A Simple Method for Resolution of Definite Reference in a Shared Visual Context

Alexander Siebert
Berlin-Brandenburgische
Akademie der Wissenschaften
siebert@bbaw.de

David Schlangen
Department of Linguistics
University of Potsdam, Germany
das@ling.uni-potsdam.de

Abstract

We present a method for resolving definite exophoric reference to visually shared objects that is based on a) an automatically learned, simple mapping of words to visual features ("visual word semantics"), b) an automatically learned, semantically-motivated utterance segmentation ("visual grammar"), and c) a procedure that, given an utterance, uses b) to combine a) to yield a resolution. We evaluated the method both on a pre-recorded corpus and in an online setting, where it performed with 81% (chance: 14%) and 66% accuracy, respectively. This is comparable to results reported in related work on simpler settings.

1 The Task

The method described in this paper is a module of a dialogue system that acts as a collaborator of a human player in the task of manipulating visually present puzzle objects. An example scene is shown in Figure 1 (the indices *a* and *b* are added here for illustrative purposes). Given utterances like those in (1), the task of the module is to identify the likely referents (here, *a* and *b*, respectively).[1]

(1) a.Take the piece in the middle on the left side.
b.Take the piece in the middle.

More formally, the task can be characterised as follows: possibly starting with an *a priori* assumption about likely referents (e.g., from knowledge of discourse salience), the module uses the evidence present in the utterance (words, syntax) and in the visual scene (visual features) to derive at a new assumption about likely referents. If we call such an assumption a *confidence function* c that assigns to each object in the domain $\mathcal{O}$, a number between 0 and 1; i.e., $c : \mathcal{O} \rightarrow \mathbb{R}$, then *reference resolution* is a function r that takes a triple of an initial confidence function c, an utterance u, and a visual scene representation v to yield an updated confidence function c'. Formally: $r : \mathcal{C} \times \mathcal{U} \times \mathcal{V} \rightarrow \mathcal{C}$.

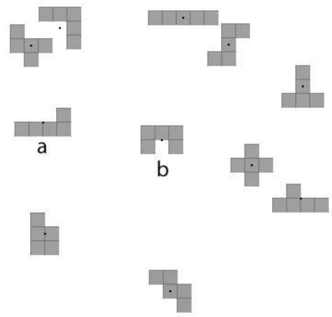

Figure 1: Example Scene

In the following, we describe the resources needed to set up the module, its subcomponents, and the evaluation we performed. We close by relating the proposed method to prior work and discussing future extensions.

2 Resources

2.1 Corpus

As our method is based on automatically learned models, a corpus is required. Our intended use case is similar to the setting described in (Schlangen and Fernández, 2007), but with the addition of a shared visual context. We collected 300 scene descriptions

[1]Our system is implemented for German input; for ease of description we use examples from our corpus translated into English here.

Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pages 84–87,
Columbus, June 2008.

(of scenes containing between 1 and 12 distinct, monochrome shapes, randomly placed and rotated on a rectangular area) using the two-part methodology of (Siebert et al., 2007) that yields recordings and quality assessments (here: attempts to follow other subjects' instructions). We also later recorded an additional 300 scene descriptions by a single speaker, to further increase our data base.

After transcription of the recordings (239 minutes of audio material), we discarded roughly 6% of the instructions because they could not be followed by the evaluators, and a further 4% because the complexity of the descriptions was outside the scope of what we wanted to model. The remaining instructions were then automatically cleaned from dysfluencies, morphologically lemmatised and POS tagged, and annotated as described below.

2.2 Computer Vision

The other required resource is a visual perception algorithm. We use it to compute a feature representation of every visual scene as presented in the data collection:[2] First, each object is represented by a number of *object features* such as size / length / height of the bounding box, center of gravity, number of edges. Second, *topological features* note for each object the distance to certain points on the board (edges, center, etc.) and to other objects. (For details on the computation of such features see for example (Regier and Carlson, 2001).) Lastly, we also compute groupings of objects by clustering along columns and rows or both (see Figure 2 for an illustration). For each group, we compute two sets of *topological features*, one for the objects within the group (e.g., distance to the center of the group), and one for the configuration of groups (distance of group to other objects). This set of features was selected to be representative of typical basic visual features.

3 Components

3.1 Visual Grammar

The 'visual grammar' segments utterances according to functional aspects on two levels. The first describes the macro-structure of a spatial expression, i.e., the division into *target* (the denoted object; T) and optional *landmark*s (other objects; LM) and their *relation* to the target (R; see example in Table 2). The second level annotates the spatial-lexical function of each word, e.g., whether the word denotes a piece or a configuration of pieces (Table 1). A fully 'parsed' example is shown in Table 2.

