Resolving References to Objects in Photographs using the Words-As-Classifiers Model

David Schlangen Sina Zarrieß Casey Kennington
Dialogue Systems Group // CITEC // Faculty of Linguistics and Literary Studies
Bielefeld University, Germany
first.last@uni-bielefeld.de

Abstract

A common use of language is to refer to visually present objects. Modelling it in computers requires modelling the link between language and perception. The "words as classifiers" model of grounded semantics views words as classifiers of perceptual contexts, and composes the meaning of a phrase through composition of the denotations of its component words. It was recently shown to perform well in a game-playing scenario with a small number of object types. We apply it to two large sets of real-world photographs that contain a much larger variety of object types and for which referring expressions are available. Using a pre-trained convolutional neural network to extract image region features, and augmenting these with positional information, we show that the model achieves performance competitive with the state of the art in a reference resolution task (*given expression, find bounding box of its referent*), while, as we argue, being conceptually simpler and more flexible.

1 Introduction

A common use of language is to refer to objects in the shared environment of speaker and addressee. Being able to simulate this is of particular importance for verbal human/robot interfaces (HRI), and the task has consequently received some attention in this field (Matuszek et al., 2012; Tellex et al., 2011; Krishnamurthy and Kollar, 2013).

Here, we study a somewhat simpler precursor task, namely that of resolution of reference to objects in static images (photographs), but use a larger set of object types than is usually done in HRI work ($>$ 300, see below). More formally, the task is to retrieve, given a referring expression e and an image I, the region bb^* of the image that is most likely to contain the referent of the expression. As candidate regions, we use both manually annotated regions as well as automatically computed ones.

As our starting point, we use the "words-as-classifiers" model recently proposed by Kennington and Schlangen (2015). It has before only been tested in a small domain and with specially designed features; here, we apply it to real-world photographs and use learned representations from a convolutional neural network (Szegedy et al., 2015). We learn models for between 400 and 1,200 words, depending on the training data set. As we show, the model performs competitive with the state of the art (Hu et al., 2016; Mao et al., 2016) on the same data sets.

Our background interest in situated interaction makes it important for us that the approach we use is 'dialogue ready'; and it is, in the sense that it supports incremental processing (giving results while the incoming utterance is going on) and incremental learning (being able to improve performance from interactive feedback). However, in this paper we focus purely on 'batch', non-interactive performance.[1]

2 Related Work

The idea of connecting words to what they denote in the real world via perceptual features goes back at least to Harnad (1990), who coined "The Symbol Grounding Problem": "*[H]ow can the semantic interpretation of a formal symbol system be made* intrinsic *to the system, rather than just parasitic on the meanings in our heads?*" The pro-

[1]The code for reproducing the results reported in this paper can be found at `https://github.com/dsg-bielefeld/image_wac`.

posed solution was to link 'categorial representations' with "learned and innate feature detectors that pick out the invariant features of object and event categories from their sensory projections".

This suggestion has variously been taken up in computational work. An early example is Deb Roy's work from the early 2000s (Roy et al., 2002; Roy, 2002; Roy, 2005). In (Roy et al., 2002), computer vision techniques are used to detect object boundaries in a video feed, and to compute colour features (mean colour pixel value), positional features, and features encoding the relative spatial configuration of objects. These features are then associated in a learning process with certain words, resulting in an association of colour features with colour words, spatial features with prepositions, etc., and based on this, these words can be interpreted with reference to the scene currently presented to the video feed.

Of more recent work, that of Matuszek et al. (2012) is closely related to the approach we take. The task in this work is to compute (sets of) referents, given a (depth) image of a scene containing simple geometric shapes and a natural language expression. In keeping with the formal semantics tradition, a layer of logical form representation is assumed; it is not constructed via syntactic parsing rules, however, but by a learned mapping (*semantic parsing*). The non-logical constants of this representation then are interpreted by linking them to classifiers that work on perceptual features (representing shape and colour of objects). Interestingly, both mapping processes are trained jointly, and hence the links between classifiers and non-logical constants on the one hand, and non-logical constants and lexemes on the other are induced from data. In the work presented here, we take a simpler approach that forgoes the level of semantic representation and directly links lexemes and perceptions, but does not yet learn the composition.

Most closely related on the formal side is recent work by Larsson (2015), which offers a very direct implementation of the 'words as classifiers' idea (couched in terms of type theory with records (TTR; (Cooper and Ginzburg, 2015)) and not model-theoretic semantics). In this approach, some lexical entries are enriched with classifiers that can judge, given a representation of an object, how applicable the term is to it. The paper also describes how these classifiers could be trained (or adapted) in interaction. The model is only specified theoretically, however, with hand-crafted classifiers for a small set of words, and not tested with real data.

