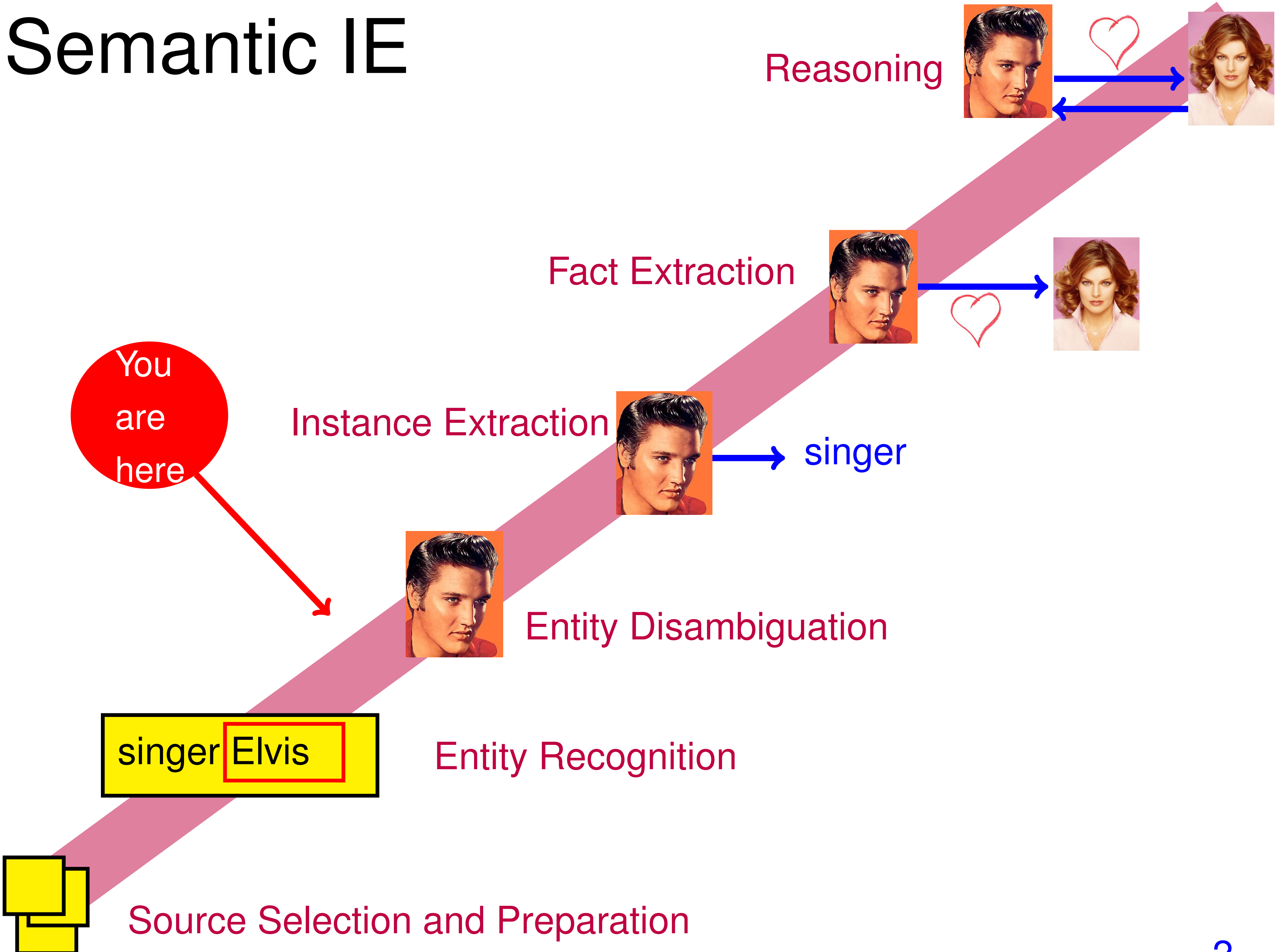
Evaluation

Fabian M. Suchanek

Semantic IE



Reasoning
Fact Extraction
You are here
Instance Extraction
singer
Entity Disambiguation
singer Elvis
Entity Recognition
Source Selection and Preparation

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

Pixelpanzer.de

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

   4 matches (1 wrong)

Pixelpanzer.de

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

   4 matches (1 wrong)

2. [A-Z][a-z]+ [A-Z][a-z]+

Pixelpanzer.de

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the
father of Bart Simpson and Lisa Simpson.
The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

   4 matches (1 wrong)

2. [A-Z][a-z]+ [A-Z][a-z]+

   5 matches (2 wrong)

Pixelpanzer.de

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

   4 matches (1 wrong)

2. [A-Z][a-z]+ [A-Z][a-z]+

   5 matches (2 wrong)

3. Homer Simpson

Pixelpanzer.de

Detect members of the Simpsons

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

1. [A-Z][a-z]+ Simpson

   4 matches (1 wrong)

2. [A-Z][a-z]+ [A-Z][a-z]+

   5 matches (2 wrong)

3. Homer Simpson

   Pixelpanzer.de

   1 match

Def: Gold Standard

The gold standard (also: ground truth) for an IE task is the set of desired results of the task on a given corpus.

