Probabilistic Refinement Algorithms for the Generation of Referring Expressions

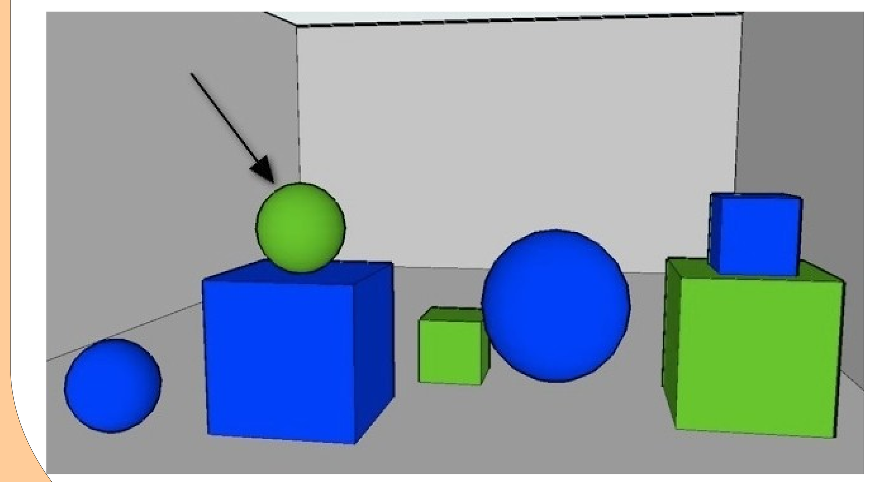
Romina Altamirano[1] Carlos Areces[1,2] Luciana Benotti [1]

(1) FaMAF, Universidad Nacional de Córdoba, Argentina, (2) Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

INTRODUCTION

We propose a **refinement** algorithm that generates referring expressions that are:

- Relational: *ball on the blue cube*
- Overspecified: *small green ball on the blue cube*
- Non-deterministic: *green ball, small ball on the right*



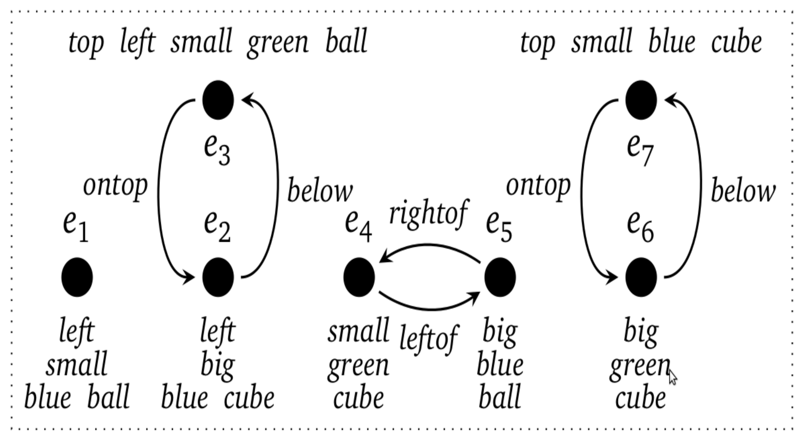


Figure 2: Scene as a relational model

LEARNING PROBABILITIES OF USE

If there is corpus available:
$R.p_{use} = \#appear(R)/\#R_E$

Feature used to learn $R.p_{use}$ when there is not corpus available

R.target-has	true if the target is in R
R.landmark-has	true if a landmark is in R
R.discrimination	1 / objects in the model that have R
#bin-relations	number of binary relations of the target
#relations	number of relations of the target

REFINEMENT ALGORITHMS NON-DETERMINISM AND OVER-SPECIFICATION

Algorithm 1: Computing $\mathscr{L}$-similarity classes

```
RE ← {⊤}
repeat
    while ∃(φ ∈ RE).(#‖φ‖ > 1) do
        RE' ← RE
        for (R, R.p_use) ∈ Rs do
            if R.rnd_use ≤ R.p_use then
                for φ ∈ RE do add_EL(R, φ, RE)
            if RE ≠ RE' then
                exit
        if RE = RE' then
            exit
    for (R,R.p_use) ∈ Rs do R.p_use ← R.p_use + R.inc_use
until ∀((R,R.p_use) ∈ Rs).(R.p_use ≥ 1)
```

Probabilistic Non-deterministic step

Completeness step

Algorithm 2: $add_{\mathscr{EL}}(R, \varphi, RE)$

```
if FirstLoop? then
    Informative ← TRUE
else Informative ← ‖ψ ⊓ ∃R.φ‖ ≠ ‖ψ‖;
for ψ ∈ RE with #‖ψ‖ > 1 do
    if ψ ⊓ ∃R.φ is not subsumed in RE and
      ‖ψ ⊓ ∃R.φ‖ ≠ ∅ and
      Informative then
        add ψ ⊓ ∃R.φ to RE
        remove subsumed formulas from RE
```

Overspecification Step/ (Egocentric)

Refinement Step/(Adjustment)