The second area to mention here is the recently very active one of image-to-text generation, which has been spurred on by the availability of large datasets and competitions structured around them. The task here typically is to generate a description (a caption) for a given image. A frequently taken approach is to use a convolutional neural network (CNN) to map the image to a dense vector (which we do as well, as we will describe below), and then condition a neural language model (typically, an LSTM) on this to produce an output string (Vinyals et al., 2015; Devlin et al., 2015). Fang et al. (2015) modify this approach somewhat, by using what they call "word detectors" first to specifically propose words for image regions, out of which the caption is then generated. This has some similarity to our word models as described below, but again is tailored more towards generation.

Socher et al. (2014) present a more compositional variant of this type of approach where sentence representations are composed along the dependency parse of the sentence. The representation of the root node is then mapped into a multimodal space in which distance between sentence and image representation can be used to guide image retrieval, which is the task in that paper. Our approach, in contrast, composes on the level of denotations and not that of representation.

Two very recent papers carry this type of approach over to the problem of resolving references to objects in images. Both (Hu et al., 2015) and (Mao et al., 2015) use CNNs to encode image information (and interestingly, both combine, in different ways, information from the candidate region with more global information about the image as a whole), on which they condition an LSTM to get a prediction score for fit of candidate region and referring expression. As we will discuss below, our approach has some similarities, but can be seen as being more compositional, as the expression score is more clearly composed out of individual word scores (with rule-driven composition, however). We will directly compare our results to those reported in these papers, as we were able to use the same datasets.

3 The "Words-As-Classifiers" Model

We now briefly review (and slightly reformulate) the model introduced by Kennington and Schlangen (2015). It has several components:

A Model of Word Meanings Let w be a word whose meaning is to be modelled, and let $\mathbf{x}$ be a representation of an object in terms of its visual features. The core ingredient then is a classifier then takes this representation and returns a score $f_w(\mathbf{x})$, indicating the "appropriateness" of the word for denoting the object.

Noting a (loose) correspondence to Montague's (1974) intensional semantics, where the intension of a word is a function from possible worlds to extensions (Gamut, 1991), the *intensional* meaning of w is then defined as the classifier itself, a function from a representation of an object to an "appropriateness score":[2]

$$[\![w]\!]_{obj} = \lambda \mathbf{x}.f_w(\mathbf{x}) \quad (1)$$

(Where $[\![.]\!]$ is a function returning the meaning of its argument, and $\mathbf{x}$ is a feature vecture as given by f_{obj}, the function that computes the representation for a given object.)

The *extension* of a word in a given (here, visual) discourse universe W can then be modelled as a probability distribution ranging over all candidate objects in the given domain, resulting from the application of the word intension to each object ($\mathbf{x}_i$ is the feature vector for object i, $normalize()$ vectorized normalisation, and I a random variable ranging over the k candidates):

$$\begin{aligned}&[\![w]\!]^W_{obj} = \\ &\quad normalize(([\![w]\!]_{obj}(\mathbf{x}_1), \ldots, [\![w]\!]_{obj}(\mathbf{x}_k))) = \\ &\quad normalize((f_w(\mathbf{x}_1), \ldots, f_w(\mathbf{x}_k))) = P(I|w) \end{aligned} \quad (2)$$

Composition Composition of word meanings into phrase meanings in this approach is governed by rules that are tied to syntactic constructions. In the following, we only use simple multiplicative composition for nominal constructions:

$$[\![[_{nom} w_1, \ldots, w_k]]\!]^W = [\![\text{NOM}]\!]^W [\![w_1, \ldots, w_k]\!]^W = \circ_{/N} ([\![w_1]\!]^W, \ldots, [\![w_k]\!]^W) \quad (3)$$

where $\circ_{/N}$ is defined as

$$\begin{aligned}&\circ_{/N} ([\![w_1]\!]^W, \ldots, [\![w_k]\!]^W) = P_\circ(I|w_1, \ldots, w_k) \\ &\quad \text{with } P_\circ(I = i|w_1, \ldots, w_k) = \\ &\quad \frac{1}{Z}(P(I = i|w_1) * \cdots * P(I = i|w_k)) \text{ for } i \in I \end{aligned} \quad (4)$$

(Z takes care that the result is normalized over all candidate objects.)

[2](Larsson, 2015) develops this intension/extension distinction in more detail for his formalisation.

Selection To arrive at the desired extension of a full referring expression—an individual object, in our case—, one additional element is needed, and this is contributed by the determiner. For uniquely referring expressions ("the red cross"), what is required is to pick the most likely candidate from the distribution:

$$[\![the]\!] = \lambda x. \arg\max_{Dom(x)} x \quad (5)$$

$$[\![[the]\ [_{nom} w_1, \ldots, w_k]]\!]^W = \arg\max_{i \in W} [\ [\![[_{nom} w_1, \ldots, w_k]]\!]^W\ ] \quad (6)$$

In other words, the prediction of an expression such as "the brown shirt guy on right" is computed by first getting the responses of the classifiers corresponding to the words, individually for each object. I.e., the classifier for "brown" is applied to objects $o_1, \ldots, o_n$. This yields a vector of responses (of dimensionality n, the number of candidate objects); similarly for all other words. These vectors are then multiplied, and the predicted object is the maximal component of the resulting vector. Figure 1 gives a schematic overview of the model as implemented here, including the feature extraction process.

