Living a discrete life in a continuous world: Reference in cross-modal entity tracking

Gemma Boleda[1] Sebastian Padó[2] Nghia The Pham[3] Marco Baroni[4]
[1]Universitat Pompeu Fabra
[2]Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart
[3]Center for Mind/Brain Sciences, University of Trento
[4]Facebook Artificial Intelligence Research, Paris
gemma.boleda@upf.edu, pado@ims.uni-stuttgart.de,
thenghia.pham@unitn.it, mbaroni@fb.com

Abstract

Reference is a crucial property of language that allows us to connect linguistic expressions to the world. Modeling it requires handling both continuous and discrete aspects of meaning. Data-driven models excel at the former, but struggle with the latter, and the reverse is true for symbolic models.

This paper (a) introduces a concrete referential task to test both aspects, called cross-modal entity tracking; (b) proposes a neural network architecture that uses external memory to build an *entity library* inspired in the DRSs of DRT, with a mechanism to dynamically introduce new referents or add information to referents that are already in the library.

Our model shows promise: it beats traditional neural network architectures on the task. However, it is still outperformed by Memory Networks, another model with external memory.

1 Introduction

Language combines discrete and continuous facets, as exemplified by the phenomenon of *reference* (Frege, 1892; Abbott, 2010): When we refer to an object in the world with the noun phrase *the mug I bought*, we use content words such as *mug*, which are notoriously fuzzy or vague in their meaning (Van Deemter, 2012; Murphy, 2002) and are best modeled through continuous means (Boleda and Herbelot, 2016). Once the referent for the mug has been established, however, it becomes a linguistic entity that we can manipulate in a largely discrete fashion, retrieving it and updating it with new information as needed (*Remember the mug I bought? My brother stole it!* Kamp and Reyle, 1993). Put differently, managing reference requires two distinct abilities:

1. The ability to *categorize*, that is, to recognize that different entities are equivalent with regard to some concept of interest (e.g. two mugs, two instances of the "things to take on a camping trip" category; Barsalou, 1983). This implies being able to aggregate seemingly diverse objects.
2. The ability to *individuate*, that is, to keep entities distinct even if they are similar with regard to many attributes (e.g. two pieces of pink granite that were collected in different national parks). This implies being able to keep seemingly similar things apart.

Data-driven, continuous models are very good at categorizing, but not at individuating, and the reverse holds for symbolic models (Boleda and Herbelot, 2016). Our long-term research goal is to build a **continuous computational model of reference** that emulates discrete referential mechanisms such as those defined in DRT (Kamp and Reyle, 1993); here we present initial work towards that goal, with two specific contributions.

Our first contribution is an experimental task (and associated dataset), **cross-modal entity tracking**, that tests the ability of computational models to refer successfully in a setting where they are required to both categorize and individuate entities. The task presents different entities (represented by pictures) repeatedly, each time with a different, linguistically conveyed attribute (e.g. a given mug is presented once with the attribute *bought* and once with *stolen*). The category label ("mug") is not given at exposure time. The task is to choose the picture of the entity that corresponds to a linguistic query that combines category information with attribute information (e.g. simulating "the mug that was bought and stolen"), among the set of all the entities presented in a given sequence. The sequences in each datapoint of our dataset contain confounders that make the task challenging: Other entities with the same category but only one matching attribute (e.g. a different mug that was bought and stored), and other entities with the same attributes but a different category (e.g. a chair that was bought and stolen). Therefore, the task requires models to 1) correctly categorize entities, recognizing which images belong to the category in the query (something that is hard for symbolic models), 2) individuate and track them, being able to distinguish among different entities based on visual and linguistic cues provided at different time steps (something that is hard for continuous models).

In DRT terms (Kamp and Reyle, 1993), each entity exposure either introduces a new discourse referent or updates the representation of an old referent with new information. To solve the task successfully, the model needs to decide, for each incoming exposure, whether to aggregate it with a previously known referent (in DRT, this means introducing an equation between two referents), or to treat it as a new referent.