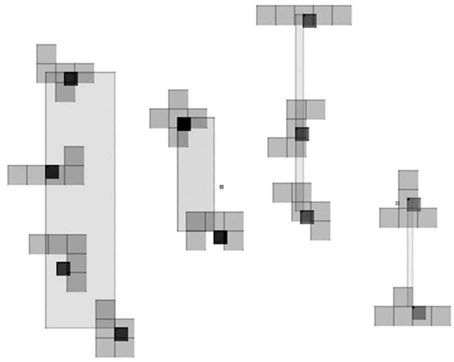

Figure 2: Scene with Horizontal Group Detection

Name	Description	Examples
l	lexical reference	***T**,**piece**,**cross***
d_r	topological direction	***top left** Corner*
d_s	topological distance	***outer** left*
d_n	numeric	***second** column*
p_g	group (perceptually active)	*from the left* **column**
g_s	synthetic group	*the three* **pieces** *on the left*
f	landmark field N	in the *Middle*
r	prepositional relation	***in** the middle*
grad	grading function	***exactly** right*

Table 1: Visual Lexical Functions of Words

the	cross	from	the	second	column	from	left	at the top
	l	r		d_n	p_g	r	d_r	d_r

(a) - **Annotation of spatial lexical functions**

	T	R		LM	LM	LM	LM	T

(b) - **Segmentation of visual spatial parts**

Table 2: Example Annotation / 'Parse'

Given the requirement for robustness, we decided against a hand-written grammar for deriving such annotations; the moderate size of our corpus on the other hand made for example Markov model-based approaches difficult to apply. We hence chose transformation-based learning to create this (shallow) segmentation grammar, converting the segmentation task into a tagging task (as is done in

[2]At the moment, the input to the algorithm is a symbolic representation of the scene (which object is where); the features are designed to also be derivable from digital images instead, using standard computer vision techniques (Shapiro and Stockman, 2001); this is future work, however.

(Ramshaw and Marcus, 1995), *inter alia*). In our approach, each token that is to be tagged is itself represented in three different forms or layers: lemmatised word, as POS-tag, and by its spatial-functional tag (as in Table 1; added by simple look-up). All these layers can be accessed in the learned rules. Apart from this, the module is a straightforward implementation of (Ramshaw and Marcus, 1995), which in turn adapts (Brill, 1993) for syntactic chunking.

3.2 Visual Word Semantics

To learn the visual semantics of words we implemented a simple technique for grounding words in perceptions. Roughly, the idea is to extract from all instances in which a word was used in the training corpus and all associated scenes a prototypical visual meaning representation by identifying those features whose values best predict the appropriateness of the word given a scene. (This is similar in spirit to the approach used in (Roy, 2002).)

As material for learning, we only used the simple expressions (target only, no landmark) in the corpus, to ensure that all words used were in some way 'about' the target. The algorithm iterates over all pairs of utterance and scene and saves for each lemma all visual information. This creates for each lemma a matrix of feature values with as many rows as there were occurrences of the lemma. The values in each column (that is, for each feature) are then normalised to the interval [-1, 1] and the standard deviation is recorded.

The next tasks then are a) to compute one single representative value for each feature, but only b) for those features that carry semantic weight for the given word (i.e., to compute a dimensionality reduction). E.g., for the lemma 'left', we want the feature *x_distance_to_center* to be part of the semantic model, but not *y_distance_to_center*.

One option for a) is to simply take the average value as representative for a feature (for a given word). While this works for some words, it causes problems for others which imply a maximisation and not a prototypisation. E.g., the lemma *left* is best represented by *maximal* values of the feature *x_distance_to_center*, not by the average of all values for all occurrences of *left* (this will yield something like *leftish*). Perhaps surprisingly, representation through the majority value, i.e., choosing the most frequent value as representative for a feature (for a given word), performed better, and is hence the method we chose.