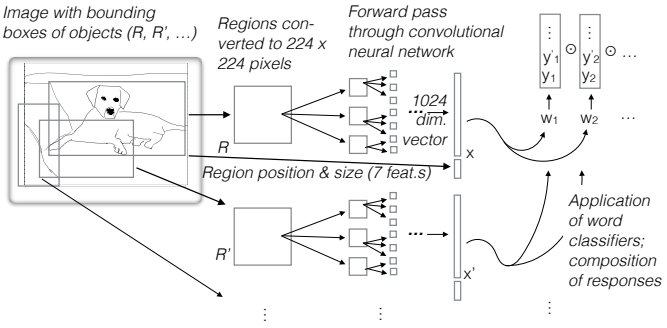

Figure 1: Overview of the model

4 Data: Images & Referring Expressions

SAIAPR TC-12 / ReferItGame The basis of this data set is the IAPR TC-12 image retrieval benchmark collection of "20,000 still natural images taken from locations around the world and comprising an assorted cross-section of still natural images" (Grubinger et al., 2006). A typical example of an image from the collection is shown in Figure 2 on the left.

This dataset was later augmented by Escalante et al. (2010) with segmentation masks identifying objects in the images (an average of 5 objects per image). Figure 2 (middle) gives an example of such a segmentation. These segmentations were

Figure 2: Image 27437 from IAPR TC-12 (left), with region masks from SAIAPR TC-12 (middle); "brown shirt guy on right" is a referring expression in REFERITGAME for the region singled out on the right

done manually and provide close maskings of the objects. This extended dataset is also known as "SAIAPR TC-12" (for "segmented and annotated IAPR TC-12").

The third component is provided by Kazemzadeh et al. (2014), who collected a large number of expressions referring to (pre-segmented) objects from these images, using a crowd-sourcing approach where two players were paired and a director needed to refer to a predetermined object to a matcher, who then selected it. (An example is given in Figure 2 (right).) This corpus contains 120k referring expressions, covering nearly all of the 99.5k regions from SAIAPR TC-12.[3] The average length of a referring expression from this corpus is 3.4 tokens. The 500k token realise 10,340 types, with 5785 hapax legomena. The most frequent tokens (other than articles and prepositions) are "left" and "right", with 22k occurrences. (In the following, we will refer to this corpus as REFERIT.)

This combination of segmented images and referring expressions has recently been used by Hu et al. (2015) for learning to resolve references, as we do here. The authors also tested their method on region proposals computed using the EdgeBox algorithm (Zitnick and Dollár, 2014). They kindly provided us with this region proposal data (100 best proposals per image), and we compare our results on these region proposals with theirs below. The authors split the dataset evenly into 10k images (and their corresponding referring expressions) for training and 10k for testing. As we needed more training data, we made a 90/10 split, ensuring that all our test images are from their test split.

[3]The IAPR TC-12 and SAIAPR TC-12 data is available from `http://imageclef.org`; REFERITGAME from `http://tamaraberg.com/referitgame`.

MSCOCO / GoogleRefExp / ReferItGame The second dataset is based on the "Microsoft Common Objects in Context" collection (Lin et al., 2014), which contains over 300k images with object segmentations (of objects from 80 pre-specified categories), object labels, and image captions. Figure 3 shows some examples of images containing objects of type "person".

This dataset was augmented by Mao et al. (2015) with what they call 'unambiguous object descriptions', using a subset of 27k images that contained between 2 and 4 instances of the same object type within the same image. The authors collected and validated 100k descriptions in a crowd-sourced approach as well, but unlike in the ReferItGame setup, describers and validators were not connected live in a game setting.[4] The average length of the descriptions is 8.3 token. The 790k token in the corpus realise 14k types, with 6950 hapax legomena. The most frequent tokens other than articles and prepositions are "man" (15k occurrences) and "white" (12k). (In the following, we will refer to this corpus as GREXP.)

The authors also computed automatic region proposals for these images, using the multibox method of Erhan et al. (2014) and classifying those using a model trained on MSCOCO categories, retaining on average only 8 per image. These region proposals are on average of a much higher quality than those we have available for the other dataset.

