theory with records. In *Proceedings of the 7th International Workshop on Constraint Solving and Language Processing (CSLPÄô12)*, pages 51–63.

Arash Eshghi, Julian Hough, Matthew Purver, Ruth Kempson, and Eleni Gregoromichelaki. 2012. Conversational interactions: Capturing dialogue dynamics. In S. Larsson and L. Borin, editors, *From Quantification to Conversation: Festschrift for Robin Cooper on the occasion of his 65th birthday*, volume 19 of *Tributes*, pages 325–349. College Publications, London.

Arash Eshghi, Julian Hough, and Matthew Purver. 2013. Incremental grammar induction from child-directed dialogue utterances. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 94–103, Sofia, Bulgaria, August. Association for Computational Linguistics.

A. Eshghi, C. Howes, E. Gregoromichelaki, J. Hough, and M. Purver. 2015. Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, London, UK. Association for Computational Linguisitics.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*.

Shen Furao and Osamu Hasegawa. 2006. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106.

Jonathan Ginzburg. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137.

Ruth Kempson, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL-IJCNLP)*. Association for Computational Linguistics.

Thomas Kollar, Jayant Krishnamurthy, and Grant Strimel. 2013. Toward interactive grounded language acqusition. In *Robotics: Science and Systems*.

Xiangnan Kong, Michael K. Ng, and Zhi-Hua Zhou. 2013. Transductive multilabel learning via label set propagation. *IEEE Trans. Knowl. Data Eng.*, 25(3):704–719.

Matej Kristan and Ales Leonardis. 2014. Online discriminative kernel density estimator with gaussian kernels. *IEEE Trans. Cybernetics*, 44(3):355–365.

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465.

Staffan Larsson. 2013. Formal semantics for perceptual classification. *Journal of logic and computation*.

Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from unscripted deictic gesture and language for human-robot interactions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2556–2563.

Verena Rieser and Oliver Lemon. 2011. Learning and evaluation of dialogue strategies for new applications: Empirical methods for optimization from small data sets. *Computational Linguistics*, 37(1):153–196.

Deb Roy. 2002. A trainable visually-grounded spoken language generation system. In *Proceedings of the International Conference of Spoken Language Processing*.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 721–732, Baltimore, Maryland, June. Association for Computational Linguistics.

Danijel Skocaj, Matej Kristan, Alen Vrecko, Marko Mahnic, Miroslav Janícek, Geert-Jan M. Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, and Kai Zhou. 2011. A system for interactive learning in dialogue with a tutor. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011, San Francisco, CA, USA, September 25-30, 2011*, pages 3387–3394.

Danijel Skočaj, Matej Kristan, and Aleš Leonardis. 2009. Formalization of different learning strategies in a continuous learning framework. In *Proceedings of the Ninth International Conference on Epigenetic Robotics; Modeling Cognitive Development in Robotic Systems*, pages 153–160. Lund University Cognitive Studies.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, pages 935–943, Lake Tahoe, Nevada, USA.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Yuyin Sun, Liefeng Bo, and Dieter Fox. 2013. Attribute based object identification. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2096–2103. IEEE.

Stefanie Tellex, Pratiksha Thaker, Joshua Mason Joseph, and Nicholas Roy. 2014. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, 94(2):151–167.

Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2015a. Comparing attribute classifiers for interactive language grounding. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 60–69, Lisbon, Portugal, September. Association for Computational Linguistics.

Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2015b. Interactive learning through dialogue for multimodal language grounding. In *SemDial 2015, Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue, Gothenburg, Sweden, August 24-26 2015*, pages 214–215.

Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2016. Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In *(under review)*.

Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM.

Jun Zheng, Furao Shen, Hongjun Fan, and Jinxi Zhao. 2013. An online incremental learning support vector machine for large-scale data. *Neural Computing and Applications*, 22(5):1023–1035.