Un algoritmo de refinamiento para la generación de expresiones referenciales

Jornadas de doctorandos 2013

Romina Altamirano
Universidad Nacional de Córdoba

Generación de expresiones referenciales

Aprendizaje de “perceptual saliency”

Algoritmo de refinamiento probabilístico

Evaluación y resultados

Generación de expresiones referenciales

Aprendiendo “perceptual saliency”

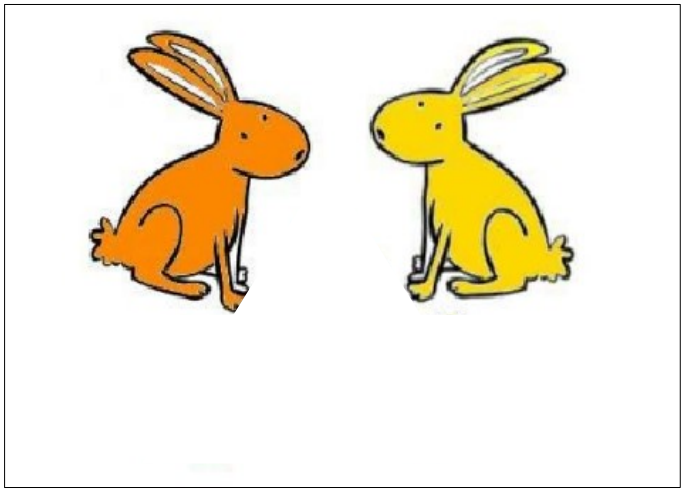
Algoritmo de refinamiento probabilístico

Evaluación y resultados

Qué es una Expresión Referencial?

Una expresión referencial es:

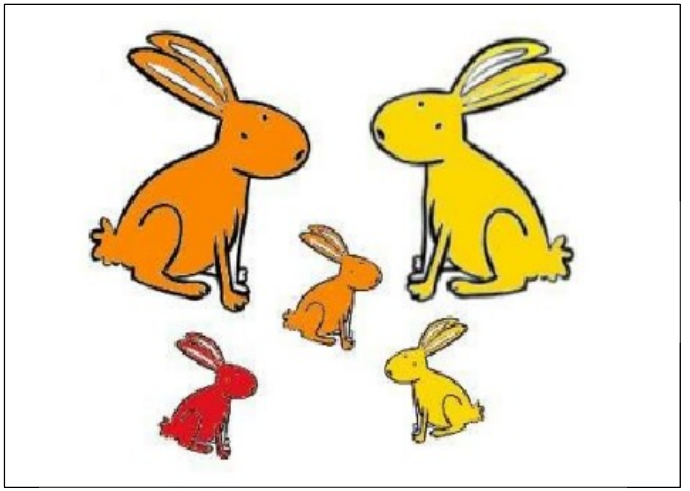
- Una frase que identifica un objeto en un contexto



Qué es una Expresión Referencial?

Una expresión referencial es:

- Una frase que identifica un objeto en un contexto

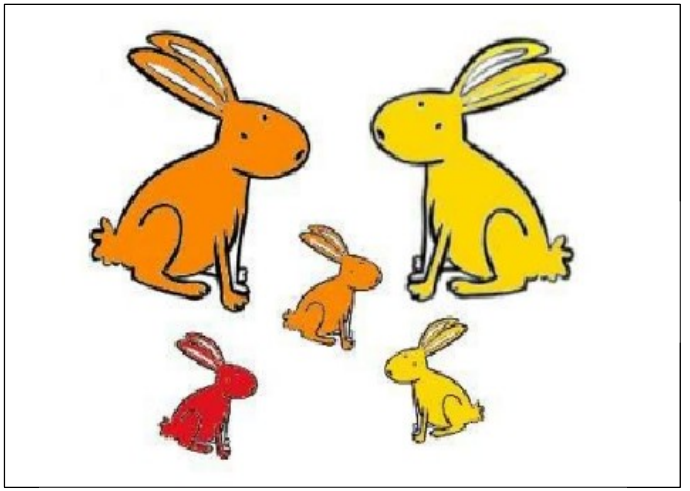


Qué es una Expresión Referencial?

Una expresión referencial es:

- Una frase que identifica un objeto en un contexto
- Sobreespecificada

  “el conejo pequeño rojo”



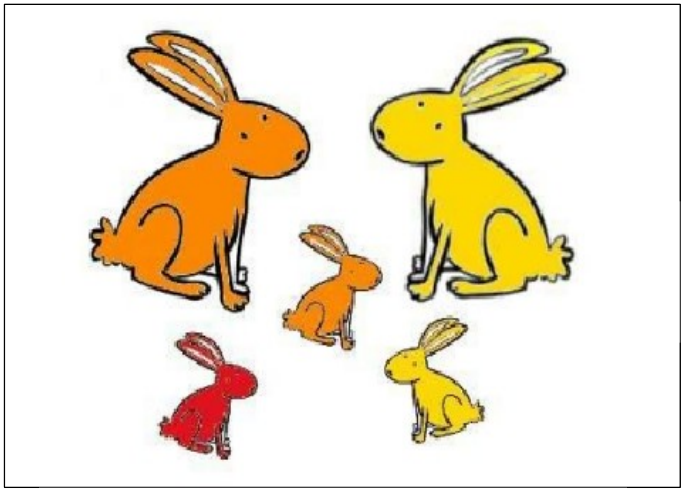
Qué es una Expresión Referencial?

Una expresión referencial es:

- Una frase que identifica un objeto en un contexto
- Sobreespecificada
  - “El conejo pequeño rojo”
- No-deterministica

  “El rojo”

  “El conejo rojo”



Qué es una Expresión Referencial?

Una expresión referencial es:

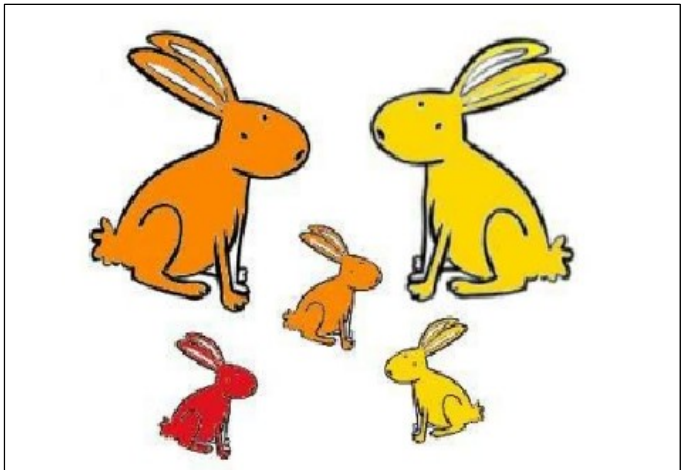
- Una frase que identifica un objeto en un contexto
- Sobreespecificada

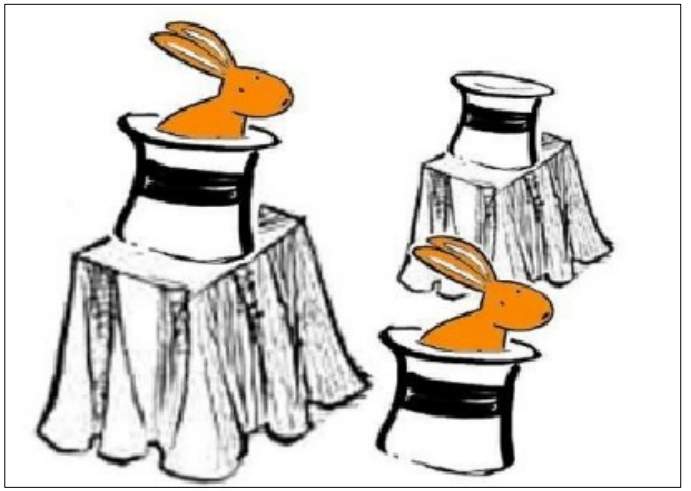
  “El conejo pequeño rojo”
- No-deterministica

  “El rojo”

  “El conejo rojo”
- Relacionales

  “El conejo en el sombrero sobre la mesa”

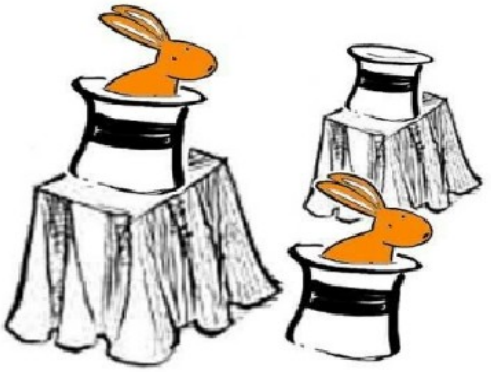




Primera generación de algoritmos

Generaban con el siguiente procedimiento:

- Mantenían una pila de objetos a describir
- Describían el objeto del tope de la pila
- Cuando una relación era usada en la descripción, el objeto relacionado se ponía en el tope de la pila



Desventajas

- Puede tener **Regresión infinita**, el conejo que esta en el sombrero el cual tiene un conejo...
- Produce **una expresión referencial** en cada ejecución (siempre la misma)
- Depende del **orden de las propiedades**
- Agrega las propiedades proposicionales antes de considerar las **relacionales**
- Carece de **sobreespecificación** como los humanos harían dependiendo del contexto

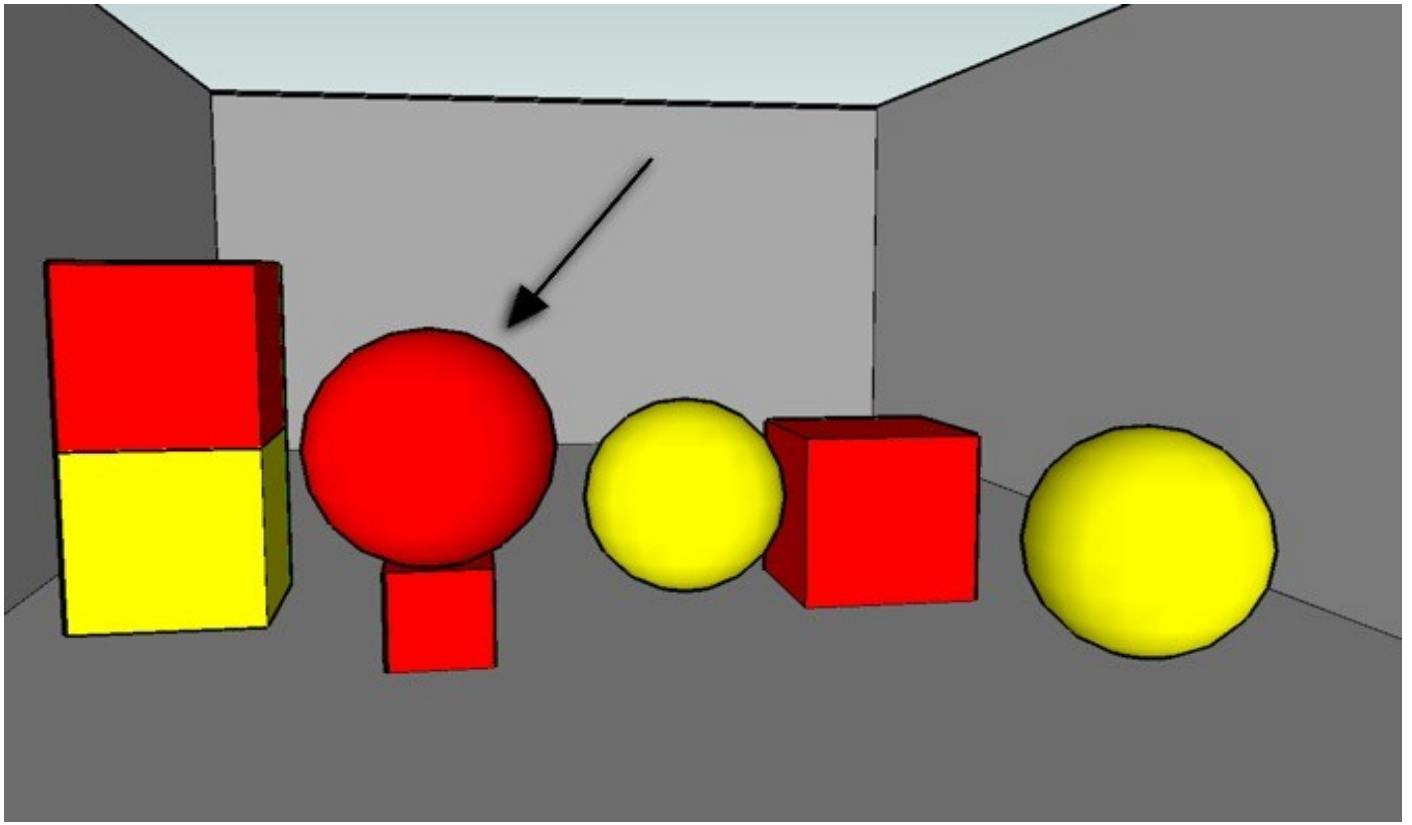
Generación de expresiones referenciales

Aprendiendo “perceptual saliency”