Our second contribution is a neural network architecture with a module for referent representations: **DIstributed model of REference**, **DIRE**. DIRE uses the concept of *external memory* from deep learning (Joulin and Mikolov, 2015; Graves et al., 2016) to build an entity library for an exposure sequence that conceptually corresponds to the set of DRT discourse referents, using similarity-based reasoning on distributed representations to decide between aggregating and initializing entity representations. In contrast to symbolic implementations of DRT (Bos, 2008), which manipulate discourse referents on the basis of manually specified algorithms, DIRE learns to make these decisions directly from observing reference acts using end-to-end training. We see our paper as a first, modest step in the direction of data-driven learning of DRT-like behavior, and are of course still far from learning anything resembling a fully fledged DRT system.

2 Cross-modal Entity Tracking: Task and Data

Task. Imagine an office, with a desk where there are three mugs and other objects. Adam tells Barbara that he just bought two of the mugs and he particularly likes the one on the right. Later they are in the kitchen, and Adam, busy preparing coffee, asks Barbara: "Remember the mugs I bought? Could you please bring the one I like?". To pick the right mug from the office, Barbara must correctly categorize the objects on the desk (identify which ones are mugs) and individuate them via their properties (singling out the one Adam is asking for). Also, she must combine visual and linguistically conveyed properties of the objects: Visual properties tell her which ones are mugs, the properties that Adam told her about help her pick the right one. Our *cross-modal entity tracking task* emulates this kind of situation. Our current study uses a simplified version of the task that allows us to carefully control all the variables involved.

We operationalize the task as one of pointing to real-life pictures of objects. Figure 1 shows a simplified example. We sample six entities belonging to two categories (in the example, where only three entities are shown, barkeepers and soldiers). Each entity is represented by one image (that is, barkeeper A is always represented by the same image). We also sample different attributes, which are compatible with both categories (in the example, "instructed", "evaluated", "amused"). In the exposure phase, we present each entity (image) twice at different time steps, each time with one of the attributes. In this phase, the category of the entity in the image is not given to the model, only the images are. At query time, we use a linguistic query with one category (e.g., "barkeeper") and two attributes (e.g., "instructed and evaluated"). The task is to retrieve the image of the entity that corresponds to the query. To solve it, it is not enough to rely on categorization or object labeling (in the actual task, there are always three entities belonging to the category in the query), nor is it enough to rely on attribute information (there will always be three entities for each attribute, and two for the combination of attributes in the query). Note that one important simplification we make, with respect to a real-life scenario, is that an entity is always represented by the

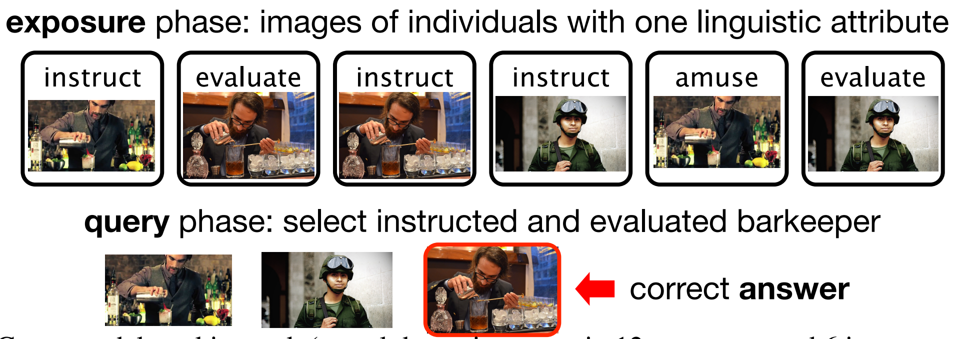
exposure phase: images of individuals with one linguistic attribute
instruct
evaluate
instruct
instruct
amuse
evaluate
query phase: select instructed and evaluated barkeeper
correct **answer**

Figure 1: Cross-modal tracking task (actual datapoints contain 12 exposures and 6 images to pick from).

very same image. The current setup is nevertheless already very challenging for current models, as the experiments below will show. Indeed, to succeed in the task a model must correctly associate the category in the query with images of the right object, it must develop a mechanism to index entities based on the images representing them, and it must learn to correctly accumulate over time the different attributes to be stored with each entity.

