Real-Time Understanding of Complex Discriminative Scene Descriptions

Ramesh Manuvinakurike[1], Casey Kennington[3*], David DeVault[1] and David Schlangen[2]
[1]USC Institute for Creative Technologies / Los Angeles, USA
[2]DSG / CITEC / Bielefeld University / Bielefeld, Germany
[3]Boise State University / Boise, USA
[1]last@ict.usc.edu, [2]first.last@uni-bielefeld.de
[3]first.last@cs.boisestate.edu

Abstract

Real-world scenes typically have complex structure, and utterances about them consequently do as well. We devise and evaluate a model that processes descriptions of complex configurations of geometric shapes and can identify the described scenes among a set of candidates, including similar distractors. The model works with raw images of scenes, and by design can work word-by-word incrementally. Hence, it can be used in highly-responsive interactive and situated settings. Using a corpus of descriptions from game-play between human subjects (who found this to be a challenging task), we show that reconstruction of description structure in our system contributes to task success and supports the performance of the word-based model of grounded semantics that we use.

1 Introduction

In this paper, we present and evaluate a language processing pipeline that enables an automated system to detect and understand complex referential language about visual objects depicted on a screen. This is an important practical capability for present and future interactive spoken dialogue systems. There is a trend toward increasing deployment of spoken dialogue systems for smartphones, tablets, automobiles, TVs, and other settings where information and options are presented on-screen along with an interactive speech channel in which visual items can be discussed (Celikyilmaz et al., 2014). Similarly, for future systems such as smartphones, quadcopters, or self-driving cars that are equipped with cameras, users may wish to discuss objects visible to the system in camera images or video streams.

A challenge in enabling such capabilities for a broad range of applications is that human speakers draw on a diverse set of perceptual and language skills to communicate about objects in situated visual contexts. Consider the example in Figure 1, drawn from the corpus of RDG-Pento games (discussed further in Section 2). In this example, a human in the *director* role describes the visual scene highlighted in red (the *target image*) to another human in the *matcher role*. The scene description is provided in one continuous stream of speech, but it includes three functional segments each providing different referential information: [*this one is kind of a uh a blue T*] [*and a wooden w sort of*] [*the T is kind of malformed*]. The first and third of these three segments refer to the object at the top left of the target image, while the middle segment refers to the object at bottom right. An ability to detect the individual segments of language that carry information about individual referents is an important part of deciphering a scene description like this. Beyond detection, actually understanding these referential segments in context seems to require perceptual knowledge of vocabulary for colors, shapes, materials and hedged descriptions like *kind of a blue T*. In other game scenarios, it's important to understand plural references like *two brown crosses* and relational expressions like *this one has the L on top of the T*.

A variety of vocabulary knowledge is needed, as different speakers may describe individual objects in very different ways (the object described as *kind of a blue T* may also be called *a blue odd-shaped piece* or *a facebook*). When many scenes are described by the same pair of speakers, the pair tends to entrain or align to each other's vocabulary (Garrod and Anderson, 1987), for example by settling on *facebook* as a shorthand description for this ob-

[*] The work was done while at Bielefeld University.

Proceedings of the SIGDIAL 2016 Conference, pages 232–241,
Los Angeles, USA, 13-15 September 2016.

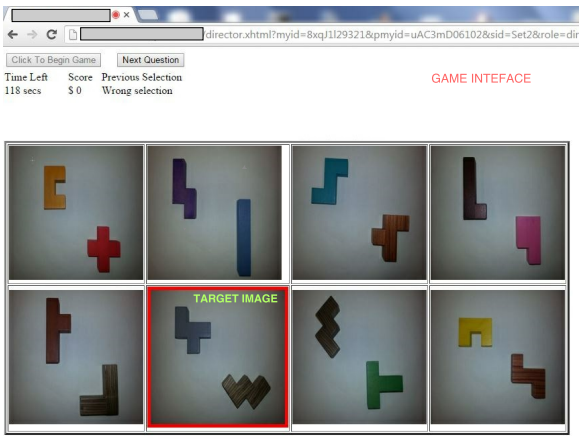
ject type. Finally, to understand a full scene description, the matcher needs to combine all the evidence from multiple referential segments involving a group of objects to identify the target image.