For b), dimensionality reduction, we again chose a very simple approach (much simpler than for example (Roy, 2002)): features are filtered out as irrelevant for a given lemma features if their variance is above a certain threshold. To give an example, for the lemma *left* the distribution of values of the feature *x_distance_to_center* varies with a σ of 0.05, that of *y_distance_to_center* with a σ of 0.41. We empirically determined the setting of the threshold such that it excluded the latter.[3]

3.3 Combination

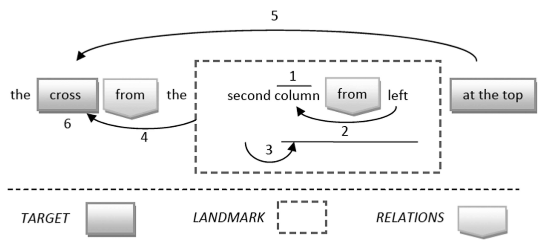

Figure 3: Steps of the Algorithm for Example Utterance

The combination algorithm works through the segmented utterance and combines visual word semantics to yield a reference hypothesis. Figure 3 illustrates this process for the example from Table 2. On detecting a landmark segment (Step 1), the resolution algorithm 'activates' the appropriate group; which one this is is determined by the *p_g* item in the landmark segment. (Here: *column*). The group is then treated as a single object, and (Step 2) the semantics of topological terms (d_r or d_s) in the landmark segment is applied to it (more on this in a second). For our example, this yields a ranking of all columns with respect to their 'left-ness'. The ordinal 'second' finally simply picks out the second element on this list–the second group w.r.t. the property of leftness (Step 3). The expressions in the target segment are now only applied to the members of the group that was selected in this way; i.e., the semantic models of 'top' and 'cross' are now only applied to the objects in that column (Steps 4 to 6).

[3]With more data and hence the possibility to set aside a development set, one could and should of course set such a threshold automatically.

Semantic word models are applied through a simple calculation of distance between values (of semantic model and actual scene): the closer, the better the match of word to scene. (Modulo selectivity of a feature; for a feature that occurred for all lemmata with a high specificity (small σ), good matches are expected to be closer to the prototype value than for features with a high variability.)

This method encodes parts of the utterance semantics procedurally, namely the way how certain phrases (here grouped under the label *landmark*) semantically modify other phrases (here grouped under the label *target*). This encoding makes the algorithm perhaps harder to understand than semantic composition rules tied to syntactic rules, but it also affords a level of abstraction over specific syntactic rules: our very general concepts of *landmark* and *target* cover various ways of modification (e.g. through PPs or relative clauses), adding to the robustness of the approach.

4 Evaluation

With an f-score of 0.985 (10-fold cross validation), the transformation-based learning of the segmentation performs quite well, roughly at the level of state-of-the-art POS-taggers (albeit with a much smaller tag inventory). Also evaluated via cross-validation on the corpus, the resolution component as a whole performs with an accuracy of 80.67% (using frequency-based word-semantic features; it drops to 66.95% for average-based). There were on average 7 objects in each scene in the corpus; i.e. the baseline of getting the reference right by chance is 14%. Our system significantly improves over this baseline.

We also evaluated the system in a more realistic application situation. We asked subjects to refer to certain pieces in presented scenes (via typed utterances); here, the system reached a success-rate of 66% (7 subjects, 100 scene / utterance pairs). While this is considerably lower than the corpus-based evaluation, it is still on a par with related systems using more complicated resolution methods (Roy, 2002; Gorniak and Roy, 2004). We also think these results represent the lower end of the performance range that can be expected in practical use, as in an interactive dialogue system users have time to adapt to the capabilities of the system.

5 Conclusions

We have presented a method for resolving definite, exophoric reference to objects that are visually co-present to user and system. The method combines automatically acquired models (a 'visual word semantics', a simple, but effective mapping between visual features and words; and a 'visual grammar', a semantically motivated segmentation of utterances) and hard-coded knowledge (combination procedure). To us, this combines the strengths of two approaches: statistical, where robustness and wide coverage is required, hard-coding, where few, but complex patterns are concerned.

We are currently integrating the module into a working dialogue system; in future work we will investigate the use of digital images as input format.

Acknowledgements

This work was supported by DFG through an Emmy Noether Programm Grant to the second author.

References

Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, University of Pennsylvania.

Peter Gorniak and Deb Roy. 2004. Grounded semantic composition for visual scenes. In *Journal of Artifical Intelligence Research*.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94.

Terry Regier and Laura A. Carlson. 2001. Grounding spatial language in perception: An empirical and computational investigation. In *Journal of Experimental Psychology*, volume 130, pages 273–298.

Deb Roy. 2002. Learning words and syntax for a visual description task. *Computer Speech and Language*, 16(3).

David Schlangen and Raquel Fernández. 2007. Beyond repair: Testing the limits of the conversational repair system. In *Proceedings of SIGdial 2007*, pages 51–54, Antwerp, Belgium, September.

Linda G. Shapiro and George C. Stockman. 2001. *Computer Vision*. Prentice Hall, New Jersey, USA.

Alexander Siebert, David Schlangen, and Raquel Fernández. 2007. An implemented method for distributed collection and assessment of speech data. In *Proceedings of SIGdial 2007*, Antwerp, Belgium.