Task: Detect Simpson members

Corpus:

in The Simpsons, Homer Simpson is the father of Bart Simpson and Lisa Simpson. The M above his ear is for Matt Groening.

Gold Standard:

{Homer Simpson, Bart Simpson, Lisa Simpson}

Def: Precision

The precision of an IE algorithm is the ratio of its outputs that are in the respective gold standard.

$$prec = \frac{|Output \cap GStandard|}{|Output|}$$

Output: {Homer, Bart, Groening}

G.Standard: {Homer, Bart, Lisa, Marge}

Def: Precision

The precision of an IE algorithm is the ratio of its outputs that are in the respective gold standard.

$$prec = \frac{|Output \cap GStandard|}{|Output|}$$

Output: {Homer, Bart, Groening}

$\checkmark$

G.Standard: {Homer, Bart, Lisa, Marge}

Def: Precision

The precision of an IE algorithm is the ratio of its outputs that are in the respective gold standard.

$$prec = \frac{|Output \cap GStandard|}{|Output|}$$

Output: {Homer, Bart, Groening}

$\surd$ $\surd$

G.Standard: {Homer, Bart, Lisa, Marge}

Def: Precision

The precision of an IE algorithm is the ratio of its outputs that are in the respective gold standard.

$$prec = \frac{|Output \cap GStandard|}{|Output|}$$

Output: {Homer, Bart, Groening}

$\checkmark$ $\checkmark$ $\times$

G.Standard: {Homer, Bart, Lisa, Marge}

Def: Precision

The precision of an IE algorithm is the ratio of its outputs that are in the respective gold standard.

$$prec = \frac{|Output \cap GStandard|}{|Output|}$$

Output: {Homer, Bart, Groening}

$\checkmark$ $\checkmark$ ✗

G.Standard: {Homer, Bart, Lisa, Marge}

=> Precision: 2/3 = 66%

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: $\{$Homer, Bart, Groening$\}$

G.Standard: $\{$Homer, Bart, Lisa, Marge$\}$

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: {Homer, Bart, Groening}

G.Standard: {Homer, Bart, Lisa, Marge}

$\checkmark$

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: $\{$Homer, Bart, Groening$\}$

G.Standard: $\{$Homer, Bart, Lisa, Marge$\}$

$\checkmark \quad \checkmark$

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: {Homer, Bart, Groening}

G.Standard: {Homer, Bart, Lisa, Marge}

$\checkmark$ $\checkmark$ $\times$

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: {Homer, Bart, Groening}

G.Standard: {Homer, Bart, Lisa, Marge}

$\checkmark$ $\checkmark$ $\times$ $\times$

Def: Recall

The recall (also: sensitivity, true positive rate, hit rate) of an IE algorithm is the ratio of the gold standard that is output.

$$rec = \frac{|Output \cap GStandard|}{|GStandard|}$$

Output: {Homer, Bart, Groening}

G.Standard: {Homer, Bart, Lisa, Marge}

✓ ✓ ✗ ✗

=> Recall: 2/4 = 50%

>F1

Precision-Recall-Tradeoff

It is very hard to get both good precision and good recall.
Algorithms usually allowing varying one at the expense of the other (e.g., by choosing different threshold values). This usually yields:

Def: F1

To trade off precision and recall, we could use the average:

Gold Standard: {Homer, Bart, Lisa, Snowball_4, ..., Snowball_100}

Output: {Homer Simpson}

Def: F1

To trade off precision and recall, we could use the average:

Gold Standard: {Homer, Bart, Lisa, Snowball_4, ..., Snowball_100}

Output: {Homer Simpson}

Precision: 1/1=100%, Recall: 1/100=1%

Average: (100%+1%)/2=50%

Outputting just a single result already gives a score of 50%!

The F1 measure is the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Precision: 1/1=100%, Recall: 1/100=1%

F1: $2 \times 100\% \times 1\% / (100\% + 1\%) = 2\%$

Task: Precision & Recall

What is the algorithm output, the gold standard, the precision and the recall in the following cases?

1. Nostradamus predicts a trip to the moon for every century from the 15th to the 20th incl.

O{15,16,17,18,19,20} GS{20}

2. The weather forecast predicts that the next 3 days will be sunny. It does not say anything about the 2 days that follow. In reality, it is sunny during all 5 days.