In this paper, we define and evaluate a language processing pipeline that allows many of these perceptual and language skills to be integrated into an automated system for understanding complex scene descriptions. We take the challenging visual reference game RDG-Pento, shown in Figure 1, as our testbed, and we evaluate both human-human and automated system performance in a corpus study. No prior work we are aware of has put forth techniques for grounded understanding of the kinds of noisy, complex, spoken descriptions of visual scenes that can occur in such interactive dialogue settings. This work describes and evaluates an initial approach to this complex problem, and it demonstrates the critical importance of segmentation and entrainment to achieving strong understanding performance. This approach extends the prior work (Kennington and Schlangen, 2015; Han et al., 2015) that assumed either that referential language from users has been pre-segmented, or that visual scenes are given not as raw images but as clean semantic representations, or that visual scenes are simple enough to be described with a one-off referring expression or caption. Our work makes none of these assumptions.

Our automated pipeline, discussed in Section 3, includes components for learning perceptually grounded word meanings, segmenting a stream of speech, identifying the type of referential language in each speech segment, resolving the references in each type of segment, and aggregating evidence across segments to select the most likely target image. Our technical approach enables all of these components to be trained in a supervised manner from annotated, in-domain, human-human reference data. Our quantitative evaluation, presented in Section 4, looks at the performance of the individual components as well as the overall pipeline, and quantifies the strong importance of segmentation, segment type identification, and speaker-specific vocabulary entrainment for improving performance in this task.

2 The RDG-Pento Game

The RDG-Pento (Rapid Dialogue Game-Pentomino) game is a two player collaborative game. RDG-Pento is a variant of the RDG-Image

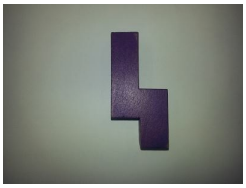


Director: this one is kind of a uh a blue T and a wooden w sort of the T is kind of malformed
Matcher: okay got it

Figure 1: In the game, the director is describing the image highlighted in red (the *target image*) to the matcher, who tries to identify this image from among the 8 possible images. The figure shows the game interface as seen by the director including a transcript of the director's speech.

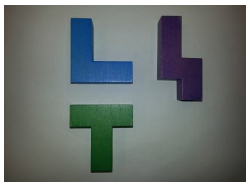
game described by Manuvinakurike and DeVault (2015). As in RDG-Image, both players see 8 images on their screen in a 2X4 grid as shown in Figure 1. One person is assigned the role of director and the other person that of matcher. The director's screen has a single *target image* (TI) highlighted with a red border. The goal of the director is to uniquely describe the TI for the matcher to identify among the distractor images. The 8 images are shown in a different order on the director and matcher screens, so that the TI cannot be identified by grid position. The players can speak freely until the matcher makes a selection. Once the matcher indicates a selection, the director can advance the game. Over time, the gameplay gets progressively more challenging as the images tend to contain more objects that are similar in shape and color. The task is complex by design.

In RDG-Pento, the individual images are taken from a real-world, tabletop scene containing an arrangement of between one and six physical Pentomino objects. Individual images with varying numbers of objects are illustrated in Figure 2. The 8 images at any one time always contain the same number of objects; the number of objects increases as the game progresses. Players play for 5 rounds,



DIRECTOR: its like a harry potter and its like maroon

Segmented & labeled output:
[SIN its like a harry potter and its like maroon]



DIRECTOR: this has the L on top of the T and then the harry potter sign next to the L

Segmented & labeled output:
[SIN this has the L] [REL on top of] [SIN the T] [SIN and then the harry potter sign] [REL next to] [SIN the L]

DIRECTOR: this one is plus sign stairs T plus sign T stairs

Segmented and labeled output:
[SIN this one is plus sign] [SIN stairs] [SIN T] [SIN plus sign] [SIN T] [SIN stairs]

Figure 2: Example scene descriptions for three TIs

alternating roles. Each round has a time limit (about 200 seconds) that creates time pressure for the players, and the time remaining ticks down in a countdown timer.

Data Set The corpus used here was collected using a web framework for crowd-sourced data collection called Pair Me Up (PMU) (Manuvinakurike and DeVault, 2015). To create this corpus, 42 pairs of native English-speakers located in the U.S. and Canada were recruited using AMT. Game play and audio data were captured for each pair of speakers (who were not colocated and communicated entirely through their web browsers), and the resulting audio data was transcribed and annotated. 16 pairs completed all 5 game rounds, while the remaining crowd-sourced pairs completed only part of the game for various reasons. As our focus is on understanding individual scene descriptions, our data set here includes data from the 16 complete games as well as partial games. A more complete description and analysis of the corpus can be found in Zarrieß et al. (2016).