As mentioned in (Mao et al., 2015), Tamara Berg and colleagues have at the same time used their ReferItGame paradigm to collect referring expressions for MSCOCO images as well. Upon request, Berg and colleagues also kindly provided us with this data—140k referring expressions, for 20k images, average length 3.5 token, 500k token altogether, 10.3k types, 5785 hapax legomena; most frequent also "left" (33k occurrences)

[4]The data is available from `https://github.com/mjhucla/Google_Refexp_toolbox`.

REFCOCO:
green woman
GREXP:
a woman wearing blue jeans

REFCOCO:
person left
GREXP:
a boy is ready to play who is wearing green color pant

Figure 3: Examples from MSCOCO

and "right" (32k). (In the following, we will call this corpus REFCOCO.) In our experiments, we use the training/validation/test splits on the images suggested by Berg et al., as the splits provided by Mao et al. (2015) are on the level of objects and have some overlap in images.

It is interesting to note the differences in the expressions from REFCOCO and GREXP, the latter on average being almost 5 token longer. Figure 3 gives representative examples. We can speculate that the different task descriptions ("refer to this object" vs. "produce an unambiguous description") and the different settings (live to a partner vs. offline, only validated later) may have caused this. As we will see below, the GREXP descriptions did indeed cause more problems to our approach, which is meant for reference in interaction.

5 Training the Word/Object Classifiers

The basis of the approach we use are the classifiers that link words and images. These need to be trained from data; more specifically, from pairings of image regions and referring expressions, as provided by the corpora described in the previous section.

Representing Image Regions The first step is to represent the information from the image regions. We use a deep convolutional neural network, "GoogLeNet" (Szegedy et al., 2015), that was trained on data from the Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) from the ImageNet corpus (Deng et al., 2009) to extract features.[5] It was optimised to recognise categories from that challenge, which are different from those occurring in either SAIAPR or COCO, but in any case we only use the final fully-connected layer before the classification layer, to give us a 1024 dimensional representation of the region. We augment this with 7 features that encode information about the region relative to the image: the (relative) coordinates of two corners, its (relative) area, distance to the center, and orientation of the image. The full representation hence is a vector of 1031 features. (See also Figure 1 above.)

Selecting candidate words How do we select the words for which we train perceptual classifiers? There is a technical consideration to be made here and a semantic one. The technical consideration is that we need sufficient training data for the classifiers, and so can only practically train classifiers for words that occur often enough in the training corpus. We set a threshold here of a minimum of 40 occurences in the training corpus, determined empirically on the validation set to provide a good tradeoff between vocabulary coverage and number of training instances.

The semantic consideration is that intuitively, the approach does not seem appropriate for all types of words; where it might make sense for attributes and category names to be modelled as image classifiers, it does less so for prepositions and other function words. Nevertheless, for now, we make the assumption that all words in a referring expression contribute information to the *visual* identification of its referent. We discuss the consequences of this decision below.

This assumption is violated in a different way in phrases that refer via a landmark, such as in "the thing next to the woman with the blue shirt". Here we cannot assume for example that the referent region provides a good instance of "blue" (since it is not the target object in the region that is described as blue), and so we exclude such phrases from the training set (by looking for a small set of expressions such as "left of", "behind", etc.; see appendix for a full list). This reduces the train-

[5] http://www.image-net.org/challenges/LSVRC/2014/.
We use the sklearn-theano (http://sklearn-theano.github.io/feature_extraction/index.html#feature-extraction) port of the Caffe replication and re-training (https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet) of this network structure.

ing portions of REFERIT, REFCOCO and GREXP to 86%, 95%, and 82% of their original size, respectively (counting referring expressions, not tokens).

Now that we have decided on the set of words for which to train classifiers, how do we assemble the training data?

Positive Instances Getting positive instances from the corpus is straightforward: We pair each word in a referring expression with the representation of the region it refers to. That is, if the word "left" occurs 20,000 times in expressions in the training corpus, we have 20,000 positive instances for training its classifier.

Negative Instances Acquiring negative instances is less straightforward. The corpus does not record inappropriate uses of a word, or 'negative referring expressions' (as in "this is not a red chair"). To create negative instances, we make a second assumption which again is not generally correct, namely that when a word was *never* in the corpus used to refer to an object, this object can serve as a negative example for that word/object classifier. In the experiments reported below, we randomly selected 5 image regions from the training corpus whose referring expressions (if there were any) did not contain the word in question.[6]

The classifiers Following this regime, we train binary logistic regression classifiers (with $\ell 1$ regularisation) on the visual object features representations, for all words that occurred at least 40 times in the respective training corpus.[7]

To summarise, we train separate binary classifiers for each word (not making any a-priori distinction between function words and others, or attribute labels and category labels), giving them the task to predict how likely it would be that the word they represent would be used to refer to the image region they are given. All classifiers are presented during training with data sets with the same balance of positive and negative examples (here, a fixed ratio of 1 positive to 5 negative). Hence, the classifiers themselves do not reflect any word frequency effects; our claim (to be validated in future work) is that any potential effects of this type are better modelled separately.