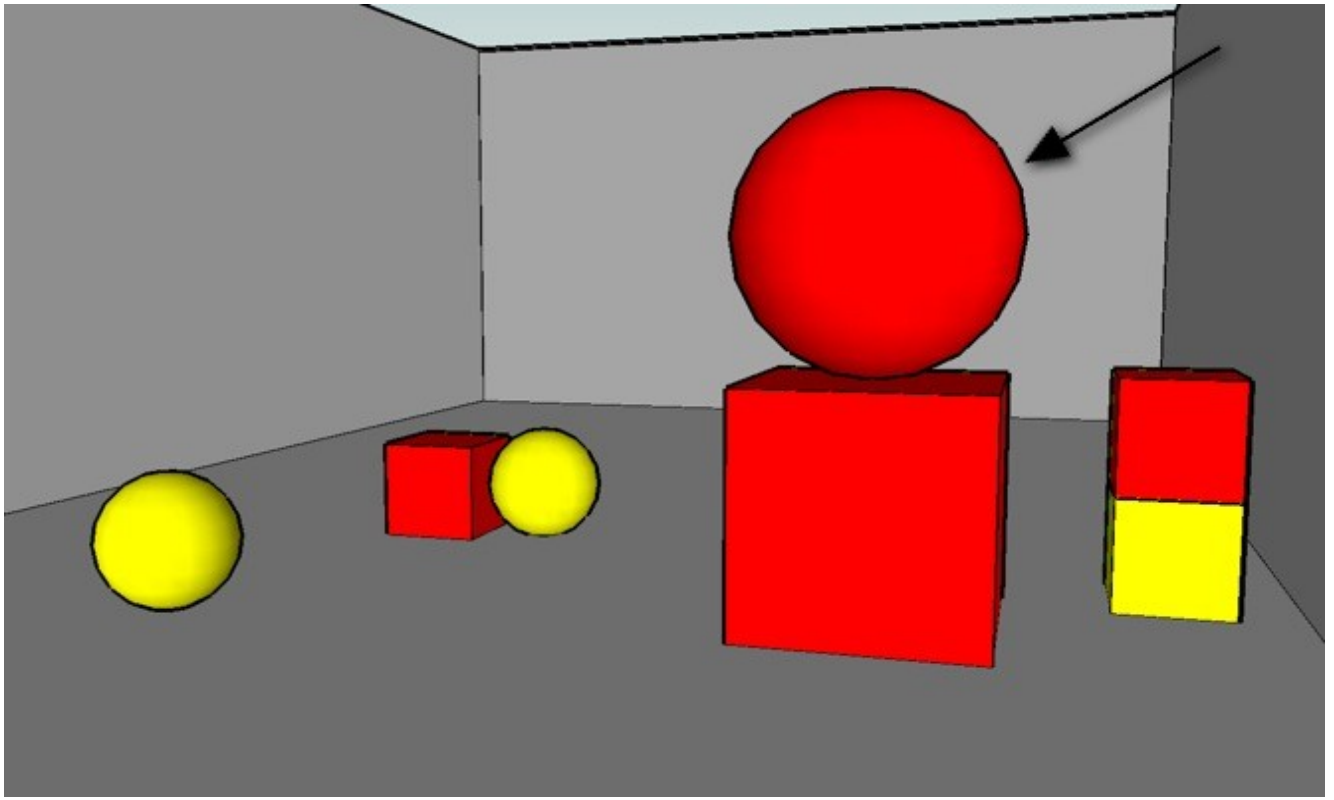
Algoritmo de refinamiento probabilístico

Evaluación y resultados

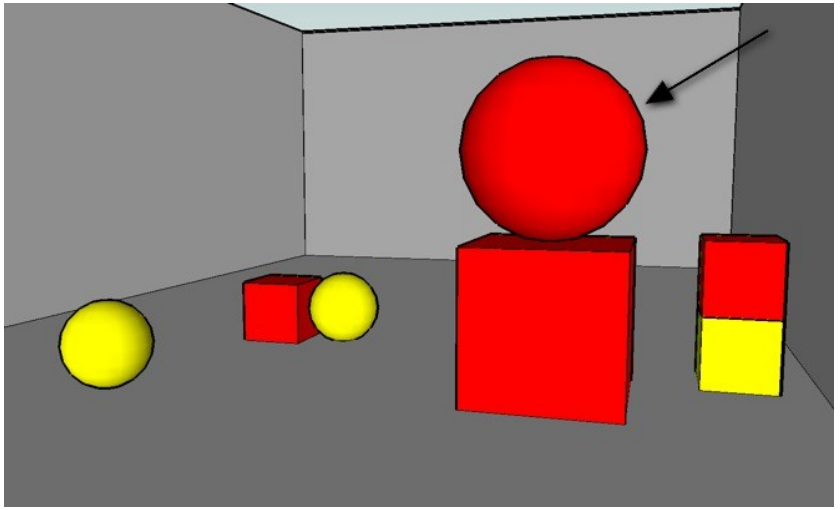
Perceptual saliency

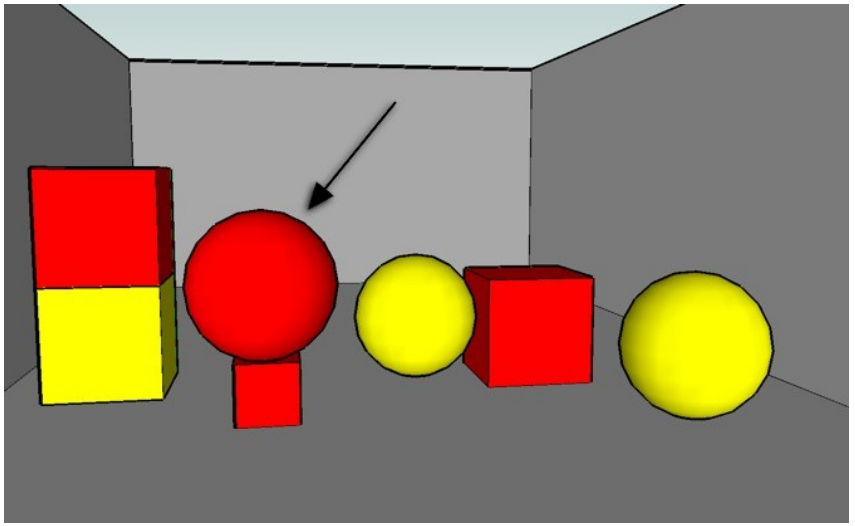


Perceptual saliency



Perceptual saliency





Imagénes del GRE3D7 corpus (Viethen & Dale, 2011)

- Propiedades que son más **salientes** son usualmente más **sobreespecificadas**
- Sobreespecificación depende del **modelo**.
- Tamaño es sobreespecificada:
  - 70% de los casos en la figura de arriba
  - 20% de los casos en la figura de abajo

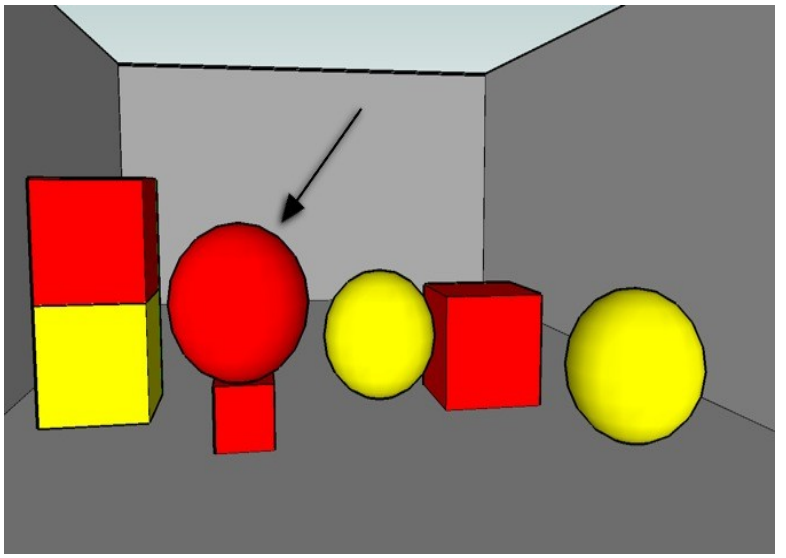
Aprendiendo “perceptual saliencies”

- Tenemos corpus?
- Que corpus tenemos?
- Machine learning sobre atributos
- Que atributos usar?