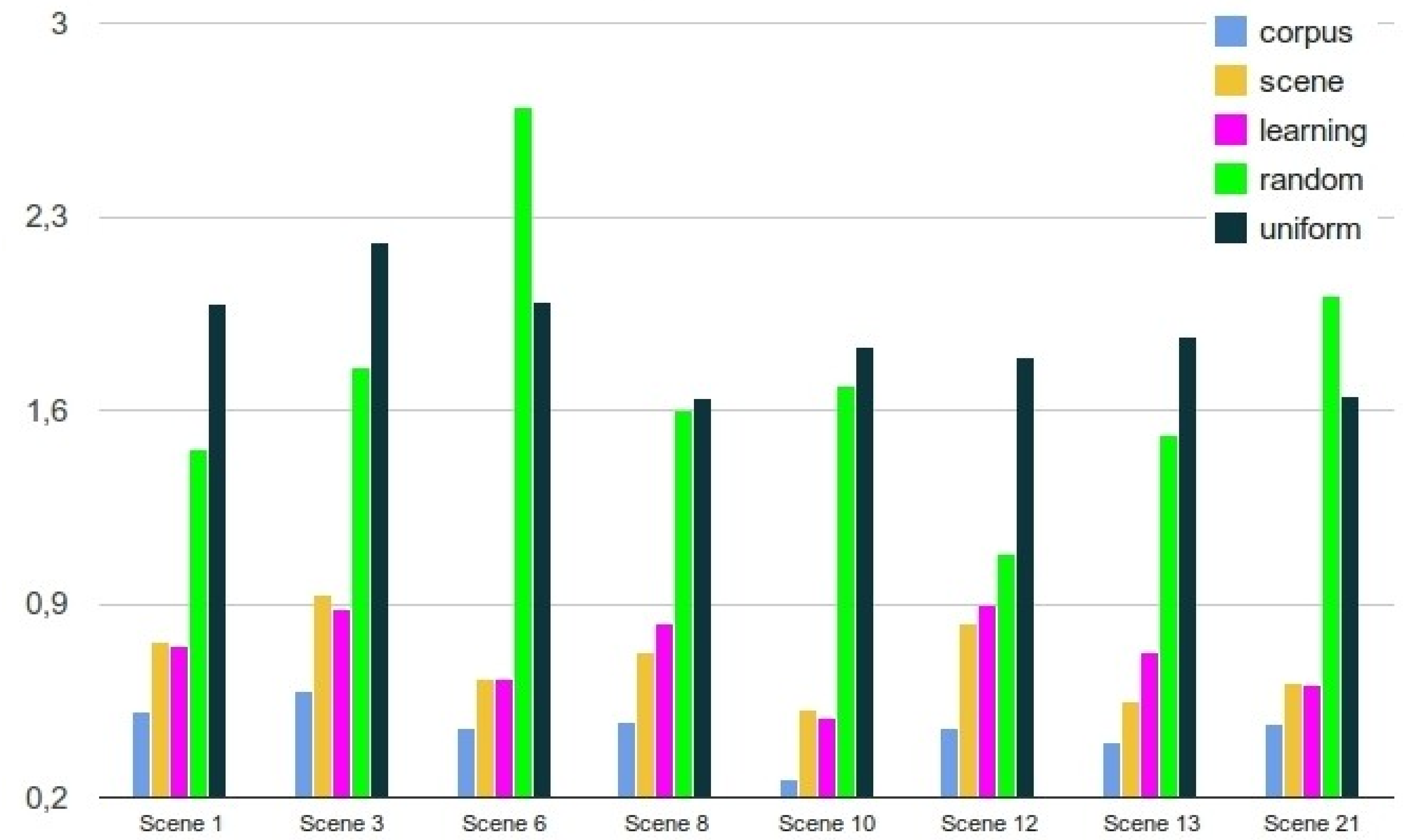
EVALUATION WITH PRECISION

Accuracy between the REs in the corpus and those generated using puse values computed from the scene, machine learned, random and uniform.

	Scene p_{use}	Learned p_{use}	Random p_{use}	Uniform p_{use}
Scene 1	85.75%	84.49%	17.95%	5.37%
Scene 3	82.81%	80.51%	9.89%	4.40%
Scene 6	90.11%	83.30%	4.13%	4.16%
Scene 8	86.52%	64.06%	16.32%	9.75%
Scene 10	89.49%	75.80%	7.56%	3.70%
Scene 12	80.21%	81.29%	57.09%	6.68%
Scene 13	89.98%	50.79%	9.30%	3.59%
Scene 21	92.13%	80.01%	8.45%	6.77%
Average	87.13%	75.03%	16.34%	5.55%

EVALUATION WITH CROSS ENTROPY

Cross-entropy between the corpus distribution and different runs of the algorithm.



- Our algorithm is able to generate different referring expressions for the same target with a frequency similar to that observed in corpora.
- Our results support the psycholinguistic theory that puts forward an egocentric explanation of language production (Keysar, 1998).