Data Annotation We annotated the transcribed director and matcher speech through a process of segmentation, segment type labeling, and referent identification. The segment types are shown in Table 1, and example annotations are provided in Figure 2. The annotation is carried out on each *target image subdialogue* in which the director and matcher discuss an individual target image. The segmentation and labeling steps create a complete partition of each speaker's speech into sequences of words with a related semantic function in our framework.[1]

Segment type	Label	Examples
Singular	SIN	this is a green t, plus sign
Multiple objects	MUL	two Zs at top, they're all green
Relation	REL	above, in a diagonal
Others	OT	that was tough, lets start

Table 1: Segment types, labels, and examples

Sequences of words that ascribe properties to a single object are joined under the SIN label. Our SIN segment type is not a simple syntactic concept like "singular NP referring expression". The SIN type includes not only simple singular NPs like *the blue s* but also clauses like *it's the blue s* and conjoined clauses like *it's like a harry potter and it's like maroon* (Figure 1). The individuation criterion for SIN is that a SIN segment must ascribe properties only to a single object; as such it may contain word sequences of various syntactic types.

Sequences of words such as *the two crosses* that ascribe properties to multiple objects are joined into a segment under the MUL label.

Sequences of words that describe a geometric relation between objects are segmented and given a REL label. These are generally prepositional expressions, and include both single-word prepositions (*underneath*, *below*) and multi-word complex prepositions (Quirk et al., 1985) which include multiple orthographic words ("next to", "left of" etc.). The REL segments generally describe geometric relations between objects referred to in SIN and MUL segments. An example would be *[MUL two crosses] [REL above] [MUL two Ts]*.

All other word sequences are assigned the type Others and given an OT label. This segment type includes acknowledgments, confirmations, feedback, and laughter, among other dialogue act types not addressed in this work.

For each segment of type SIN, MUL, or REL, the correct referent object or objects within the target image are also annotated.

In the data set, there are a total of 4132 *target*

[1]The annotation scheme was developed iteratively while keeping the reference resolution task and the WAC model (see Section 3.3.1) in mind. The annotation was done by an expert annotator.

image speaker transcripts in which either the director or the matcher's transcribed speech for a target image is annotated. There are 8030 annotated segments (5451 director segments and 2579 matcher segments). There are 1372 word types and 55,238 word tokens.

3 Language Processing Pipeline

In this section, we present our language processing pipeline for segmentation and understanding of complex scene descriptions. The modules, decision-making, and information flow for the pipeline are visualized in Figure 3. The pipeline modules include a Segmenter (Section 3.1), a Segment Type Classifier (Section 3.2), and a Reference Resolver (Section 3.3).

In this paper, we focus on how our pipeline could be used to automate the role of the matcher in the RDG-Pento game. We consider the task of selecting the correct target image based on a human director's transcribed speech drawn from our RDG-Pento corpus. The pipeline is designed however for eventual real-time operation using incremental ASR results, so that in the future it can be incorporated into a real-time interactive dialogue system. We view it as a crucial design constraint on our pipeline modules that the resolution process must take place *incrementally*; i.e., processing must not be deferred until the end of the user's speech. This is because humans resolve (i.e., comprehend) speech as it unfolds (Tanenhaus, 1995; Spivey et al., 2002), and incremental processing (i.e., processing word by word) is important to developing an efficient and natural speech channel for interactive systems (Skantze and Schlangen, 2009; Paetzel et al., 2015; DeVault et al., 2009; Aist et al., 2007). In the current study, we have therefore provided the human director's correctly transcribed speech as input to our pipeline on a word-by-word basis, as visualized in Figure 3.

3.1 Segmenter

The segmenter module is tasked with identifying the boundary points between segments. In our pipeline, this task is performed independently of the determination of segment types, which is handled by a separate classifier (Section 3.2).

Our approach to segmentation is similar to Celikyilmaz et al. (2014) which used CRFs for a similar task. Our pipeline currently uses linear-chain CRFs to find the segment boundaries (implemented with Mallet (McCallum, 2002)). Using a CRF trained on the annotated RDG-Pento data set, we identify the most likely sequence of word-level boundary tags, where each tag indicates if the current word ends the previous segment or not.[2] An example segmentation is shown in Figure 3, where the word sequence *weird L to the top left of* is segmented into two segments, *[weird L]* and *[to the top left of]*. The features provided to the CRF include unigrams[3], the speaker's role, part-of-speech (POS) tags obtained using the Stanford POS tagger (Toutanova et al., 2003), and information about the scene such as the number of objects.