The task is related to coreference resolution (see Poesio et al., 2017, for a recent survey), but focuses on identifying language-external objects from images rather than mentions of a referent in text; to Visual Question Answering (Antol et al., 2015), but it cannot be solved with visual information alone; and to Referring Expression Generation (Krahmer and Van Deemter, 2012), but involves identification rather than generation.

Dataset. We have constructed a dataset for the task containing 40k sequences for training, 5k for validation and 10k for testing.[1] It is assembled on the basis of 2k object categories with 50 ImageNet[2] images each, sampled from a larger dataset (Lazaridou et al., 2015). These are natural images, which makes the task challenging. The object categories given in the queries are those specified in ImageNet.

We build a set of linguistic attributes for each object by first extracting the 500 most associated, and thus plausible, syntactic neighbors for the category according to the DM resource (Baroni and Lenci, 2010). This excludes nonsensical combinations such as *repair:dog*. We further retain only (relatively) abstract verbs taking the target item as direct object.[3] This is because (a) concrete verbs are likely to have strong visual correlates that could conflict with the image (cf. *walk dog*); and (b) referential expressions routinely successfully mix concrete and abstract cues (e.g., *the dog I own*). We remove all verbs with a score over 2.5 (on a 1–5 scale) in the concreteness norms of Brysbaert et al. (2014).

We then construct each sequence as follows. First, we sample two random categories, and three random entities (distinct images) for each category (total: six entities). We then sample three attributes compatible with both categories, giving us three attribute sets of size two (a1+a2, a1+a3, a2+a3) to associate with the entities. We create a completely balanced set of exposures by randomly pairing up each of the three entities of each category with each of the three attribute sets. Since this process gives us two exposures for each entity (one with the first attribute, one with the second), it yields a sequence of twelve exposures. The query is a random combination of a category and two attributes, guaranteed to match exactly one entity.

3 The DIRE Model

The core novelty of our model, DIRE (for DIstributed model of REference), is a method to dynamically construct an *entity library*, conceptually inspired in 1) the DRSs of DRT,[4] and 2) Joulin and Mikolov

[1] Available at `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/dire`.

[2] `http://imagenet.stanford.edu`

[3] We use the base form of verbs rather than the past participle for simplicity.

[4] DRSs represent many types of information; as explained above, here we focus on entity-related information.

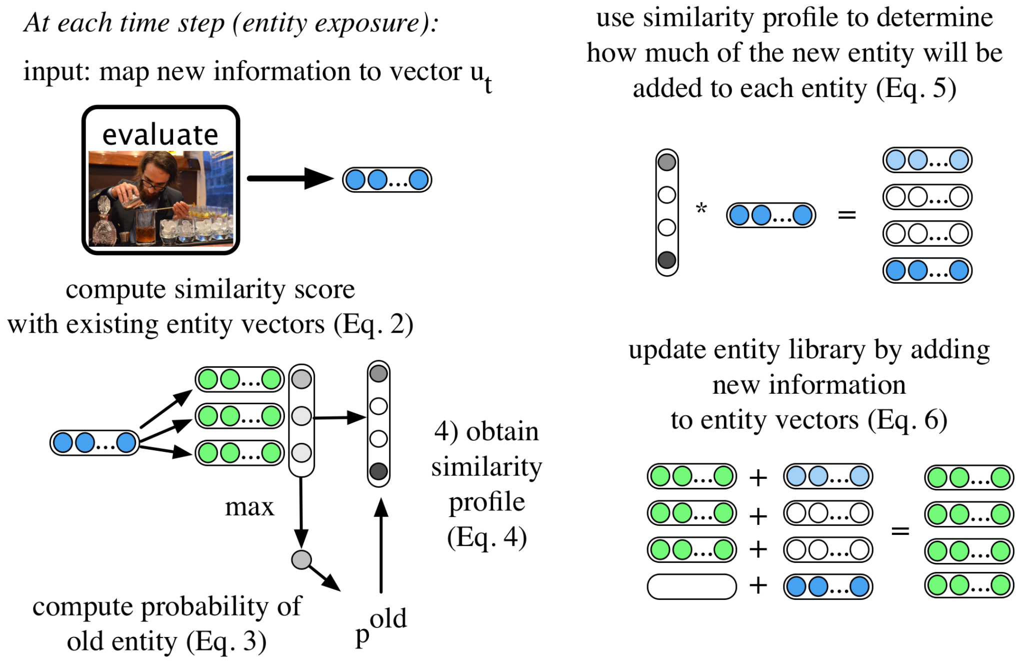
At each time step (entity exposure):
input: map new information to vector u_t
evaluate
compute similarity score with existing entity vectors (Eq. 2)
max
compute probability of old entity (Eq. 3)
p^{old}
4) obtain similarity profile (Eq. 4)
use similarity profile to determine how much of the new entity will be added to each entity (Eq. 5)
*
=
update entity library by adding new information to entity vectors (Eq. 6)
+
=