Aprendiendo “perceptual saliencies”

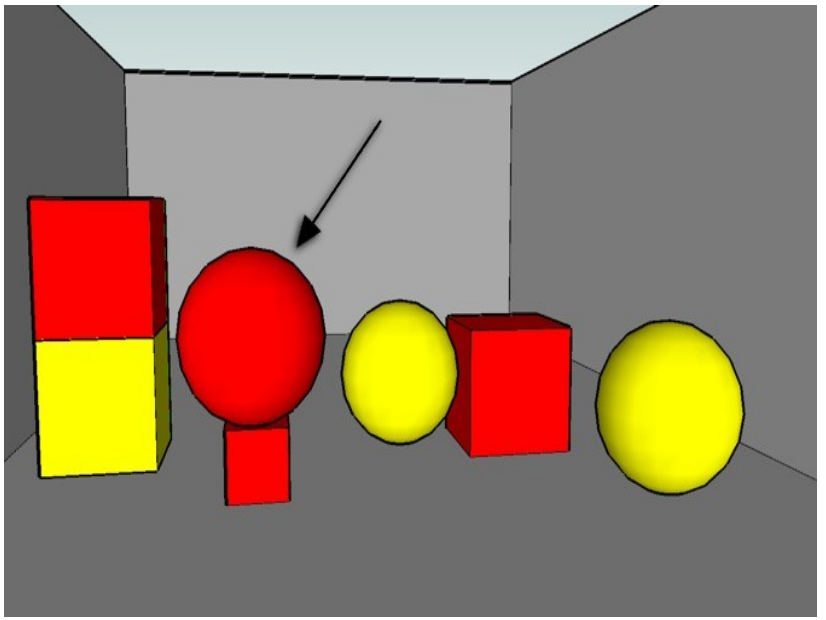
Imágenes del GRE3D7 corpus (Viethen & Dale, 2011)

- Imágenes de 7 objetos en 3D
- 32 imágenes (GRE3D7)
- 140 RE por imagen
- 294 participantes 16 RE c/u



Aprendiendo “perceptual saliencies”

Atributos para aprendizaje automático



- target-tiene(R):

  true si el target esta en R
- landmark-tiene(R):

  true si el landmark tiene R
- discriminacion(R):

  1 / #objetos en el modelo que tienen R

Aprendiendo “perceptual saliencies”

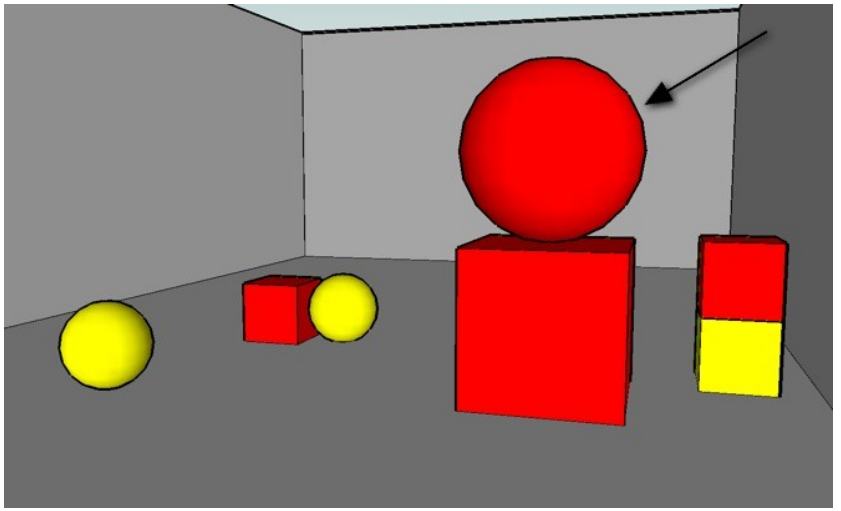
- P_use(R): (# de REs R aparece)/(# REs del corpus).
- Que aprendemos?
  - Una función de regresión lineal sobre los atributos de la escena
  - Ej:

    P_use(big): 0.5 - target-tiene(big) + landmark-tiene(big) + discriminacion(big)

Aprendiendo “perceptual saliencies”

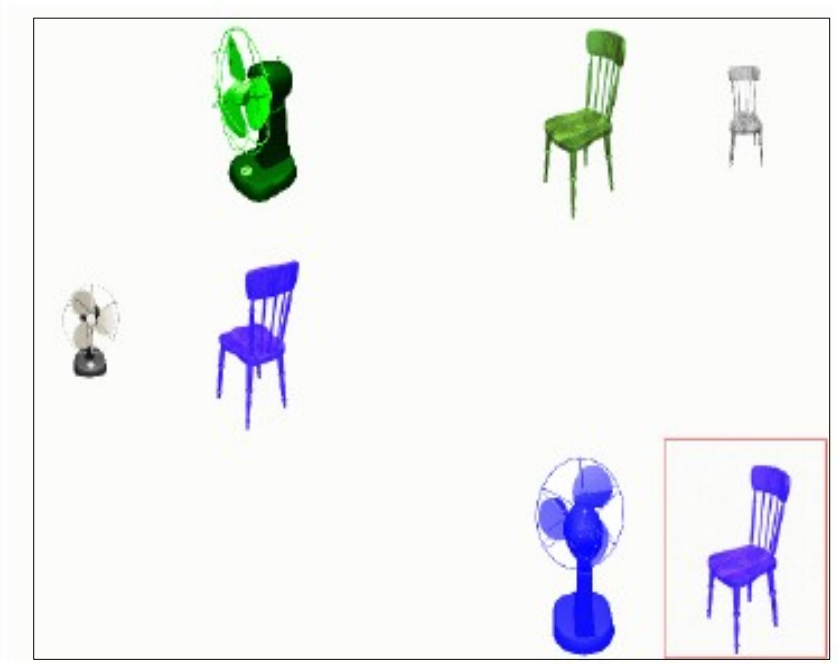
Ej:

P_use(big): 0.5 - target-tiene(big) + landmark-tiene(big) + discriminacion(big)



P_use(big): 0.5 -1 + 1 + 2/7 = 0.78

Aprendiendo del TUNA corpus



Aprendiendo “perceptual saliencies”

Características del TUNA corpus

- 780 RE singulares
- Dominios Furniture y People
- 1 RE por imagen
- Puede contener o no información de localización física en la figura

Aprendiendo “perceptual saliencies”

- Que atributos usamos

  target-tiene(R)

  tiene-locacion

  discriminacion(R)

- Machine learning entrenando con todas las imágenes menos la usada para testear

Nuestro objetivo

Desarrollar un algoritmo

- Que pueda generar RE sobreespecificadas de manera similar a lo que lo que las personas hacen
- Generación no-determinística similar a las personas
- Que evite regresión infinita y que sea eficiente

Generación de expresiones referenciales

Aprendiendo **“perceptual saliency”**

Algoritmo de refinamiento probabilístico

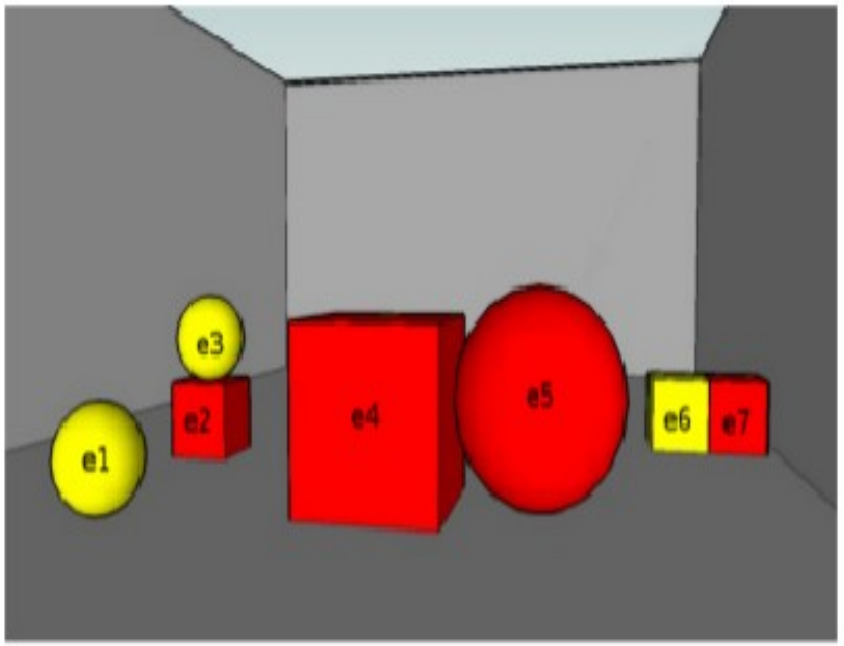
Evaluación y resultados

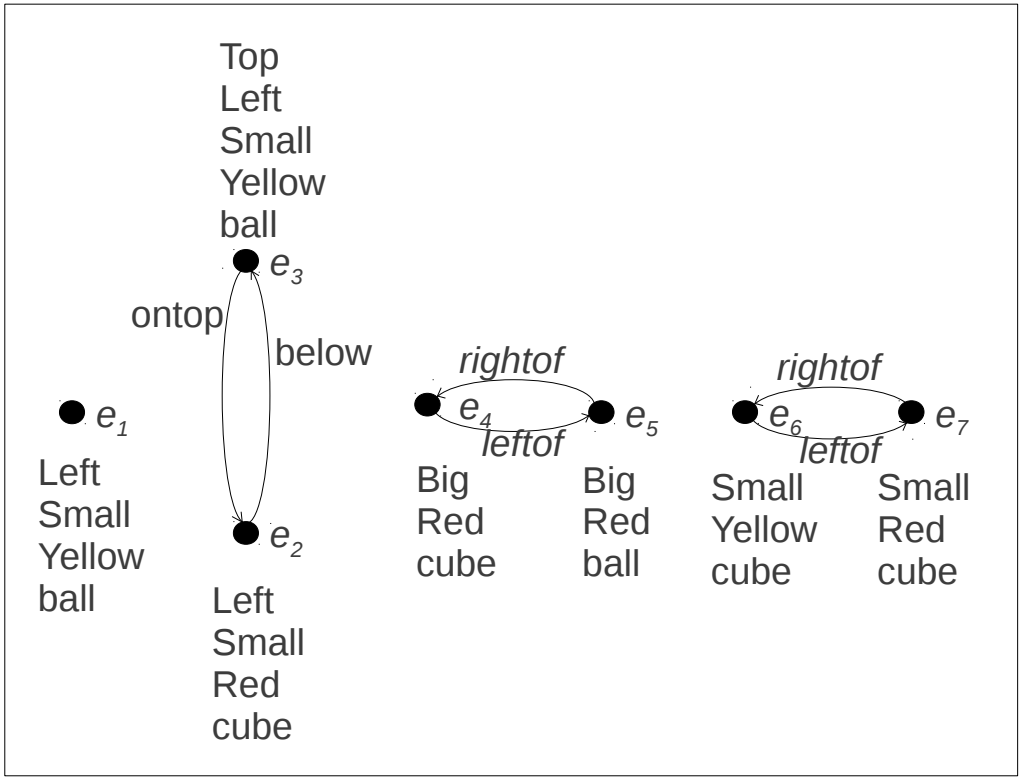
Motivación del algoritmo probabilístico

Nuestro algoritmo esta inspirado en la teoría psicolingüística de producción egocéntrica del lenguaje (Keysar, 1998)

- Fase egocéntrica
  - Guiada por la "perceptual saliency"
  - Proceso heurístico de bajo costo
  - Resulta en sobreespecification
- Fase de refinamiento
  - Identificación unívoca
  - Proceso de refinamiento de alto costo