3.2 Segment Type Classifier

The segment type classifier assigns each detected segment with one of the type labels in Table 1 (SIN, MUL, REL, OT). This label informs the Reference Resolver module in how to proceed with the resolution process, as explained below.

The segment type labeler is an SVM classifier implemented in LIBSVM (Chang and Lin, 2011). Features used include word unigrams, word POS, user role, number of objects in the TI, and the top-level syntactic category of the segment as obtained from the Stanford parser (Klein and Manning, 2003). Figure 3 shows two examples of output from the segment type classifier, which assigns SIN to *[weird L]* and REL to *[to the top left of]*.

3.3 Reference Resolver

We introduce some notation to help explain the operation of the reference resolver (RR) module. When a scene description is to be resolved, there is a visual context in the game which we encode as a context set $\mathcal{C} = I_1, ..., I_8$ containing the eight visible images (see Figure 1). Each image I_k contains n objects $\{o_1^k, \ldots, o_n^k\}$, where n is fixed per context set, but varies across context sets from $n = 1$ to $n = 6$. The set of all objects in all images is $\mathcal{O} = \{o_l^k\}$, with $0 < k \leq 8, 0 < l \leq n$.

When the RR is invoked, the director has spoken some sequence of words which has been segmented by earlier modules into one or more segments $S_j = w_{1:m_j}$, and where each segment has been assigned a segment type $\text{type}(S_j) \in \{\text{SIN}, \text{MUL}, \text{REL}, \text{OT}\}$. For exam-

[2] We currently adopt this two-tag approach rather than BIO tagging as our tag-set provides a complete partition of each speaker's speech.

[3] Words of low frequency (i.e., <5) are replaced with a fixed symbol.

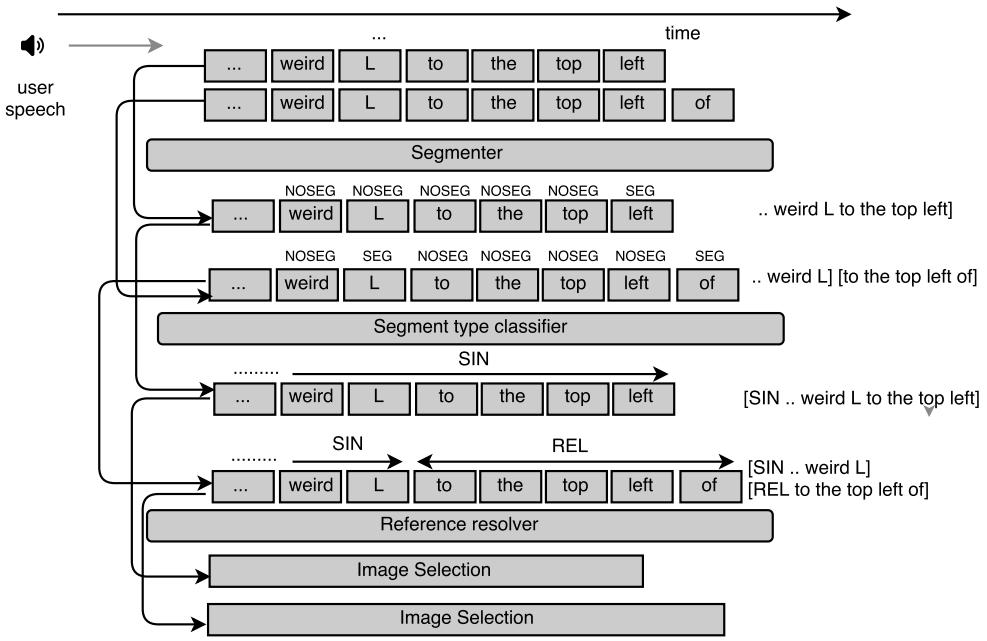


Figure 3: Information flow during processing of an utterance. The modules operate incrementally, word-by-word; as shown here, this can lead to revisions of decisions.

ple, $S_1 = \langle \text{weird}, \text{L} \rangle, S_2 = \langle \text{to}, \text{the}, \text{top}, \text{left}, \text{of} \rangle$ and $\text{type}(S_1) = \text{SIN}, \text{type}(S_2) = \text{REL}$.

The RR then tries to understand the individual words, typed segments, and the full scene description in terms of the visible objects o_l^k and the images I_k in the context set. We describe how words, segments, and scene descriptions are understood in the following three sections.

3.3.1 Understanding words

We understand individual words using the Words-as-Classifiers (WAC) model of Kennington and Schlangen (2015). In this model, a classifier is trained for each word w_p in the vocabulary. The model constructs a function from the perceptual features of a given object to a judgment about how well those features "fit" together with the word being understood. Such a function can be learned using a logistic regression classifier, separately for each word.