[6]This approach is inspired by the negative sampling technique of Mikolov et al. (2013) for training textual word embeddings.

[7]We used the implementation in the `scikit learn` package (Pedregosa et al., 2011).

	%tst	acc	mrr	arc	>0	acc
REFERIT	1.00	0.65	0.79	0.89	0.97	0.67
REFERIT; NR	0.86	0.68	0.82	0.91	0.97	**0.71**
(Hu et al., 2015)	–	0.73	–	–	–	–
REFCOCO	1.00	0.61	0.77	0.91	0.98	0.62
REFCOCO; NR	0.94	0.63	0.78	0.92	0.98	**0.64**
(Mao et al., 2015)	–	0.70	–	–	–	–
GREXP	1.00	0.43	0.65	0.86	1.00	0.43
GREXP; NR	0.82	0.45	0.67	0.88	1.00	**0.45**
(Mao et al., 2015)	–	0.61	–	–	–	–

Table 1: Results; separately by corpus. See text for description of columns and rows.

6 Experiments

The task in our experiments is the following: Given an image I together with bounding boxes of regions $(bb_1, \ldots, bb_n)$ within it, and a referring expression e, predict which of these regions contains the referent of the expression.

By Corpus We start with training and testing models for all three corpora (REFERIT, REFCOCO, GREXP) separately. But first, we establish some baselines. The first is just randomly picking one of the candidate regions. The second is a 1-rule classifier that picks the largest region. The respective accuracies on the corpora are as follows: REFERIT 0.20/0.19; REFCOCO 0.16/0.23; GREXP 0.19/0.20.

Training on the training sets of REFERIT, REFCOCO and GREX with the regime described above (min. 40 occurrences) gives us classifiers for 429, 503, and 682 words, respectively. Table 1 shows the evaluation on the respective test parts: accuracy (*acc*), mean reciprocal rank (*mrr*) and for how much of the expression, on average, a word classifier is present (*arc*). '>0' shows how much of the testcorpus is left if expressions are filtered out for which not even a single word is the model (which we evaluate by default as false), and accuracy for that reduced set. The 'NR' rows give the same numbers for reduced test sets in which all relational expressions have been removed; '%tst' shows how much of a reduction that is relative to the full testset. The rows with the citations give the best reported results from the literature.[8]

As this shows, in most cases we come close, but do not quite reach these results. The distance is the biggest for GREXP with its much longer expressions. As discussed above, not only are the descriptions longer on average in this corpus, the

[8]Using a different split than (Mao et al., 2015), as their train/test set overlaps on the level of images.

vocabulary size is also much higher. Many of the descriptions contain action descriptions ("the man smiling at the woman"), which do not seem to be as helpful to our model. Overall, the expressions in this corpus do appear to be more like 'mini-captions' describing the region rather than referring expressions that efficiently single it out among the set of distractors; our model tries to capture the latter.

Combining Corpora A nice effect of our setup is that we can freely mix the corpora for training, as image regions are represented in the same way regardless of source corpus, and we can combine occurrences of a word across corpora. We tested combining the testsets of REFERIT and REFCOCO (RI+RC in the Table below), REFCOCO and GREXP (RC+GR), and all three (REFERIT, REFCOCO, and GREXP; RI+RC+GR), yielding models for 793, 933, 1215 words, respectively (with the same "min. 40 occurrences" criterion). For all testsets, the results were at least stable compared to Table 1, for some they improved. For reasons of space, we only show the improvements here.

	%tst	acc	mrr	arc	>0	acc
RI+RC/RC	1.00	**0.63**	0.78	0.92	0.98	**0.64**
RI+RC/RC; NR	0.94	**0.65**	0.79	0.93	0.98	**0.66**
RI+RC+GR/RC	1.00	**0.63**	0.78	0.94	0.99	**0.64**
RI+RC+GR/RC; NR	0.94	**0.65**	0.79	0.95	0.99	**0.66**
RI+RC+GR/GR	1.00	**0.47**	0.68	0.90	1.00	**0.47**
RI+RC+GR/GR; NR	0.82	**0.49**	0.70	0.91	1.00	**0.49**

Table 2: Results, combined corpora

Computed Region Proposals Here, we cannot expect the system to retrieve exactly the ground truth bounding box, since we cannot expect the set of automatically computed regions to contain it. We follow Mao et al. (2015) in using *intersection over union* (IoU) as metric (the size of the intersective area between candidate and ground truth bounding box normalised by the size of the union) and taking an IoU ≥ 0.5 of the top candidate as a threshold for success (P@1). As a more relaxed metric, we also count for the SAIAPR proposals (of which there are 100 per image) as success when at least one among the top 10 candidates exceeds this IoU threshold (R@10). (For MSCOCO, there are only slightly above 5 proposals per image on average, so computing this more relaxed measure does not make sense.) The random baseline (RND) is computed by applying the P@1 criterion to a randomly picked region proposal. (That it is higher than 1/#regions for SAIAPR shows that the regions cluster around objects.)