Figure 2: Building the DIRE entity library.

(2015) and Graves et al. (2016), who simulate discrete memory-building operations in a differentiable continuous setup.[5] The model is a feed-forward network enhanced with a dynamic memory (the entity library), as well as mechanisms to interact with it.

The entity library is updated after reading an input exposure by either creating a new entity slot for the exposure, or adding the exposure contents to an existing entity slot. This decision is based on the similarity between the current input and the entities already in the library. This generic mechanism (Section 3.1) can be applied in any setting that accumulates information about entities over time. We explain how we use it for our cross-modal tracking task in Section 3.2.

3.1 Building the DIRE Entity Library

Figure 2 depicts the entity library building mechanism. The input to the model is a set of subsequent exposures $x_1, x_2, \ldots, x_n$ which are represented by vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$. At the t-th exposure, the entity library is updated to state $\mathbf{E}_t$ as follows. The first exposure vector $\mathbf{u}_1$ is added to the entity library as is (Equation 1). For $\mathbf{u}_{i>1}$, we obtain a similarity profile $\mathbf{s}_i$ by taking its dot product with the entity vectors in the library (Equation 2; note that $\mathbf{s}_i$ has $i-1$ dimensions):

$$\mathbf{E}_1 = \mathbf{u}_1^\intercal \tag{1}$$

$$\mathbf{s}_i = \mathbf{E}_{i-1}\mathbf{u}_i \tag{2}$$

The maximum similarity to an existing entity, $s_i^{max} = \max(\mathbf{s}_i)$, cues whether x_i is an instance of an entity that has already been encountered before. We transform s_i^{max} into p_i^{old}, the probability that exposure x_i corresponds to an "old" entity, as follows (with the scalar w, b parameters shared across all exposures for $i > 1$):

$$p_i^{old} = \sigma(w s_i^{max} + b) \tag{3}$$

The entity library is updated by "soft insertion" (Joulin and Mikolov, 2015) of the current exposure vector $\mathbf{u}_i$ into the library. Concretely, we add the vector to each entity in the library, weighted by the

[5] While we developed DIRE, Henaff et al. (2016) proposed a similar architecture; we leave a comparison to future work.