The inputs to the classifier are the low-level continuous features that represent the object (RGB values, HSV values, number of detected edges, x/y coordinates and radial distance from the center) extracted using OpenCV.[4] These classifiers are learned from instances of language use, i.e., by observing referring expressions paired with the object referred to. Crucially, once learned, these word classifiers can be applied to any number of objects in a scene.

We trained a WAC model for each of the (non-relational) words in our RDG-Pento corpus, using the annotated correct referent information for our segmented data. After training, words can be applied to objects to yield a score:

$$score(w_p, o_l^k) = w_p(o_l^k) \quad (1)$$

(Technically, the score is the response of the classifier associated with word w_p applied to the feature representation of object o_l^k.)

Note that relational expressions are trained slightly differently than non-relational words. Examples of relational expressions include *underneath*, *below*, *next to*, *left of*, *right of*, *above*, and *diagonal*. A WAC classifier is trained for each full relational expression e_q (treated as a single token), and the 'fit' for a relational expression's classifier is a fit for a *pair* of objects: (The features used for such a classifier are comparative features, such as the euclidean distance between the two objects, as well as x and y distances.)

$$score_{rel}(e_q, o_{l_1}^k, o_{l_2}^k) = e_q(o_{l_1}^k, o_{l_2}^k) \quad (2)$$

There are about 300 of these expressions in RDG-Pento. *[SIN x] [REL r] [SIN y]* is resolved as

[4] http://opencv.org

r(x,y), so x and y are jointly constrained. See Kennington and Schlangen (2015) for details on this training.

3.3.2 Understanding segments

Consider an arbitrary segment $S_j = w_{1:m_j}$ such as $S_1 = \langle \text{weird}, \text{L} \rangle$. For a segment (SIN or MUL), we attempt to understand the segment as referring to some object or set of objects. To do so, we combine the word-level scores for all the words in the segment to yield a segment-level score[5] for each object o_l^k:

$$score(S_j, o_l^k) = score(w_1, o_l^k) \odot \ \ldots \ \odot \ score(w_{m_j}, o_l^k) \quad (3)$$

Each segment $S_j = w_{1:m_j}$ hence induces an order R_j on the object set $\mathcal{O}$, through the scores assigned to each object o_l^k. With these ranked scores, we look at the type of segment to compute a final score $score_k^*(S_j)$ for each image I_k. For SIN segments, $score_k^*(S_j)$ is the score of the top-scoring object in I_k. For MUL segments with a cardinality of two (e.g., *two red crosses*), $score_k^*(S_j)$ is the sum of the scores of the top two objects in I_k, and so on.

Obtaining the final score $score_k^*(S_j)$ for REL segments is done in a similar manner with some minor differences. Because REL segments express a relation between *pairs* of objects (referred to in neighboring segments), a score for the relational expression in S_j can be computed for any pair of distinct objects $o_{l_1}^k$ and $o_{l_2}^k$ in image I_k using Eq. (2). We let $score_k^*(S_j)$ equal the score computed for the top-scoring objects $o_{l_1}^k$ and $o_{l_2}^k$ of the neighboring segments.

3.3.3 Understanding scene descriptions

In general, a scene description consists of segments $S_1, ..., S_z$. Composition takes segments $S_1, ..., S_z$ and produces a ranking over images. For this particular task, we make the following assumption: in each segment, the speaker is attempting to refer to a specific object (or set of objects), which from our perspective as matcher could be in any of the images. A good candidate I_k for the target image will have high scoring objects, all drawn from the same image, for all the segments $S_1, ..., S_z$.

We therefore obtain a final score for each image as shown in Eq. (4):

[5]The composition operator $\odot$ is left-associative and hence incremental. In this paper, word-level scores are composed by multiplying them.

Label	Precision	Recall	F-Score
SEG	0.85	0.74	0.79
NOSEG	0.93	0.97	0.95

Table 2: Segmenter performance

Label	Precision	Recall	F-score	% of segments
SIN	0.91	0.96	0.93	57
REL	0.97	0.85	0.91	6
MUL	0.86	0.60	0.71	3
OT	0.96	0.97	0.96	34

Table 3: Segment type classifier performance

$$score(I_k) = \sum_{j=1}^{z} score_k^*(S_j) \quad (4)$$

The image I_k^* selected by our pipeline for a full scene description is then given by:

$$I_k^* = \underset{k}{\text{argmax}}\, score(I_k) \quad (5)$$

4 Experiments & Evaluations

We first evaluate the segmenter and segment type classifier as individual modules. We then evaluate the entire processing pipeline and explore the impact of several factors on pipeline performance.