	RP@1	RP@10	rnd
REFERIT	0.09	0.24	0.03
REFERIT; NR	**0.10**	0.26	0.03
(Hu et al., 2015)	0.18	0.45	
REFCOCO	0.52	–	0.17
REFCOCO; NR	**0.54**	–	0.17
(Mao et al., 2015)	0.52		
GREXP	0.36	–	0.16
GREXP; NR	**0.37**	–	0.17
(Mao et al., 2015)	0.45		

Table 3: Results on region proposals

With the higher quality proposals provided for the MSCOCO data, and the shorter, more prototypical referring expressions from REFCOCO, we narrowly beat the reported results. (Again, note that we use a different split that ensures separation on the level of images between training and test.) (Hu et al., 2015) performs relatively better on the region proposals (the gap is wider), on GREXP, we come relatively closer using these proposals. We can speculate that using automatically computed boxes of a lower selectivity (REFERIT) shifts the balance between needing to actually recognise the image and getting information from the shape and position of the box (our *positional* features; see Section 5).

Ablation Experiments To get an idea about what the classifiers actually pick up on, we trained variants given only the positional features (POS columns below in Table 4) and only object features (NOPOS columns). We also applied a variant of the model with only the top 20 classifiers (in terms of number of positive training examples; TOP20). We only show accuracy here, and repeat the relevant numbers from Table 1 for comparison (FULL).

	nopos	pos	full	top20
RI	0.53	0.60	0.65	0.46
RI; NR	0.56	0.62	0.68	0.48
RC	0.44	0.55	0.61	0.52
RC; NR	0.45	0.57	0.63	0.53

Table 4: Results with reduced models

This table shows an interesting pattern. To a large extent, the object image features and the positional features seem to carry redundant information, with the latter on their own performing better than the former on their own. The full model, however, still gains something from the combination

of the feature types. The top-20 classifiers (and consequently, top 20 most frequent words) alone reach decent performance (the numbers are shown for the full test set here; if reduced to only utterances where at least one word is known, the numbers rise, but the reduction of the testset is much more severe than for the full models with much larger vocabulary).

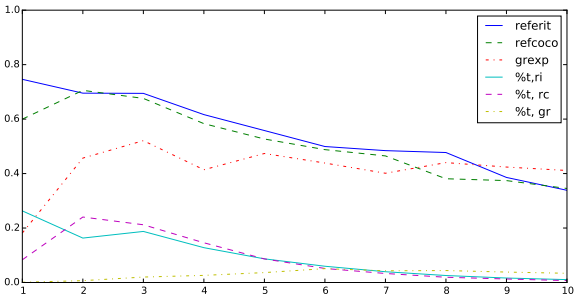

Figure 4: Accuracy by expression length (top 3 lines); percentage of expressions with this length (lower 3 lines).

Error Analysis Figure 4 shows the accuracy of the model split by length of the referring expression (top lines; lower lines show the proportion of expression of this length in the whole corpus). The pattern is similar for all corpora (but less pronounced for GREXP): shorter utterances fare better.

Manual inspection of the errors made by the system further corroborates the suspicion that composition as done here neglects too much of the internal structure of the expression. An example from REFERIT where we get a wrong prediction is "second person from left". The model clearly does not have a notion of counting, and here it wrongly selects the leftmost person. In a similar vein, we gave results above for a testset where spatial relations where removed, but other forms of relation (e.g., "child sitting on womans lap") that weren't modelled still remain in the corpus.

We see as an advantage of the model that we can inspect words individually. Given the performance of short utterances, we can conclude that the word/object classifiers themselves perform reasonably well. This seems to be somewhat independent of the number of training examples they received. Figure 5 shows, for REFERIT, # training instances (x-axis) vs. average accuracy on the validation set, for the whole vocabulary. As this shows, the classifiers tend to get better with more training instances, but there are good ones even with very little training material.

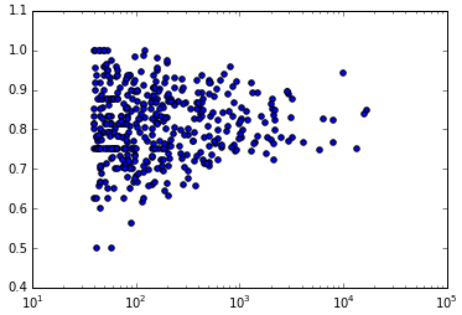

Figure 5: Average accuracy vs. # train instanc.