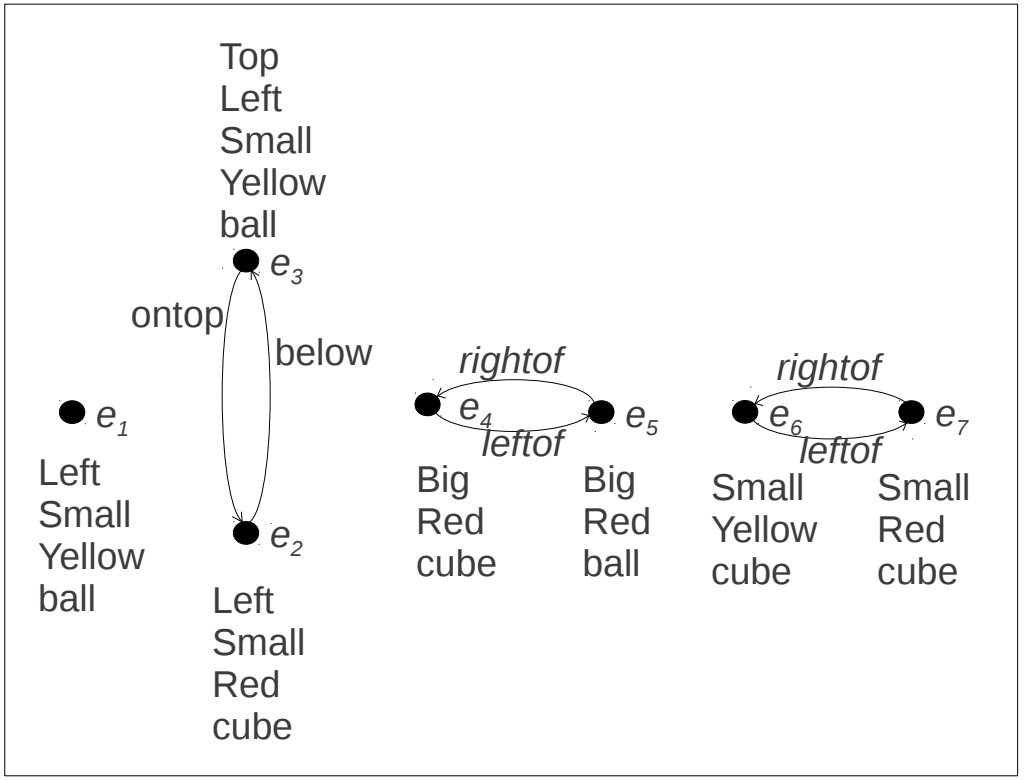
Ejecutando el algoritmo





Ejecutando el algoritmo

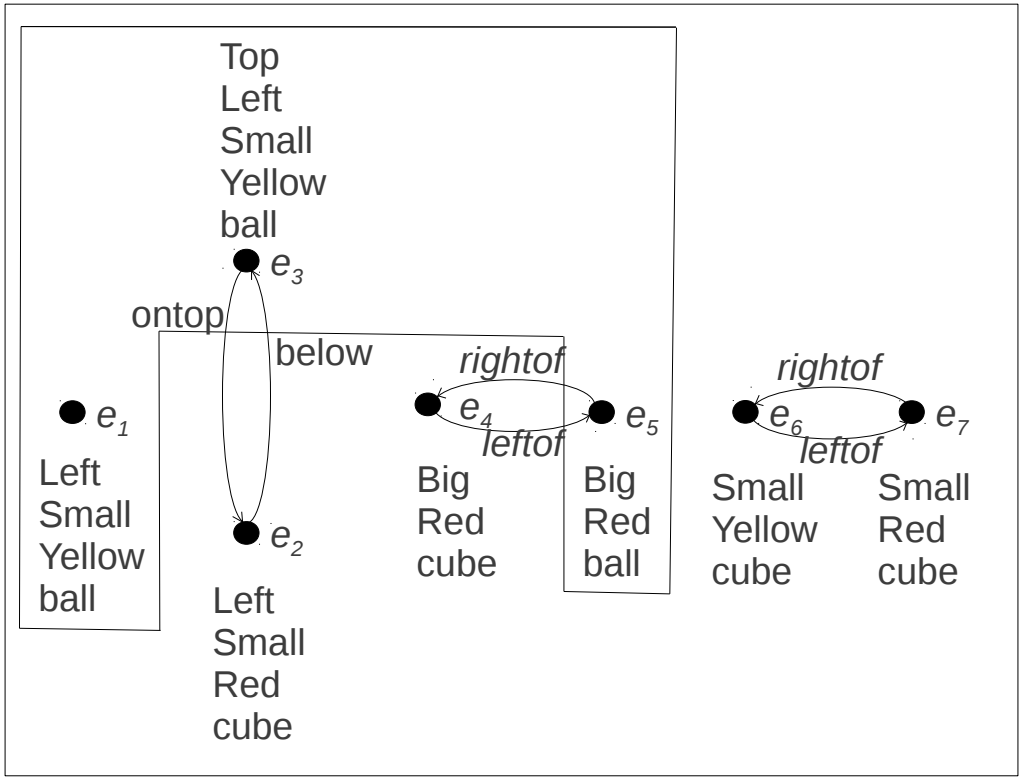
- Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

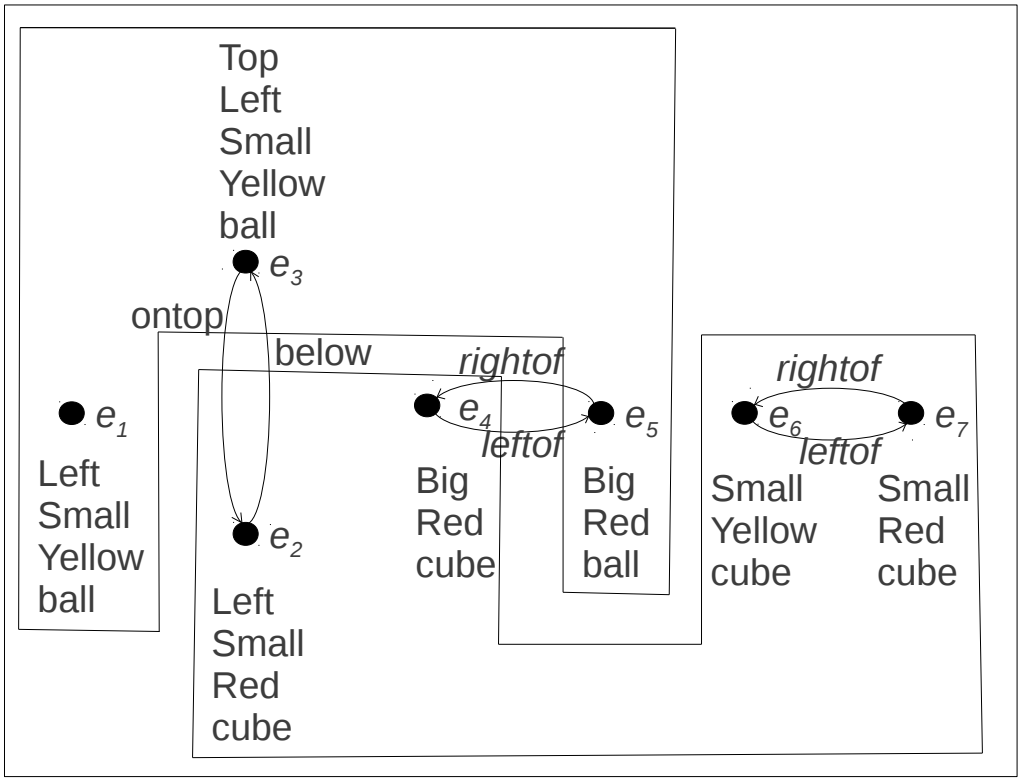
- Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7
- Ball:e_1, e_3, e_5



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

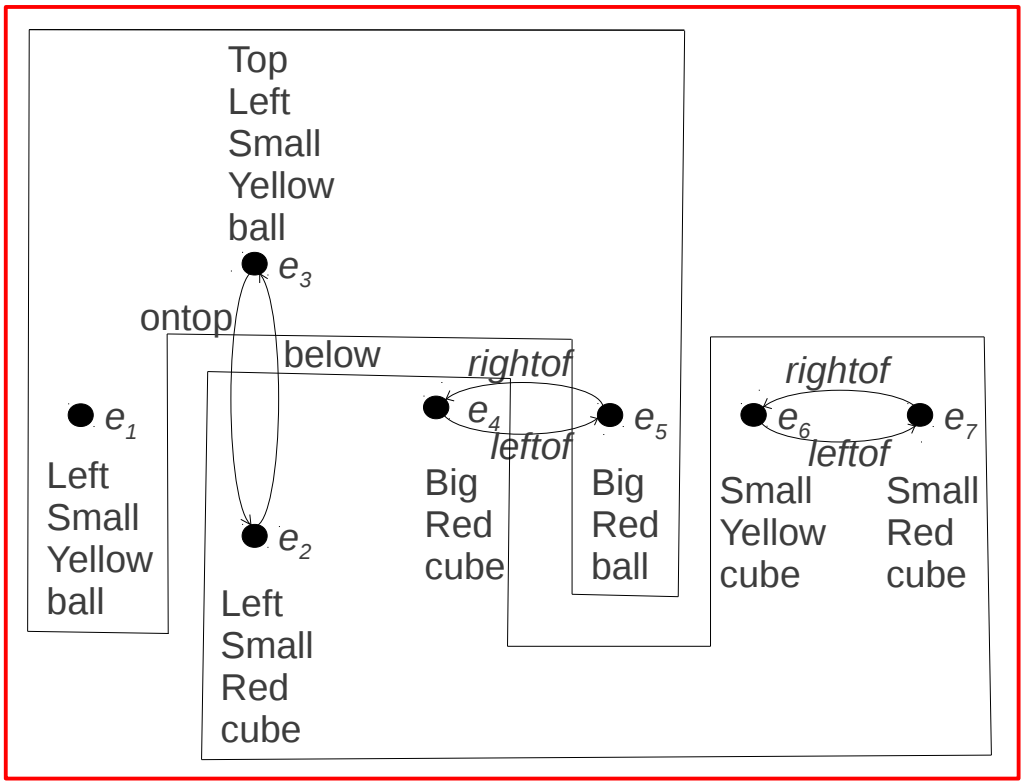
- Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7
- Ball:e_1, e_3, e_5
- Cube: e_2, e_4, e_6, e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

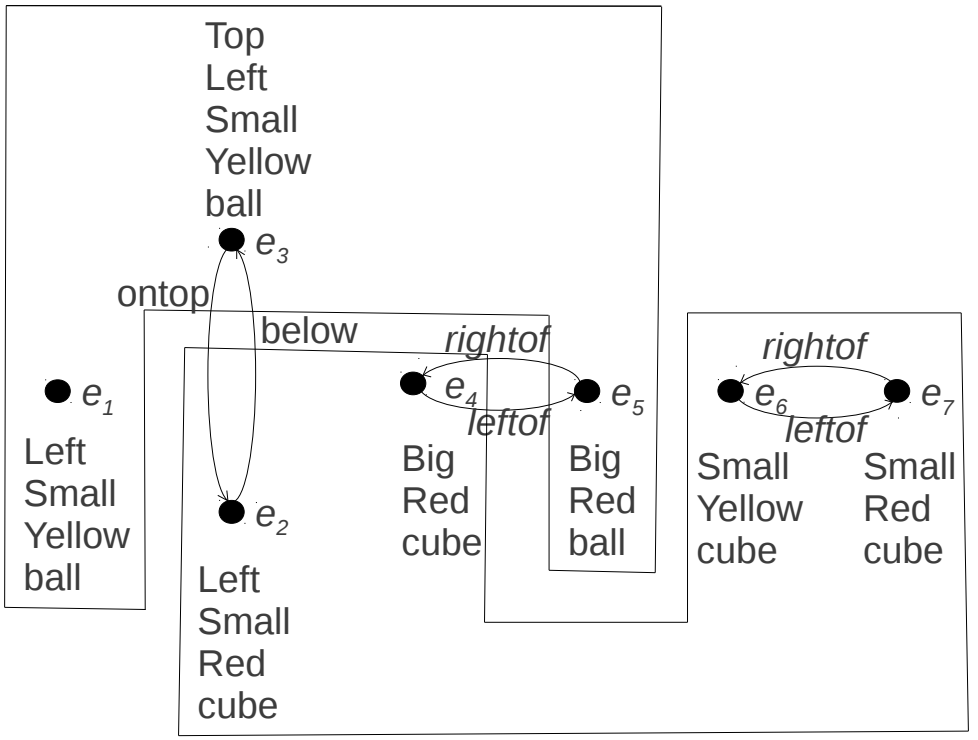
- Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7
- Ball:e_1, e_3, e_5
- Cube: e_2, e_4, e_6, e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