4.1 Segmenter Evaluation

Task & Data We used the annotated RDG-Pento data to perform a "hold-one-dialogue-pair-out" cross-validation of the segmenter. The task is to segment each speaker's speech for each target image by tagging each word using the tags SEG and NOSEG. The SEG tag here indicates the last word in the current segment. Figure 3 gives an example of the tagging.

Results The results are presented in Table 2. These results show that the segmenter is working with some success, with precision 0.85 and recall 0.74 for the SEG tag indicating a word boundary. Note that occasional errors in segment boundaries may not be overly problematic for the overall pipeline, as what we ultimately care most about is accurate target image selection. We evaluate the overall pipeline below (Section 4.3).

4.2 Segment Type Classifier Evaluation

Task & Data We used the annotated RDG-Pento data to perform a hold-one-pair-out cross-validation of the segment type classifier, training a

SVM classifier to predict labels SIN, MUL, REL, and OT using the features described in Section 3.2.

Results The results are given in Table 3. We also report the percentage of segments that have each label in the corpus. The segment type classifier performs well on most of the class labels. Of slight concern is the low-frequency MUL label. One factor here is that people use number words like *two* not just to refer to multiple objects, but also to describe individual objects, e.g., *the two red crosses* (a MUL segment) vs. *the one with two sides* (a SIN segment).

4.3 Pipeline Evaluation

We evaluated our pipeline under varied conditions to understand how well it works when segmentation is not performed at all, when the segmentation and type classifier modules produce perfect output (using oracle annotations), and when entrainment to a specific speaker is possible. We evaluate our pipeline on the accuracy of the task of image retrieval given a scene description from our data set.

4.3.1 Three baselines

We compare against a weak random baseline (1/8 = 0.125) as well as a rather strong one, namely the accuracies of the human-human pairs in the RDG-Pento corpus. As Table 4 shows, in the simplest case, with only one object per image, the average human success rate is 85%, but this decreases to 60% when there are four objects/image. It then increases to 68% when 6 objects are present, possibly due to the use of a more structured description ordering in the six object scenes. We leave further analysis of the human strategies for future work. These numbers show that the game is challenging for humans.

We also include in Table 4 a simple Naive Bayes classification approach as an alternative to our entire pipeline. In our study, there were only 40 possible image sets that were fixed in advance. For each possible image set, a different Naive Bayes classifier is trained using Weka (Hall et al., 2009) in a hold-one-pair-out cross-validation. The eight images are treated as atomic classes to be predicted, and unigram features drawn from the union of all (unsegmented) director speech are used to predict the target image. This method is broadly comparable to the NLU model used in (Paetzel et al., 2015) to achieve high performance in resolving references to pictures of single objects. As can be seen, the accuracy for this method is as high as 43% for single object TIs in the RDG-Pento data set, but the accuracy rapidly falls to near the random baseline as the number of objects/image increases. This weak performance for a classifier without segmentation confirms the importance of segmenting complex descriptions into references to individual objects in the RDG-Pento game.

4.3.2 Five versions of the pipeline

Table 4 includes results for 5 versions of our pipeline. The versions differ in terms of which segment boundaries and segment type labels are used, and in the type of cross-validation performed. A first version (I) explores how well the pipeline works if unsegmented scene descriptions are provided and a SIN label is assumed to cover the entire scene description. This model is broadly comparable to the Naive Bayes baseline, but substitutes a WAC-based NLU component. The evaluation of version (I) uses a hold-one-pair-out (HOPO) cross-validation, where all modules are trained on every pair except for the one being used for testing. A second version (II) uses automatically determined segment boundaries and segment type labels, in a HOPO cross-validation, and represents our pipeline as described in Section 3. A third version (III) substitutes in human-annotated or "oracle" segment boundaries and type labels, allowing us to observe the performance loss associated with imperfect segmentation and type labeling in our pipeline. The fourth and fifth versions of the pipeline switch to a hold-one-episode-out (HOEO) cross-validation, where only the specific scene description ("episode") being tested is held out from training. When compared with a HOPO cross-validation, the HOEO setup allows us to investigate the value of learning from and entraining to the specific speaker's vocabulary and speech patterns (such as calling the purple object in Figure 2 a "harry potter").