Mean average precision (i.e., area under the precision / recall curve) over all classifiers (exemplarily computed for the RI+RC set, 793 words) is 0.73 (std 0.15). Interestingly, the 155 classifiers in the top range (average precision over 0.85) are almost all for concrete nouns; the 128 worst performing ones (below 0.60) are mostly other parts of speech. (See appendix.) This is, to a degree, as expected: our assumption behind training classifiers for *all* ocurring words and not pre-filtering based on their part-of-speech or prior hypotheses about visual relevance was that words that can occur in all kinds of visual contexts will lead to classifiers whose contributions cancel out across all candidate objects in a scene.

However, the mean average precision of the classifiers for colour words is also relatively low at 0.6 (std 0.08), for positional words ("left", "right", "center", etc.) it is 0.54 (std 0.1). This might suggest that the features we take from the CNN might indeed be more appropriate for tasks close to what they were originally trained on, namely category and not attribute prediction. We will explore this in future work.

7 Conclusions

We have shown that the "words-as-classifiers" model scales up to a larger set of object types with a much larger variety in appearance (SAIAPR and MSCOCO); to a larger vocabulary and much less restricted expressions (REFERIT, REFCOCO, GREXP); and to use of automatically learned feature types (from a CNN). It achieves results that are comparable to those of more complex models.

We see as advantage that the model we use is "transparent" and modular. Its basis, the word/object classifiers, ties in more directly with

more standard approaches to semantic analysis and composition. Here, we have disregarded much of the internal structure of the expressions. But there is a clear path for bringing it back in, by defining other composition types for other construction types and different word models for other word types. Kennington and Schlangen (2015) do this for spatial relations in their simpler domain; for our domain, new and more richly annotated data such as VISUALgenome looks promising for learning a wide variety of relations.[9] The use of denotations / extensions might make possible transfer of methods from extensional semantics, e.g. for the addition of operators such as negation or generalised quantifiers. The design of the model, as mentioned in the introduction, makes it amenable for use in interactive systems that learn; we are currently exploring this avenue. Lastly, the word/object classifiers also show promise in the reverse task, generation of referring expressions (Zarrieß and Schlangen, 2016).

All this is future work. In its current state—besides, we believe, strongly motivating this future work—, we hope that the model can also serve as a strong baseline to other future approaches to reference resolution, as it is conceptually simple and easy to implement.

Acknowledgments

We thank Hu et al. (2016), Mao et al. (2016) and Tamara Berg for giving us access to their data. Thanks are also due to the anonymous reviewers for their very insightful comments. We acknowledge support by the Cluster of Excellence "Cognitive Interaction Technology" (CITEC; EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG), and by the DUEL project, also funded by DFG (grant SCHL 845/5-1).

References

Robin Cooper and Jonathan Ginzburg. 2015. Type theory with records for natural language semantics. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantic Theory 2nd edition*. Wiley-Blackwell.

Jia Deng, W. Dong, Richard Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language models for image captioning: The quirks and what works. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 100–105, Beijing, China, July. Association for Computational Linguistics.

Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. 2014. Scalable Object Detection Using Deep Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2155–2162.

Hugo Jair Escalante, Carlos a. Hernández, Jesus a. Gonzalez, a. López-López, Manuel Montes, Eduardo F. Morales, L. Enrique Sucar, Luis Villaseñor, and Michael Grubinger. 2010. The segmented and annotated IAPR TC-12 benchmark. *Computer Vision and Image Understanding*, 114(4):419–428.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back. In *Proceedings of CVPR*, Boston, MA, USA, June. IEEE.

L. T. F. Gamut. 1991. *Logic, Language and Meaning: Intensional Logic and Logical Grammar*, volume 2. Chicago University Press, Chicago.

Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. 2006. The IAPR TC-12 benchmark: a new evaluation resource for visual information systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, pages 13–23, Genoa, Italy.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D*, 42:335–346.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2015. Natural language object retrieval. *ArXiv / CoRR*, abs/1511.04164.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of CVPR 2016*, Las Vegas, USA, June.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 787–798, Doha, Qatar.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental

[9] http://visualgenome.org/

reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301, Beijing, China, July. Association for Computational Linguistics.

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206.

Staffan Larsson. 2015. Formal semantics for perceptual classification. *Journal of logic and computation*, 25(2):335–369.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C.Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision ECCV 2014*, volume 8693, pages 740–755. Springer International Publishing.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2015. Generation and comprehension of unambiguous object descriptions. *ArXiv / CoRR*, abs/1511.02283.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of CVPR 2016*, Las Vegas, USA, June.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proceedings of the International Conference on Machine Learning (ICML 2012)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013 (NIPS 2013)*, pages 3111–3119, Lake Tahoe, Nevada, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Deb Roy, Peter Gorniak, Niloy Mukherjee, and Josh Juster. 2002. A trainable spoken language understanding system for visual object selection. In *Proceedings of the International Conference on Speech and Language Processing 2002 (ICSLP 2002)*, Colorado, USA.