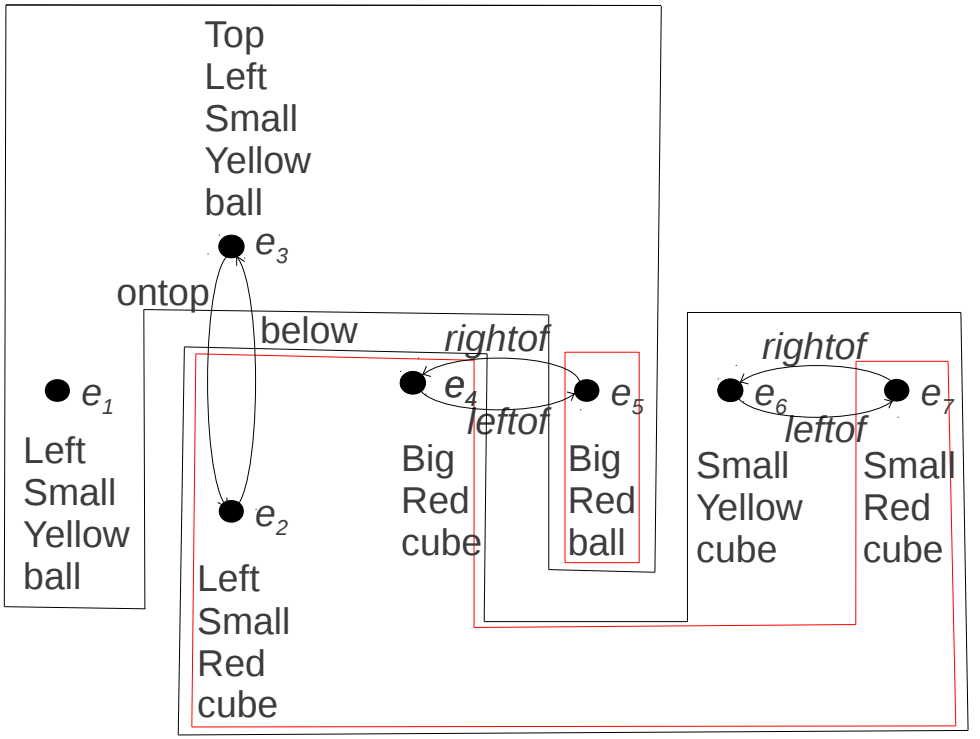
- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- Ball:e_1, e_3, e_5
- Cube: e_2, e_4, e_6, e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

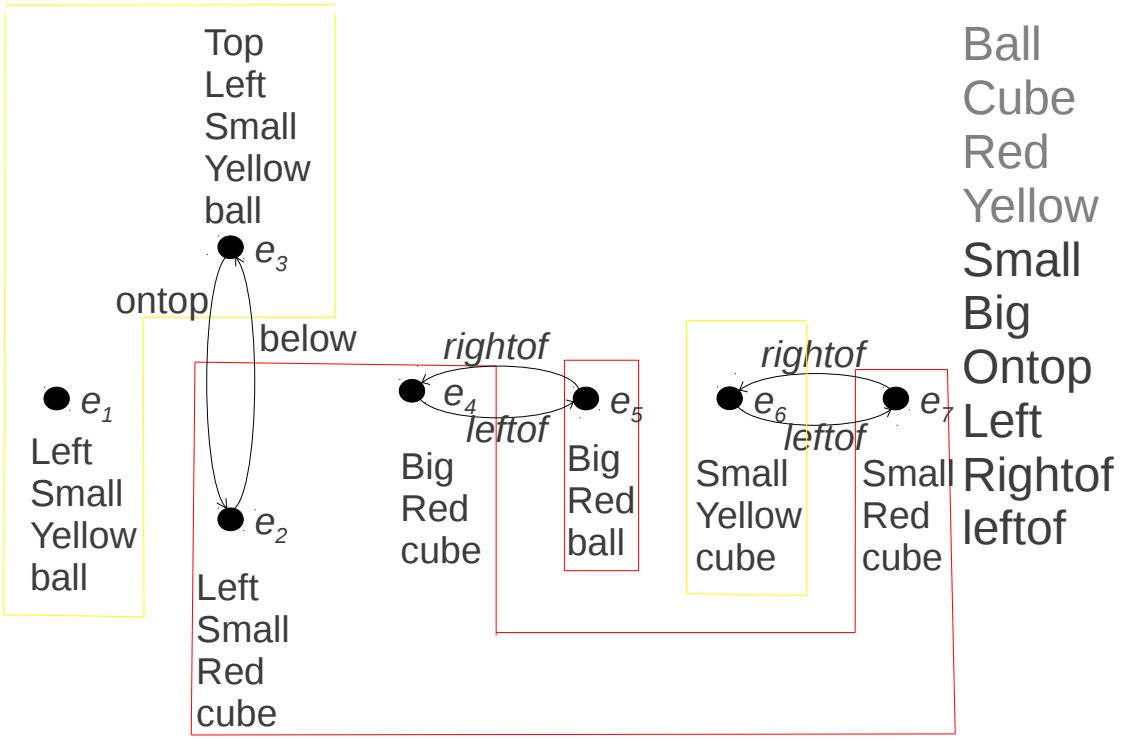
- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- Ball:e_1, e_3, e_5
- Cube: e_2, e_4, e_6, e_7
- Red, ball:e_5
- Red, cube:e_2, e_4, e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

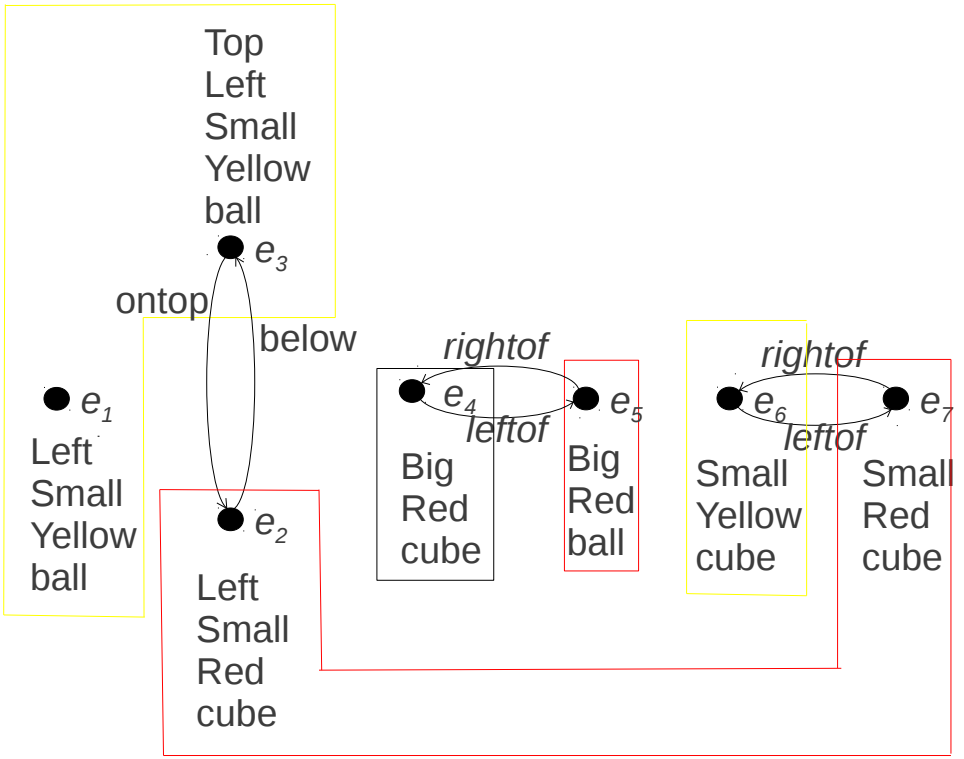
Ejecutando el algoritmo

- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- Red, ball:e_5
- Red, cube:e_2, e_4, e_7
- Yellow ball:e_1 ,e_3
- Yellow cube:e_6



Ejecutando el algoritmo

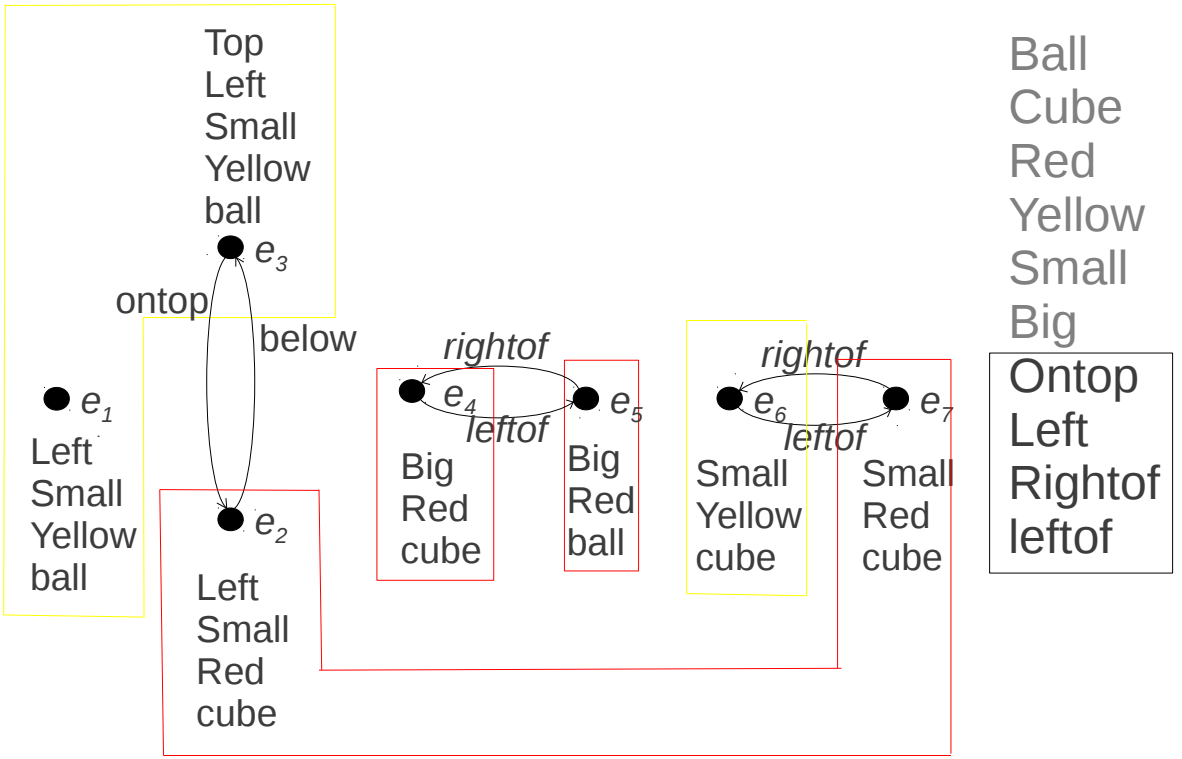
- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