4.3.3 Results

Table 4 summarizes the image retrieval accuracies for our three baselines and five versions of our pipeline. We discuss here some observations from these results. First, in comparing pipeline versions (I) and (II), we observe that the use of automated segmentation and a segment type classifier in (II) leads to a substantial increase in accuracy of 5-20% ($p<0.001$)[6] depending on the

[6] wilcoxon rank sum test

		#objects per TI				
		1	2	3	4	6
Random baseline		0.13	0.13	0.13	0.13	0.13
Naive Bayes baseline		0.43	0.20	0.14	0.14	0.13
Seg+lab	X-validation					
(I) None	HOPO	0.47	0.20	0.24	0.13	0.15
(II) Auto	HOPO	0.52	0.40	0.31	0.24	0.23
(III) Oracle	HOPO	0.54	0.42	0.32	0.30	0.26
(IV) Auto	HOEO	0.60	0.46	0.37	0.25	0.23
(V) Oracle	HOEO	0.64	0.50	0.41	0.34	0.44
Human-human baseline		0.85	0.73	0.66	0.60	0.68

Table 4: Image retrieval accuracies for five versions of the pipeline and three baselines.

number of objects/image. Comparing (II) and (III), we see that if our segmenter and segment type classifier could reproduce the human segment annotations perfectly, an additional improvement of 1-6% ($p<0.001$) accuracy would be possible. Comparing (II) to (IV), we see that exposing our pipeline training to the idiosyncratic speech and vocabulary of a given speaker would hypothetically enable an increase in accuracy of up to 8% ($p<0.001$). Note however that this setup cannot easily be replicated in a real-time system, as our HOEO training provides not only samples of the transcribed speech of the same speaker, but also human annotations of the segment boundaries, segment types, and correct referents for this speech (which would not generally be available for immediate use in a run-time system). Comparing (IV) to (V), we see that oracle segment boundaries and types also improve accuracies in a HOEO evaluation between 4-19% ($p<0.001$). Comparing our fully automated HOPO pipeline (II) to the baselines, we see that our pipeline performs considerably better than the random and Naive Bayes baselines. At the same time, there is still much room for improvement when we compare to human-human accuracy. Segmentation is harder the more objects (and hence segments) there are. Compared to HOEO, HOPO is additionally hurt by idiosyncratic vocabulary that isn't learned, so even with oracle segmentations, performance does not increase as much.

4.4 Evaluation of Object Retrieval

Table 4 shows that even when there is just one object in each of the eight images, our pipeline (II) only selects the correct image 52% of the time given the complete scene description, while humans succeed 85% of the time. We further investigated our performance at understanding descriptions of individual objects by defining a constructed "object retrieval" problem. In this problem, individual SIN segments from the RDG-Pento corpus are considered one at a time, and the correct target image is provided by an oracle. The only task is to use the WAC model to select the correct referent object within the image for a single SIN segment. An example of the object retrieval problem is to select the correct referent for the SIN segment *and a wooden w sort of* in the known target image of Figure 1.

n	1	2	3	4	6
accuracy	1	.88	.77	.60	.66

Table 5: Accuracy for object retrieval in target images with n objects.

The results are shown in Table 5. We can observe that object retrieval is by itself a non-trivial problem for our WAC model, especially as the number of objects increases. This is somewhat by design in that the multiple objects present within an image are often selected to be fairly similar in their properties, and multiple objects may match ambiguous SIN segments such as *the T* or *the plus sign*. We speculate that we could gain here from factoring in positional information implicit in description strategies such as going from top left to bottom right in describing the objects.

5 Related Work

The work described in this paper directly builds off of Paetzel et al. (2015) as the same RDG game scenario was used, however reference was only made to single objects in that work. The work here also builds off of Kennington and Schlangen (2015) in the same way in that their work only focused on reference to single objects. The extension of this previous work to handle more complex scene descriptions required substantial composition on the word and segment levels. The segmentation presented here was fairly straight forward (similar in spirit to chunking as in Marcus (1995)). Composition is currently an active area in distributional semantics where word meanings are represented by high-dimensional vectors and composition amounts to some kind of vector operation (see (Milajevs et al., 2014) for a comparison of methods). An important difference is that here words and segments are composed at the denotational level (i.e., on the scores given by the

WAC model, akin to *referentially afforded concept composition* (Mcnally and Boleda, 2015)). Also related are the recent efforts in automatic image captioning and retrieval, where the task is to generate a description (a caption) for a given image or retrieve one being given a description. A frequently taken approach is to use a convolutional neural network to map the image into a dense vector, and then to condition a neural language model on this to produce an output string or using it to map the description into the same space (Vinyals et al., 2015; Devlin et al., 2015; Socher et al., 2014). See also Fang et al. (2015), which is more directly related to our model in that they use "word detectors" to propose words for image regions.