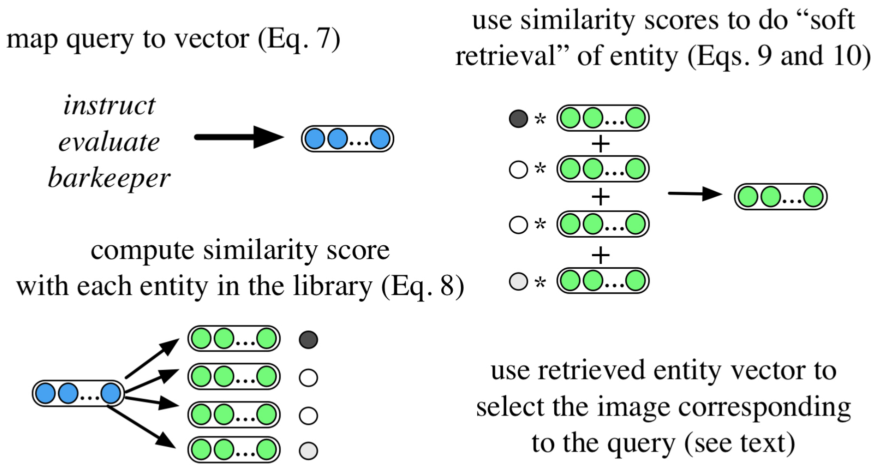
map query to vector (Eq. 7)
instruct
evaluate
barkeeper
use similarity scores to do "soft retrieval" of entity (Eqs. 9 and 10)
compute similarity score with each entity in the library (Eq. 8)
use retrieved entity vector to select the image corresponding to the query (see text)

Figure 3: Querying the DIRE entity library.

probability that the current exposure is an instance of that entity. For the $i-1$ existing entities, this probability is obtained by distributing the p_i^{old} mass across them, according to their probability of being the matching entity, conditional on the exposure being old. The latter probability is estimated by softmax-normalizing the similarity profile $\mathbf{s}_i$ from above. The probability that x_i is new is obviously $1 - p_i^{old}$. This results in the following distribution, where $\|$ stands for concatenation:

$$\mathbf{z}_i = p_i^{old} * \mathrm{softmax}(\mathbf{s}_i)\|(1 - p_i^{old}) \tag{4}$$

Note that z_i has one value more than the current number of stored entities, expressing the probability that the current exposure instantiates a new entity.

The entity library is then updated as:

$$\mathbf{U}_i = \mathbf{z}_i \mathbf{u}_i^\intercal \tag{5}$$

$$\mathbf{E}_i = (\mathbf{E}_{i-1}\|\mathbf{0}) + \mathbf{U}_i \tag{6}$$

Thus, we insert a 0 vector of the same dimensionality as the $\mathbf{u}_i$ vectors at the end of the library, initializing a blank slot to store a new entity. As a consequence, the library in its end state will always contain as many entity vectors as exposures. However, we expect those inserted for exposures of old entities (that is, when $p_i^{old} \approx 1$) to be near zero, and removable from the library along the lines of Graves et al. (2016).

3.2 Cross-modal Entity Tracking with DIRE

We use DIRE for the cross-modal entity tracking task as follows (see Figure 3). Given pre-trained image and verbal attribute representations, we first derive a multimodal representation $\mathbf{u}_i$ for each exposure x_i and update the entity library as explained in Section 3.1. The linguistic query is mapped to the same multimodal space where entities live, and the most relevant entity is retrieved. Finally, the images the model has to choose from (candidate set) are also mapped to multimodal space, and the correct answer is picked based on their similarity with the retrieved entity. We share the same $\mathbf{V}$ projection across all images (in the exposures as well as in the candidate set at query time), a single $\mathbf{A}$ projection for the verbal attributes (in the exposures and in the query), and a matrix $\mathbf{C}$ for the category name in the query. Details on each of the steps follow.

Multimodal Mapping. Exposures are linearly mapped to a multimodal space combining visual and linguistic information, building the $\mathbf{u}_i$ vector by separately embedding each image vector $\mathbf{i}_i$ and attribute vector $\mathbf{a}_i$ using a matrix $\mathbf{V}$ for images[6] and a matrix $\mathbf{A}$ for linguistic attributes[7] and adding up the result (Equation 7). Storage takes place by feeding the $\mathbf{u}_i$ vectors sequentially to the entity library (cf. Section 3.1).