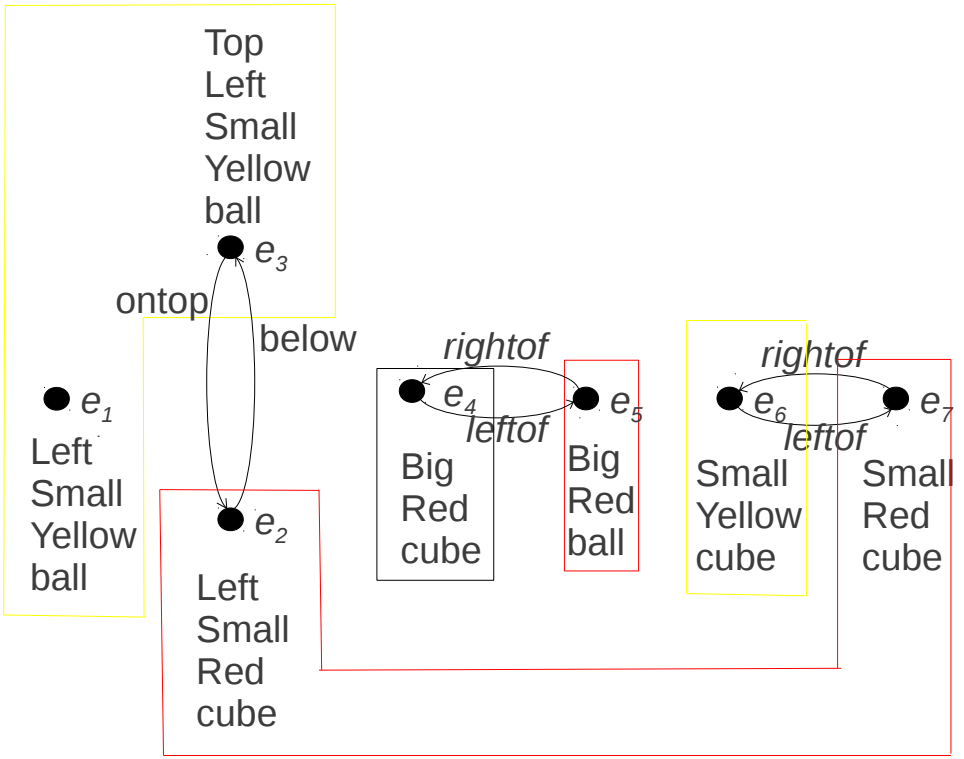
Ejecutando el algoritmo

- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5



Ejecutando el algoritmo

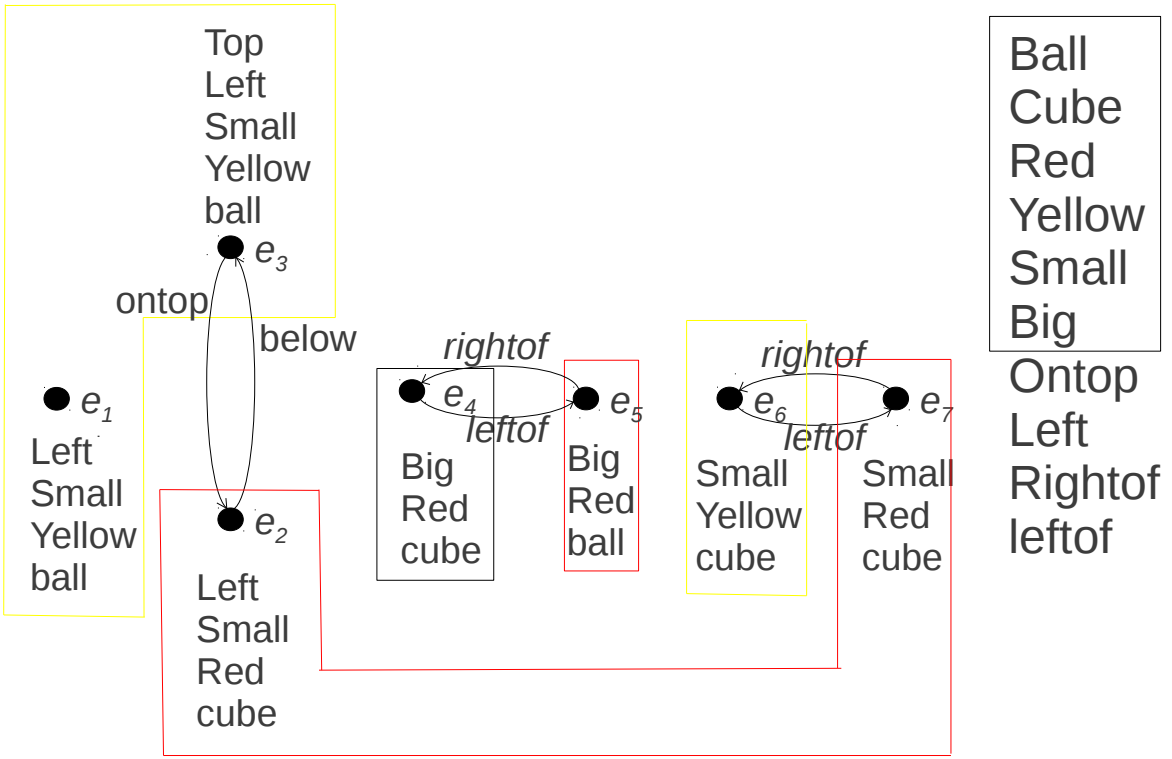
- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

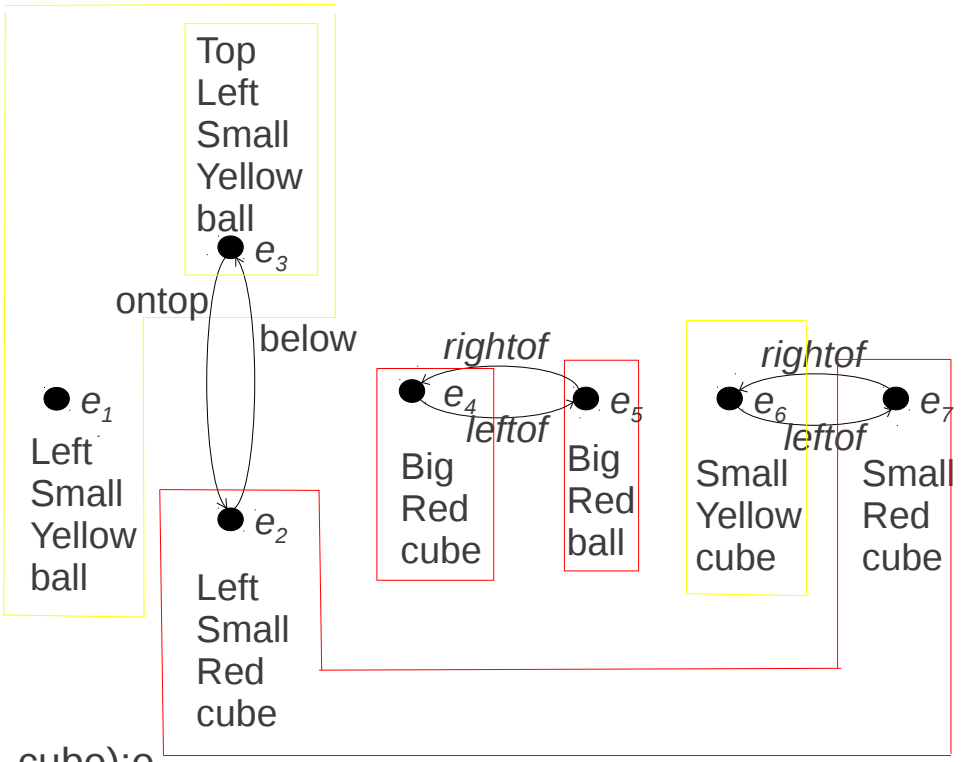
Ejecutando el algoritmo

- ~~Top: e_1, e_2, e_3, e_4, e_5, e_6, e_7~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5



Ejecutando el algoritmo

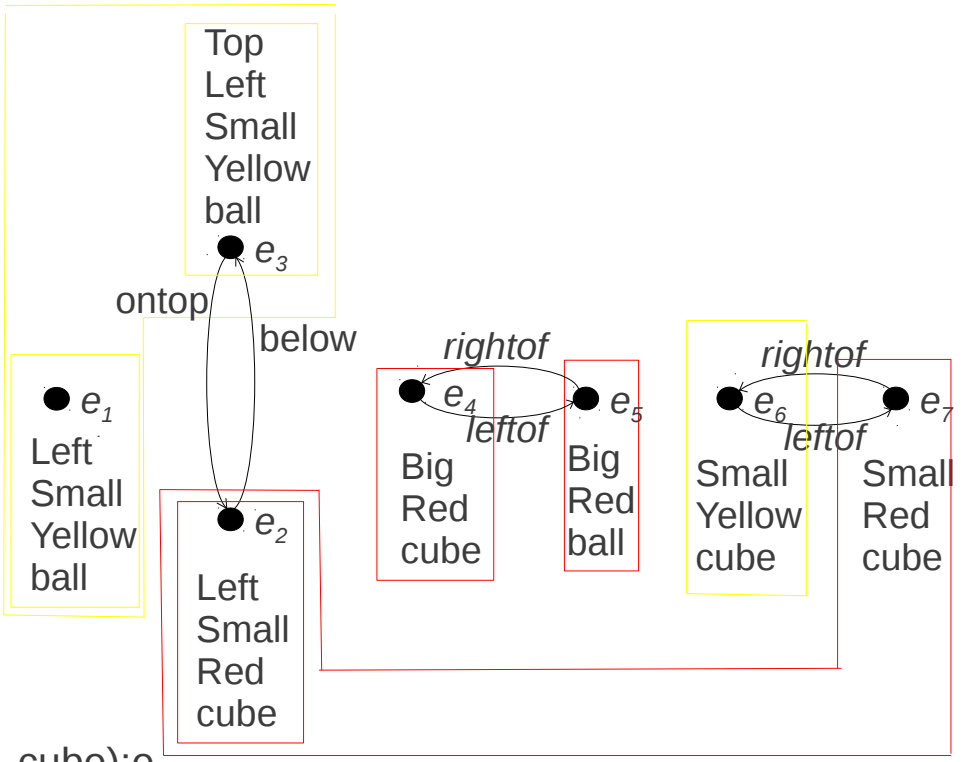
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

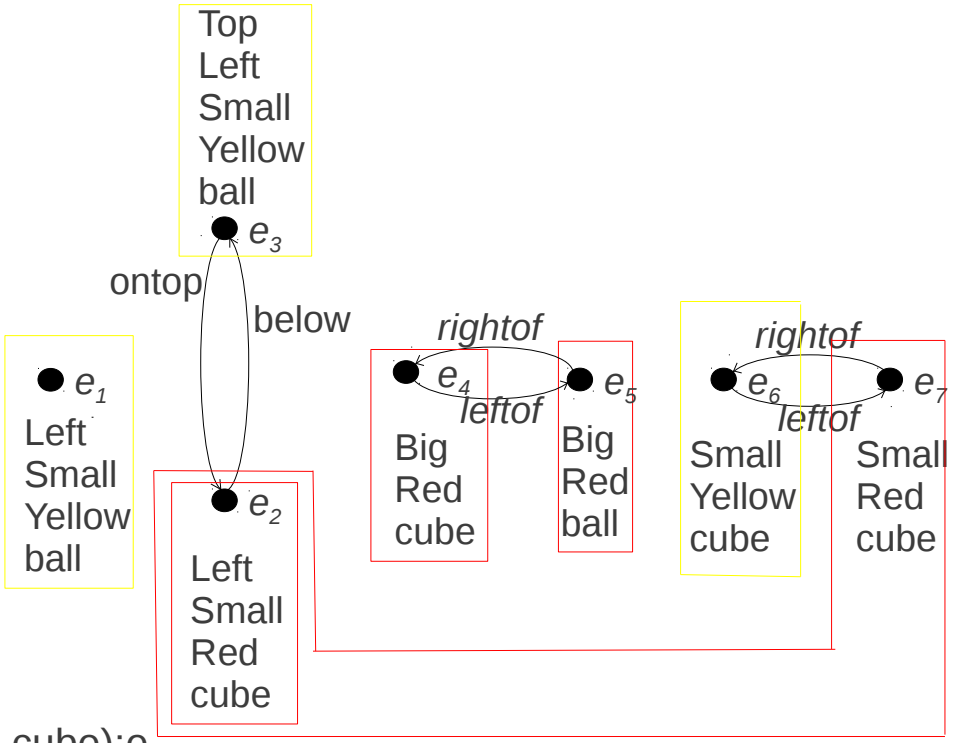
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- Small, yellow, ball: e_1, e_3
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