6 Conclusions & Future work

We have presented an approach to understanding complex, multi-utterance references to images containing spatially complex scenes. The approach by design works incrementally, and hence is ready to be used in an interactive system. We presented evaluations that go end-to-end from utterance input to resolution decision (but not yet taking in speech). We have shown that segmentation is a critical component for understanding complex visual scene descriptions. This work opens avenues for future explorations in various directions. Intra- and inter-segment composition (through multiplication and addition, respectively) are approached somewhat simplistically, and we want to explore the consequences of these decisions more deeply in future work. Additionally, as discussed above, there seems to be much implicit information in how speakers go from one reference to the next, which might be possible to capture in a transition model. Finally, in an online setting, there is more than just the decision "this is the referent"; one must also decide when and how to act based on the confidence in the resolution. Lastly, our results have shown that human pairs do align on their conceptual description frames (Garrod and Anderson, 1987). Whether human users would also do this with an artificial interlocutor, if it were able to do the required kind of online learning, is another exciting question for future work, enabled by the work presented here. We also plan to extend our work in the future to include descriptions which contain relations between non singular objects (Ex: [MUL two red crosses] [REL above] [SIN brown L], [MUL two red crosses] [REL on top of] [MUL two green Ts] etc.). However, such descriptions were very rare in the corpus.

Obtaining samples for training the classifiers is another issue. One source of sparsity is idiosyncratic descriptions like 'harry potter' or 'facebook'. In dialogue (our intended setting), these could be grounded through clarification requests. A more extensive solution would address metaphoric or meronymic usage ("looks like xyz"). We will explore this in future work.

Acknowledgments

This work was in part supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG), and the KogniHome project, funded by BMBF. This work was supported in part by the National Science Foundation under Grant No. IIS-1219253 and by the U.S. Army. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views, position, or policy of the National Science Foundation or the United States Government, and no official endorsement should be inferred.

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K Tanenhaus. 2007. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *the 29th Annual Conference of the Cognitive Science Society*.

Asli Celikyilmaz, Zhaleh Feizollahi, Dilek Hakkani-Tur, and Ruhi Sarikaya. 2014. Resolving referring expressions in conversational dialogs for natural user interfaces. In *Proceedings of EMNLP*.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish?: learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–20. Association for Computational Linguistics.

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.

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.

Simon Garrod and Anthony Anderson. 1987. Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27:181–218.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Ting Han, Casey Kennington, and David Schlangen. 2015. Building and Applying Perceptually-Grounded Representations of Multimodal Scene Descriptions. In *Proceedings of SEMDial*, Gothenburg, Sweden.

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)*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Ramesh Manuvinakurike and David DeVault. 2015. Pair me up: A web framework for crowd-sourced spoken dialogue collection. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 189–201. Springer.

Mitchell P Marcus. 1995. Text Chunking using Transformation-Based Learning. In *In Pro- ceedings of the Third Workshop on Very Large Cor- pora*, pages 82–94.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Louise Mcnally and Gemma Boleda. 2015. Conceptual vs. Referential Affordance in Concept Composition. In *Compositionality and Concepts in Linguistics and Psychology*, pages 1–20.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, Matthew Purver, Computer Science, and Mile End Road. 2014. Evaluating Neural Word Representations in Tensor-Based Compositional Settings. In *EMLNP*, pages 708–719.

Maike Paetzel, Ramesh Manuvinakurike, and David DeVault. 2015. "So, which one is it?" The effect of alternative incremental architectures in a high-performance game-playing agent. In *The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SigDial)*.

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive grammar of the English language*. General Grammar Series. Longman.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 745–753. Association for Computational Linguistics.

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)*.

Michael J Spivey, Michael K Tanenhaus, Kathleen M Eberhard, and Julie C Sedivy. 2002. Eye movements and spoken language comprehension: Effects of visual context on syntactic ambiguity resolution. *Cognitive Psychology*, 45(4):447–481.

Michael Tanenhaus. 1995. Integration of Visual and Linguistic Information in Spoken Language Comprehension. *Science*, 268:1632–1634.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

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, Julian Hough, Casey Kennington, Ramesh Manuvinakurike, David DeVault, and David Schlangen. 2016. Pentoref: A corpus of spoken references in task-oriented dialogues.