$$\mathbf{u}_i = \mathbf{V}\mathbf{i}_i + \mathbf{A}\mathbf{a}_i \tag{7}$$

Query and Retrieval. To select the best entity match for the query, we compute a "soft retrieval" operation inspired by Sukhbaatar et al. (2015). To query the entity library, we first map the query (a linguistic referring expression, consisting of one noun and two attributes) to multimodal space. We embed the attribute vectors $\mathbf{a}_1^q$, $\mathbf{a}_2^q$ with the matrix $\mathbf{A}$ learned during storage and the noun vector $\mathbf{c}$ with matrix $\mathbf{C}$, and we sum the result (Eq. 8). The query vector $\mathbf{q}$ lives in the same space as the entity vectors, which enables similarity computations.

$$\mathbf{q} = \mathbf{C}\mathbf{c} + \mathbf{A}\mathbf{a}_1^q + \mathbf{A}\mathbf{a}_2^q \tag{8}$$

We then retrieve the entity representation that matches the query, by first computing the similarity of the mapped query $\mathbf{q}$ to each entity vector $\mathbf{e}_i$ through a normalized dot product, $\mathbf{g}$ (Eq. 9), and then using those similarities as weights to perform a "soft retrieval" of the entity that best matches the query, summing up the vectors in the entity library multiplied by $\mathbf{g}$ (Eq. 10). Note that if only one entity is significantly similar to the query (so that the corresponding entry in the similarity profile tends to 1, while all other entries tend to 0), this is equivalent to retrieving that entity.

$$\mathbf{g} = \mathrm{softmax}(\mathbf{E}_n\mathbf{q}) \tag{9}$$

$$\mathbf{r} = \mathbf{E}_n^{\intercal}\mathbf{g} \tag{10}$$

Picking the Right Image. Finally, we use the retrieved entity representation, $\mathbf{r}$, to pick among the k images that represent the entities. We map the candidate image vectors $\mathbf{d}_1 \ldots \mathbf{d}_k$ to multimodal space using the same visual matrix $\mathbf{V}$ as above. We compare the query with each of the images using a dot product, again obtaining a similarity profile, that we softmax-normalize to obtain the final probability distribution that will give us the candidate image, namely, the one corresponding to the argmax of the probability distribution. Note that we need a probability distribution because we use a cross-entropy cost function when training the model.

The whole architecture is differentiable, allowing end-to-end training by gradient descent; in particular, the cross-modal mapping is learned as the model learns to refer.[8] At the same time, it emulates discrete-like operations like insertion and retrieval of entity representations, that, in frameworks such as DRT, are performed entirely in symbolic terms, and are manually coded in the DRT system Boxer (Bos, 2008). This has the advantage that the entity representations can be continuous, enabling their matching with continuous representations of language as well as cross-modal reasoning (for instance, using *cup* for something that a different speaker calls *mug*, or mixing visual and linguistically conveyed information). The model is rather parsimonious, with parameters limited to three mapping matrices ($\mathbf{V}$, $\mathbf{A}$, $\mathbf{C}$) and the bias and weight terms for p^{old}.

4 Experiments

Experimental details. Images are represented by 4096-dimensional vectors produced by passing images through the pre-trained VGG 19-layer CNN of Simonyan and Zisserman (2015) (trained on the ILSVRC-2012 data), and extracting the corresponding activations on the topmost fully connected layer.[9] Linguistic

[6] Size $v \times m$, where v is the size of the image vector and m the multimodal dimensionality.

[7] Size $t \times m$, where t is the size of the attribute vector. Both matrices, V and A, are learned.

[8] Note that the input vectors for images are only visual, and those for nouns and attributes are only textual.

[9] We use the MatConvNet toolkit, `http://www.vlfeat.org/matconvnet`.

Baseline		Standard models		DIRE				MemN			
Random	0.17	FF	0.27	1m	0.64	2m	0.65	1m-1h	0.59	2m-1h	0.67
		RNN	0.28					1m-2h	0.67	2m-2h	**0.69**
								1m-3h	0.30	2m-3h	0.30

Table 1: Tracking results (accuracy on test set).

representations are given by 400-dimensional `cbow` embeddings from Baroni et al. (2014), trained on about 2.8 billion tokens of raw text. We map to a 1K-dimensional multimodal space. The parameters of **DIRE** are estimated by stochastic gradient descent with 0.09 learning rate, 10 minibatch size, 0.5 dropout probability, and maximally 150 epochs (here and below, hyperparameter values as in Baroni et al., 2017).