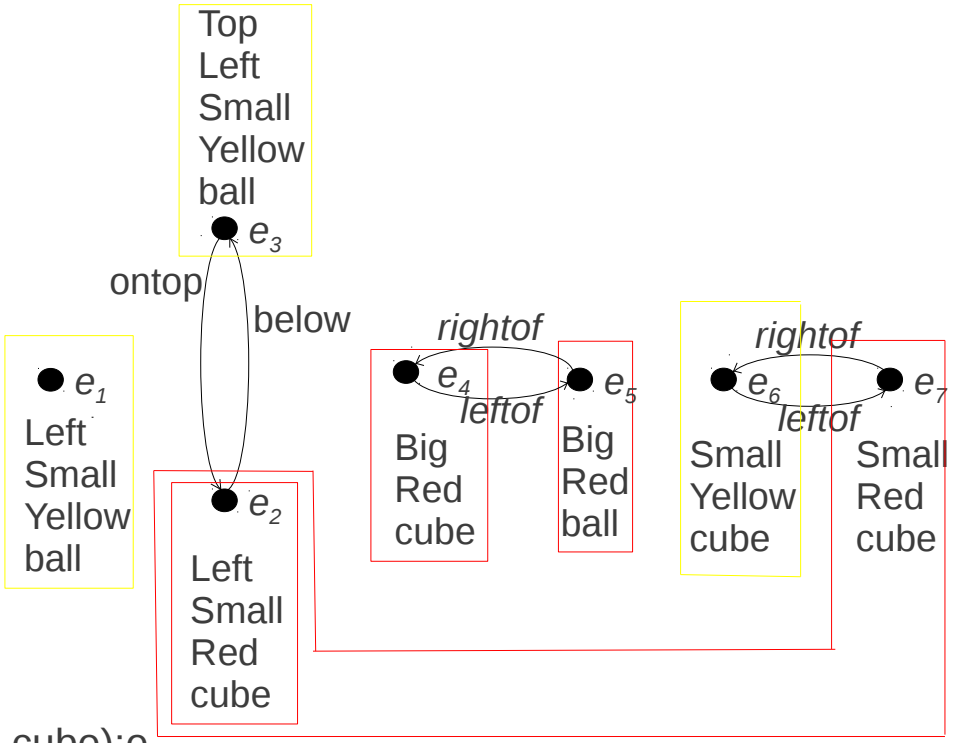
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- ~~Small, yellow, ball: e_1, e_3~~
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

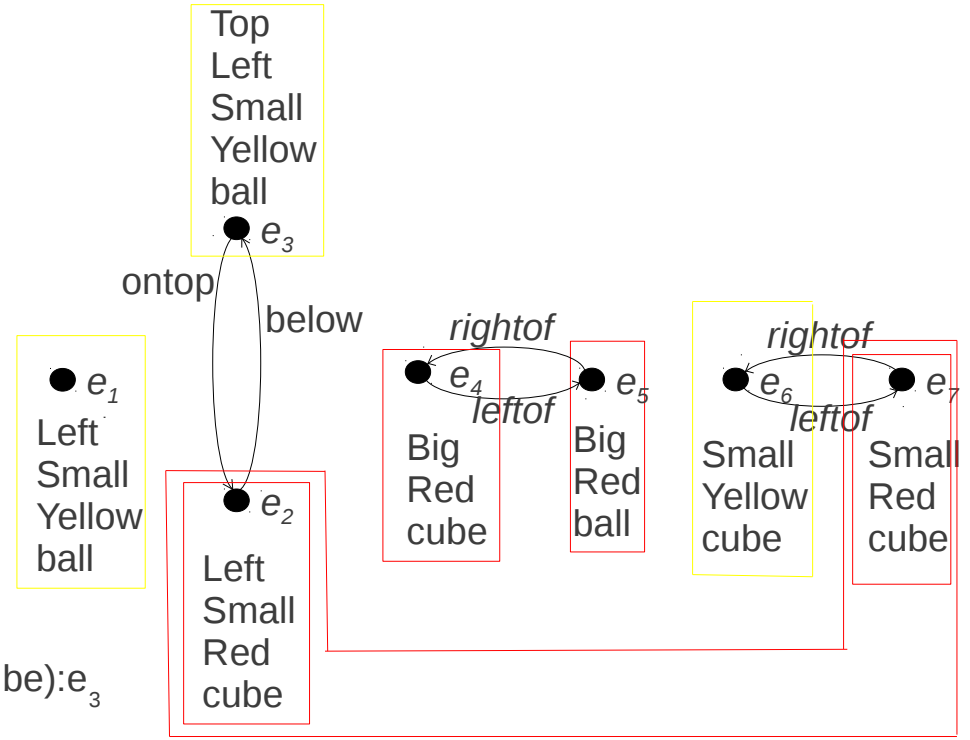
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- ~~Small, yellow, ball: e_1, e_3~~
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

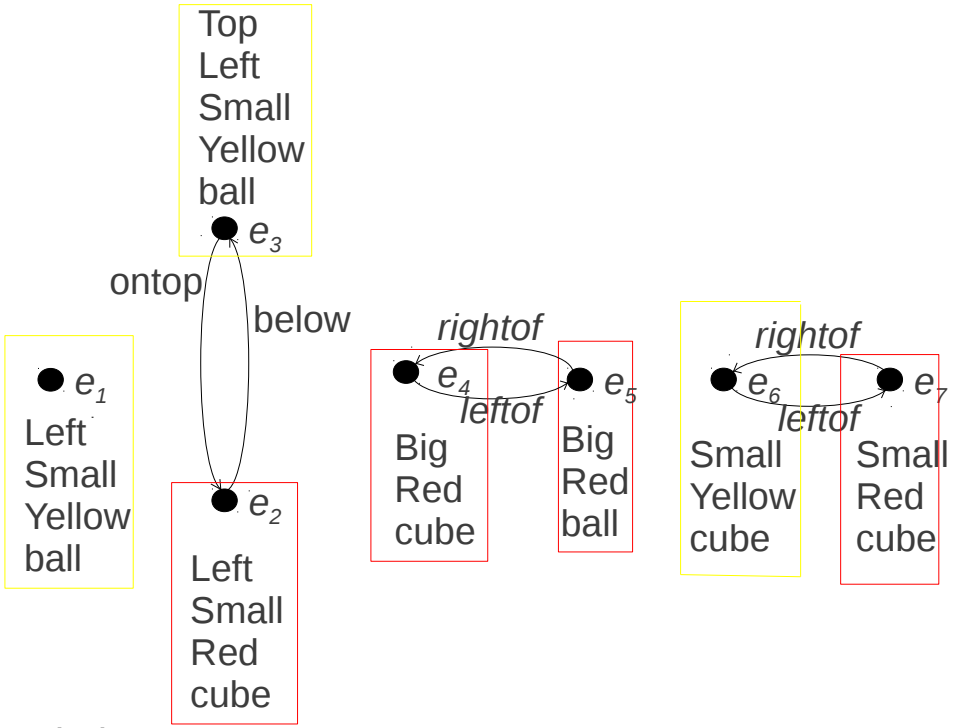
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- ~~Small, yellow, ball: e_1, e_3~~
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2
- Small, red, cube rightof (small, yellow, cube): e_7



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

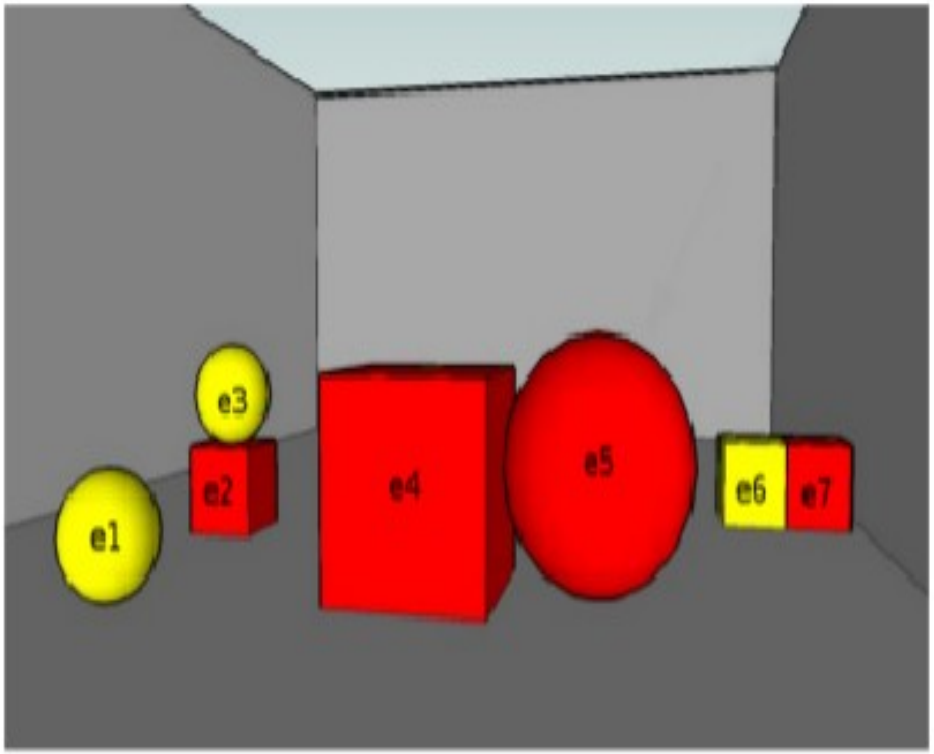
- ~~Top: $e_1, e_2, e_3, e_4, e_5, e_6, e_7$~~
- ~~Ball:e_1, e_3, e_5~~
- ~~Cube: e_2, e_4, e_6, e_7~~
- ~~Red, ball:e_5~~
- ~~Red, cube:e_2, e_4, e_7~~
- ~~Yellow cube:e_6~~
- ~~Small, yellow, ball: e_1, e_3~~
- Small, yellow, cube:e_6
- Small, red, cube: e_2, e_7
- Big, red, cube:e_4
- Big, red ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2