Deb K. Roy. 2002. Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3).

Deb Roy. 2005. Grounding words in perception and action: Computational insights. *Trends in Cognitive Sciene*, 9(8):389–396.

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 ACL (TACL)*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR 2015*, Boston, MA, USA, June.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *AAAI Conference on Artificial Intelligence*, pages 1507–1514.

Richmond H. Thomason, editor. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven and London.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition*.

Sina Zarrieß and David Schlangen. 2016. Easy things first: Installments improve referring expression generation for objects in photographs. In *Proceedings of ACL 2016*, Berlin, Germany, August.

C Lawrence Zitnick and Piotr Dollár. 2014. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer.

A Supplemental Material

Filtering relational expressions As described above, we filter out all referring expressions during training that contain either of the following tokens:

```
RELWORDS = ['below',
            'above',
            'between',
            'not',
            'behind',
            'under',
            'underneath',
            'front of',
            'right of',
            'left of',
            'ontop of',
            'next to',
            'middle of']
```

Average Precision See Section 6.
Classifiers with average precision over 0.85:

```
['giraffe', 'coffee', 'court', 'riding', 'penguin', 'balloon', 'ball',
'mug', 'turtle', 'tennis', 'beer', 'seal', 'cow', 'bird', 'horse',
'drink', 'koala', 'sheep', 'ceiling', 'parrot', 'bike', 'cactus',
'sun', 'smoke', 'llama', 'fruit', 'ruins', 'waterfall', 'nightstand',
'books', 'night', 'coke', 'skirt', 'leaf', 'wheel', 'label', 'pot',
'animals', 'cup', 'tablecloth', 'pillar', 'flag', 'field', 'monkey',
'bowl', 'curtain', 'plate', 'van', 'surfboard', 'bottle', 'fish',
'umbrella', 'bus', 'shirtless', 'train', 'bed', 'painting', 'lamp',
'metal', 'paper', 'sky', 'luggage', 'player', 'face', 'going', 'desk',
'ship', 'raft', 'lying', 'vehicle', 'trunk', 'couch', 'palm', 'dress',
'doors', 'fountain', 'column', 'cars', 'flowers', 'tire', 'plane',
'against', 'bunch', 'car', 'shelf', 'bunk', 'boat', 'dog', 'vase',
'animal', 'pack', 'anyone', 'clock', 'glass', 'tile', 'window',
'chair', 'phone', 'across', 'cake', 'branches', 'bicycle', 'snow',
'windows', 'book', 'curtains', 'bear', 'guitar', 'dish', 'both',
'tower', 'truck', 'bridge', 'creepy', 'cloud', 'suit', 'stool', 'tv',
'flower', 'seat', 'buildings', 'shoes', 'bread', 'hut', 'donkey',
'had', 'were', 'fire', 'food', 'turned', 'mountains', 'city', 'range',
'inside', 'carpet', 'beach', 'walls', 'ice', 'crowd', 'mirror',
'brush', 'road', 'anything', 'blanket', 'clouds', 'island',
'building', 'door', '4th', 'stripes', 'bottles', 'cross', 'gold',
'smiling', 'pillow']
```

Classifiers with average precision below 0.6:

```
['shadow', "woman's", 'was', 'bright', 'lol', 'blue', 'her', 'yes',
'blk', 'this', 'from', 'almost', 'colored', 'looking', 'lighter',
'far', 'foreground', 'yellow', 'looks', 'very', 'second', 'its',
'dat', 'stack', 'dudes', 'men', 'him', 'arm', 'smaller', 'half',
'piece', 'out', 'item', 'line', 'stuff', 'he', 'spot', 'green',
'head', 'see', 'be', 'black', 'think', 'leg', 'way', 'women',
'furthest', 'rt', 'most', 'big', 'grey', 'only', 'like', 'corner',
'picture', 'shoulder', 'no', 'spiders', 'n', 'has', 'his', 'we',
'bit', 'spider', 'guys', '2', 'portion', 'are', 'section', 'us',
'towards', 'sorry', 'where', 'small', 'gray', 'image', 'but',
'something', 'center', 'i', 'closest', 'first', 'middle', 'those',
'edge', 'there', 'or', 'white', '-', 'little', 'them', 'barely',
'brown', 'all', 'mid', 'is', 'thing', 'dark', 'by', 'back', 'with',
'other', 'near', 'two', 'screen', 'so', 'front', 'you', 'photo', 'up',
'one', 'it', 'space', 'okay', 'side', 'click', 'part', 'pic', 'at',
'that', 'area', 'directly', 'in', 'on', 'and', 'to', 'just', 'of']
```

Code The code required for reproducing the results reported here can be found at `https://github.com/dsg-bielefeld/image_wac`.