As competitors, we train standard feed-forward (**FF**) and recurrent (**RNN**) networks which have no external memory, using two 300-dimensional hidden layers and sigmoid nonlinearities. We also implement the related Memory Network model (**MemN**; Sukhbaatar et al., 2015). Like DIRE, MemN controls a memory structure, but stores each input exposure separately in the memory. At the same time, MemN can perform multiple "hops" at query time. Each hop consists in soft-retrieving a vector from the memory, where the probing vector is the sum of the input query vector and the vector retrieved in the previous hop (null for the first hop). Conceptually, DIRE attempts to merge different instances of the same entity at input processing time, whereas MemN stores each piece of input separately and aggregates relevant information at query time. MemN can thus use the query to guide the search for relevant information. At the same time, it does not optimize the way in which it stores information in memory. Another difference with DIRE is that MemN uses *two* sets of mapping matrices: One to derive the vectors used at query time, the other for the vectors used for retrieval. We employ the same hyperparameters for MemN (also multimodal vector size) as for our model.

Results. Table 1 shows that DIRE outperforms the standard networks (FF and RNN) by a large margin, confirming the importance of a discrete memory structure for reference tracking. If we make the MemN architecture completely comparable to our model (with one matrix and one hop, MemN-1m-1h), our model achieves higher results (0.64 for DIRE-1m, 0.59 for MemN-1m-1h), which indicates that the basic architecture of the model holds promise. However, MemN outperforms DIRE when using two matrices, two hops (0.67 MemN-2m-1h/MemN-1m-2h vs. 0.65 DIRE-2m), or both (0.69 MemN-2m-2h). For MemN, this seems to be the upper bound, as increasing to three hops greatly harms results (see last row).

Further analysis suggests that DIRE successfully addresses the two challenges set out in the introduction: (i) It learns to categorize: Only for 8% of the datapoints does the model pick an image of the wrong category, and these are cases where confounders belong to visually similar or related categories to the target (*cottage-chalet*, *youngster-enthusiast*, *witch-potion*). It is worth noting that the model learns to categorize directly from reference acts: At exposure time, the image is not provided with a category label, so the model needs to induce the category as part of solving the reference task. (ii) DIRE also learns to individuate by combining visual and linguistically-conveyed information: The similarity of the exposure to the query goes to near-zero when the attribute is wrong, even when the category is the same. Together, these two properties make it able to ground linguistic expressions to entities represented in images. However, the entity creation mechanism still needs to be fine-tuned, as currently DIRE creates a new entity vector for nearly every exposure. More work is needed for this crucial part of the model.

5 Discussion

Providing a continuous model of reference that can emulate discrete reasoning about entities is an ambitious research programme. We have reported on work in progress on such a model, DIRE, which, unlike Memory Networks, and emulating formal approaches such as DRT within an end-to-end neural architecture, is designed to make decisions as to how to store the information *at input processing time*, in a way that aids further reasoning, namely, organizing it by entity. Results suggest that merging complementary aspects of DIRE and MemN could be fruitful. We have also presented a new task, cross-modal entity tracking, that tests the categorization and individuation capabilities of computational models, and a challenging dataset for the task.

Our project is related to several areas of active research. Reference is a classic topic in philosophy of language and linguistics (Frege, 1892; Abbott, 2010; Kamp and Reyle, 1993; Kamp, 2015); emulating discrete aspects of language and reasoning through continuous means is a long-standing goal in artificial intelligence (Smolensky, 1990; Joulin and Mikolov, 2015), and recent work focuses on reference (Baroni et al., 2017; Herbelot, 2015; Herbelot and Vecchi, 2015); grounding language in perception (Chen and Mooney, 2011; Bruni et al., 2012; Silberer et al., 2013), as well as reference and co-reference (Krahmer and Van Deemter, 2012; Poesio et al., 2017) are important subjects in Computational Linguistics. Our programme puts these different strands together.