Ball
Cube
Red
Yellow
Small
Big
Ontop
Left
Rightof
leftof

Ejecutando el algoritmo

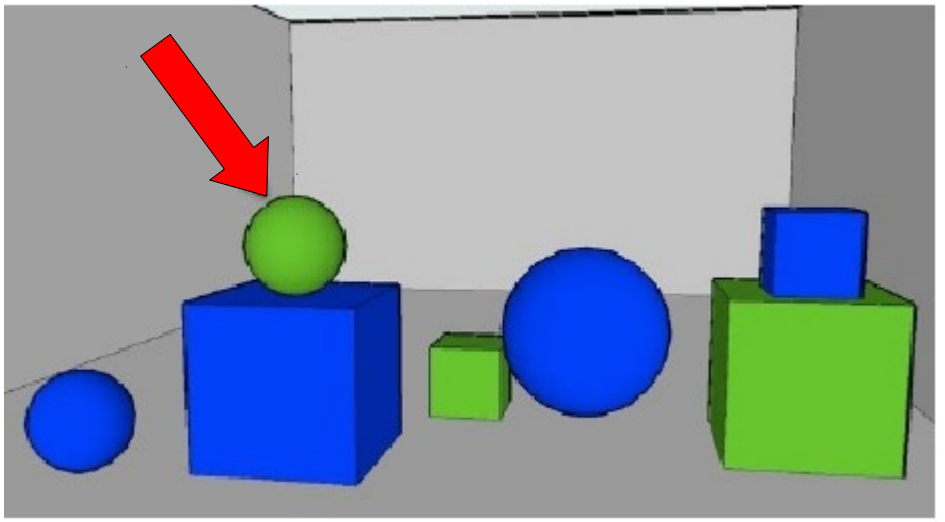
- Small, yellow, cube:e_6
- Big, red, cube:e_4
- Big, red, ball:e_5
- Small, yellow, ball, ontop (small, red, cube):e_3
- Left, small, yellow, ball:e_1
- Left, small, red, cube:e_2
- Small, red, cube rightof (small, yellow, cube): e_7



Comportamiento no-determinístico

ball,green
ball,green,small
ball,green,small,on-top(blue,cube,large)
ball,green,on-top(blue,cube)
ball,green,on-top(blue,cube,large)
ball,green,small,on-top(blue,cube)
ball,on-top(cube)
ball,green,small,on-top(blue,cube,large,left)
ball,small,on-top(cube,large)
ball,green,top
ball,small,on-top(cube)
ball,green,on-top(cube)
ball,front,green
ball,front,green,small
ball,front,top
ball,green,left
ball,top
ball,green,left,small
ball,left,top
ball,small,top

- No-determinístico
- 80% de exactitud respecto para esta imagen:



Ventajas del algoritmo de refinamiento

- Evita regresión infinita: no progresa → termina
- Todas las expresiones referenciales para una escena son producidas al mismo tiempo
- Sobreespecificación y no-determinismo
- Aprende desde un corpus a usar propiedades como ellas fueron usadas en contextos similares
- Maneja descripciones relacionales desde el principio y puede ser fácilmente extendido para tratar con plurales

Generación de expresiones referenciales

Aprendiendo “perceptual saliency”

Algoritmo de refinamiento probabilístico

Evaluación y resultados

Evaluación y resultados

- Evaluación automática
  - Gold standard
- Evaluación manual
  - Jueces
- Evaluación dentro de una actividad
  - Tarea, como evaluar...

Evaluación automática

Métricas usadas

- Coeficiente DICE(A,B) = (2 X |A int B|)/(|A|+|B|)
- MASI score: Delta X |A int B|/|A union B|
  con Delta de 0 a 1
- Accuracy: Porcentaje de matching exacto

Evaluación automática

- Métricas usadas

	Dice	MASI	ACCURACY
GRAPH system, Furniture domain	.80	.59	.48
GRAPH system, People domain	.72	.48	.28
Our system, Furniture domain (top 1)	.80	.60	.47
Our system, People domain (top 1)	.65	.37	.19
Our system, Furniture domain (top 20)	.87	.75	.65
Our system, People domain (top 20)	.81	.68	.60

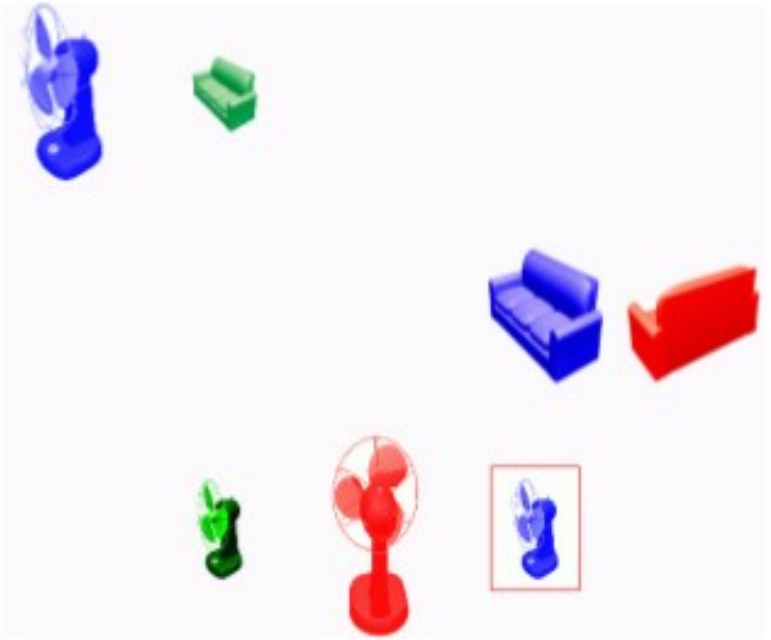
Evaluación manual TUNA corpus

92% de las REs son consideradas mejores o iguales por un juez humano

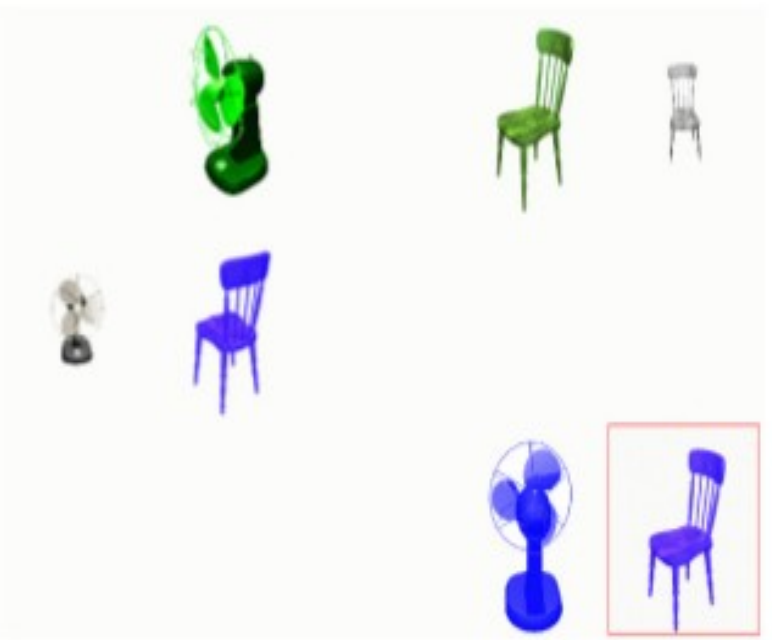
	Furniture domain	People domain	Weighted mean
system equal to human	.46	.19	.33
system better by 2 judges	.29	.24	.27
system better by 1 or 2 judges	.51	.68	.59
system worse by 2 judges	.03	.13	.08
system equal or better by 2 judges	.75	.43	.60
system equal or better by 1 judge	.97	.87	.92

Evaluando sobre TUNA corpus

Ambos jueces prefirieron la RE generada por el sistema



Humano: blue fan
Sistema: small blue fan



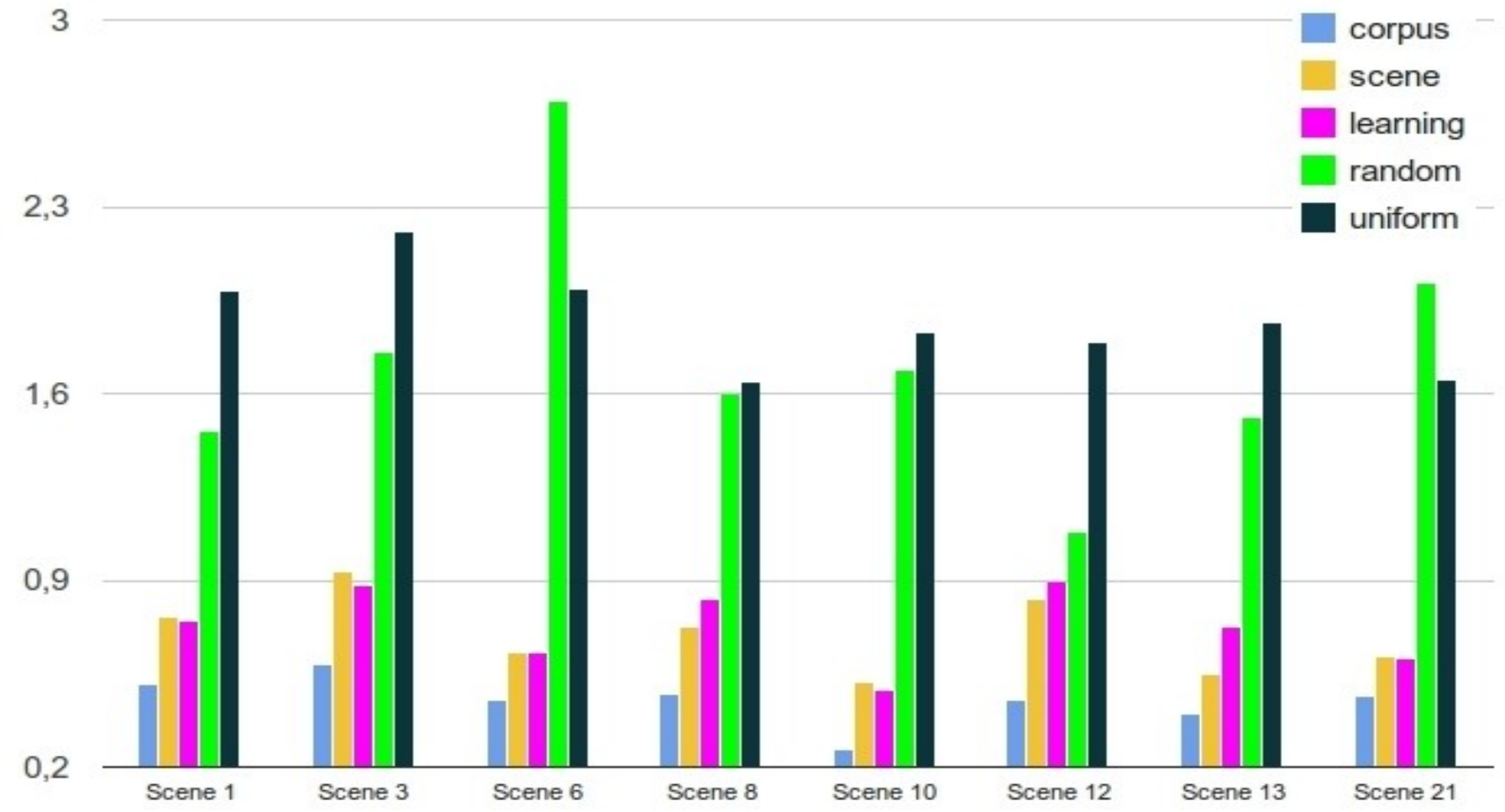
Humano: blue frontal chair
Sistema: the blue chair in the bottom

Evaluación sobre el GRE3D7

Comparando distribuciones de frecuencias

- Corpus (nuestro gold standar)
- Escena (haciendo trampa)
- Machine learning (función de regresión lineal)
- Uniforme (a cada propiedad la misma probabilidad)
- Random (probabilidad aleatoria para las propiedades)

Evaluación sobre el GRE3D7



Conclusiones

- Extendimos algoritmo de refinamiento con sobreespecificación y no-determinismo guiado por probabilidades aprendidas desde un corpus
- Evaluamos sobre dos corpus: RE del sistema fueron juzgadas iguales o mejores que las humanas en 92% de los casos (por un juez)
- El diseño de nuestro algoritmo esta inspirado en la teoría psicolingüística de la producción egocéntrica del lenguaje

Preguntas?