Acknowledgments: We thank Angeliki Lazaridou for help producing the visual vectors used in the paper. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 715154; AMORE); EU Horizon 2020 programme under the Marie Skłodowska-Curie grant agreement No 655577 (LOVe); ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES); DFG (SFB 732, Project D10). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs to U. Trento and U. Pompeu Fabra. This paper reflects the authors' view only, and the EU is not responsible for any use that may be made of the information it contains.

References

Abbott, B. (2010). *Reference*. Oxford, UK: Oxford University Press.

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh (2015). VQA: Visual Question Answering. In *Proceedings of ICCV*, Santiago de Chile, Chile.

Baroni, M., G. Boleda, and S. Padó (2017). Show me the cup: Reference with continuous representations. In *Proceedings of CICLing (International Conference on Computational Linguistics and Intelligent Text Processing)*.

Baroni, M., G. Dinu, and G. Kruszewski (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, Baltimore, MD, pp. 238–247.

Baroni, M. and A. Lenci (2010). Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics 36*(4), 673–721.

Barsalou, L. W. (1983). Ad hoc categories. *Memory & Cognition 11*(3), 211–227.

Boleda, G. and A. Herbelot (2016). Formal distributional semantics: Introduction to the special issue. *Computational Linguistics 42*(4), 619–635.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pp. 277–286. Association for Computational Linguistics.

Bruni, E., G. Boleda, M. Baroni, and N. K. Tran (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, pp. 136–145.

Brysbaert, M., A. B. Warriner, and V. Kuperman (2014). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods 46*, 904–911.

Chen, D. and R. Mooney (2011). Learning to interpret natural language navigation instructions from observations. In *Proceedings of AAAI*, San Francisco, CA, pp. 859–865.

Frege, G. (1892). Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik 100*, 25–50.

Graves, A., G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. Gómez Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. Moritz Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis (2016). Hybrid computing using a neural network with dynamic external memory. *Nature 538*(7626), 471–476.

Henaff, M., J. Weston, A. Szlam, A. Bordes, and Y. LeCun (2016). Tracking the World State with Recurrent Neural Networks. `https://arxiv.org/abs/1612.03969`.

Herbelot, A. (2015). Mr Darcy and Mr Toad, gentlemen: distributional names and their kinds. In *Proceedings of the 11th International Conference on Computational Semantics*, London, UK, pp. 151–161. Association for Computational Linguistics.

Herbelot, A. and E. M. Vecchi (2015). Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 22–32. Association for Computational Linguistics.

Joulin, A. and T. Mikolov (2015). Inferring algorithmic patterns with stack-augmented recurrent nets. In *Proceedings of NIPS*, Montreal, Canada.

Kamp, H. (2015). Entity Representations and Articulated Contexts: An Exploration of the Semantics and Pragmatics of Definite Noun Phrases. Ms. University of Stuttgart.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

Krahmer, E. and K. Van Deemter (2012). Computational generation of referring expressions: A survey. *Computational Linguistics 38*(1), 173–218.

Lazaridou, A., N. Pham, and M. Baroni (2015). Combining language and vision with a multimodal skip-gram model. In *Proceedings of NAACL*, Denver, CO, pp. 153–163.

Murphy, G. (2002). *The Big Book of Concepts*. Cambridge, MA: MIT Press.

Poesio, M., R. Stuckardt, and Y. Versley (2017). *Anaphora Resolution: Algorithms, Resources, and Applications*. Springer. In press.

Silberer, C., V. Ferrari, and M. Lapata (2013). Models of semantic representation with visual attributes. In *Proceedings of ACL*, Sofia, Bulgaria, pp. 572–582.

Simonyan, K. and A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of ICLR Conference Track*, San Diego, CA. Published online: `http://www.iclr.cc/doku.php?id=iclr2015:main`.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence 46*, 159–216.

Sukhbaatar, S., A. Szlam, J. Weston, and R. Fergus (2015). End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pp. 2440–2448. Montral, Canada.

Van Deemter, K. (2012). *Not exactly: In praise of vagueness*. Oxford University Press.