O{1,2,3} GS{1,2,3,4,5}

Songs{e1,e2,…,e90,01,…,o10}
O{e1,e2,…,e15,o1,…,o5}
GS{e1,e2,…,e90}

3. On Elvis Radio™ , 90% of the songs are by Elvis. An algorithm learns to detect Elvis songs. Out of 100 songs on Elvis Radio, the algorithm says that 20 are by Elvis (and says nothing about the other 80). Out of these 20 songs, 15 were by Elvis and 5 were not.

4. How can you improve the algorithm?

O = Songs

>ROC

Imbalanced classes

Population:	{Snowball_1,..., Snowball_99, Snowball_100}
Gold Standard:	{Snowball_1,..., Snowball_99}
Output:	{Snowball_1,..., Snowball_99, Snowball_100}

Def: Problem of imbalanced classes

Population: {Snowball_1,..., Snowball_99, Snowball_100}

Gold Standard: {Snowball_1,..., Snowball_99}

Output: {Snowball_1,..., Snowball_99, Snowball_100}

Precision: 99/100=99%

Recall: 99/99=100%

If there are very few negatives, just outputting all elements gives great scores.

The problem of imbalanced classes appears when only very few of the items of the population are not in the gold standard: An approach that outputs the entire population has a very high precison and a perfect recall.

>ROC

Def: Confusion Matrix

Population: {Snowball_1,..., Snowball_99, Snowball_100}

Gold Standard: {Snowball_1,..., Snowball_99}

Output: {Snowball_1,..., Snowball_99, Snowball_100}

The confusion matrix for the output of an algorithm looks as follows:

Items of the population that are not in the gold standard

		Gold standard		
		Positive	Negative	$\sum$
Output	Positive	True Positives 15	False Positives 5	Predicted Positives 20
	Negative	False Negatives 75	True Negatives 5	Predicted Negatives 80
	$\sum$	(Gold) Positives 90	(Gold) Negatives 10	

Items of the population that are not output

"Negative" because it was not output, "True" because that was correct.

Def: Confusion Matrix

Population:	{Snowball_1,..., Snowball_99, Snowball_100}
Gold Standard:	{Snowball_1,..., Snowball_99}
Output:	{Snowball_1,..., Snowball_99, Snowball_100}

The confusion matrix for the output of an algorithm looks as follows:

1 item was output as positive, but was negative in the gold standard

		Gold standard: Positive	Gold standard: Negative	
Output	Positive	99	1	100
	Negative	0	0	0
		99	1	

Precision = true positives / predicted positives = 99/100 = 99%

Recall = true positives / gold positives = 99/99 = 100%

>ROC

Confusion with confusion matrixes

A confusion matrix does not always make sense in an information extraction scenario:

Population:	{H, Ho, Hom, ..., o, om, ome, ..., r Sim, r Simps, ...}
Gold Standard:	{Homer}
Output:	{Homer}

		Gold standard: Positive	Gold standard: Negative	
Output	Positive	1	39462440205	39462440206
	Negative	0	0	

A confusion matrix makes sense only when the population is limited (e.g., in classification tasks)!

Back to our problem

Population: {Snowball_1,..., Snowball_99, Snowball_100}

Gold Standard: {Snowball_1,..., Snowball_99}

Output: {Snowball_1,..., Snowball_99, Snowball_100}

		Gold standard	
		Positive	Negative
Output	Positive	99	1
	Negative	0	0

The problem is that the algorithm did not catch the negatives,
it has a “low recall” on the negatives.

Def: True Negative Rate & FPR

Population:	{Snowball_1,..., Snowball_99, Snowball_100}
Gold Standard:	{Snowball_1,..., Snowball_99}
Output:	{Snowball_1,..., Snowball_99, Snowball_100}

The true negative rate (also: TNR, specificity, selectivity) is the ratio of negatives that are output as negatives (= the recall on the negatives):

TNR = true negatives / gold negatives = 0 / 1 = 0%

		Positive	Negative
Output	Positive	99	1
	Negative	0	0

The False Positive Rate (also: FPR, fall-out) is 1-TNR.

>ROC

TNR & Precision

Population: {Snowball_1,..., Snowball_99, Snowball_100}

Gold Standard: {Snowball_1,..., Snowball_99}

Output: {Snowball_1,..., Snowball_99, Snowball_100}

Precision: 99/100=99%

Recall: 99/99=100%

TNR: 0/1=0%

TNR and precision both measure the “correctness” of the output.

Precision:

- measures wrt. the output
- suffers from imbalanced classes
- works if population is infinite
  (e.g., set of all extractable entities)

TNR:

- measures wrt. the population
- guards against imbalance
- works if population is limited
  (e.g., in classification)

>ROC

Informedness

Population: {Snowball_1,..., Snowball_99, Snowball_100}

Gold Standard: {Snowball_1,..., Snowball_99}

Output: {Snowball_1,..., Snowball_99, Snowball_100}

Precision: 99/100=99%

Recall: 99/99=100%

TNR: 0/1=0%

The informedness (also: Youden's J statistic, Youden's index) combines TNR and Recall as follows:

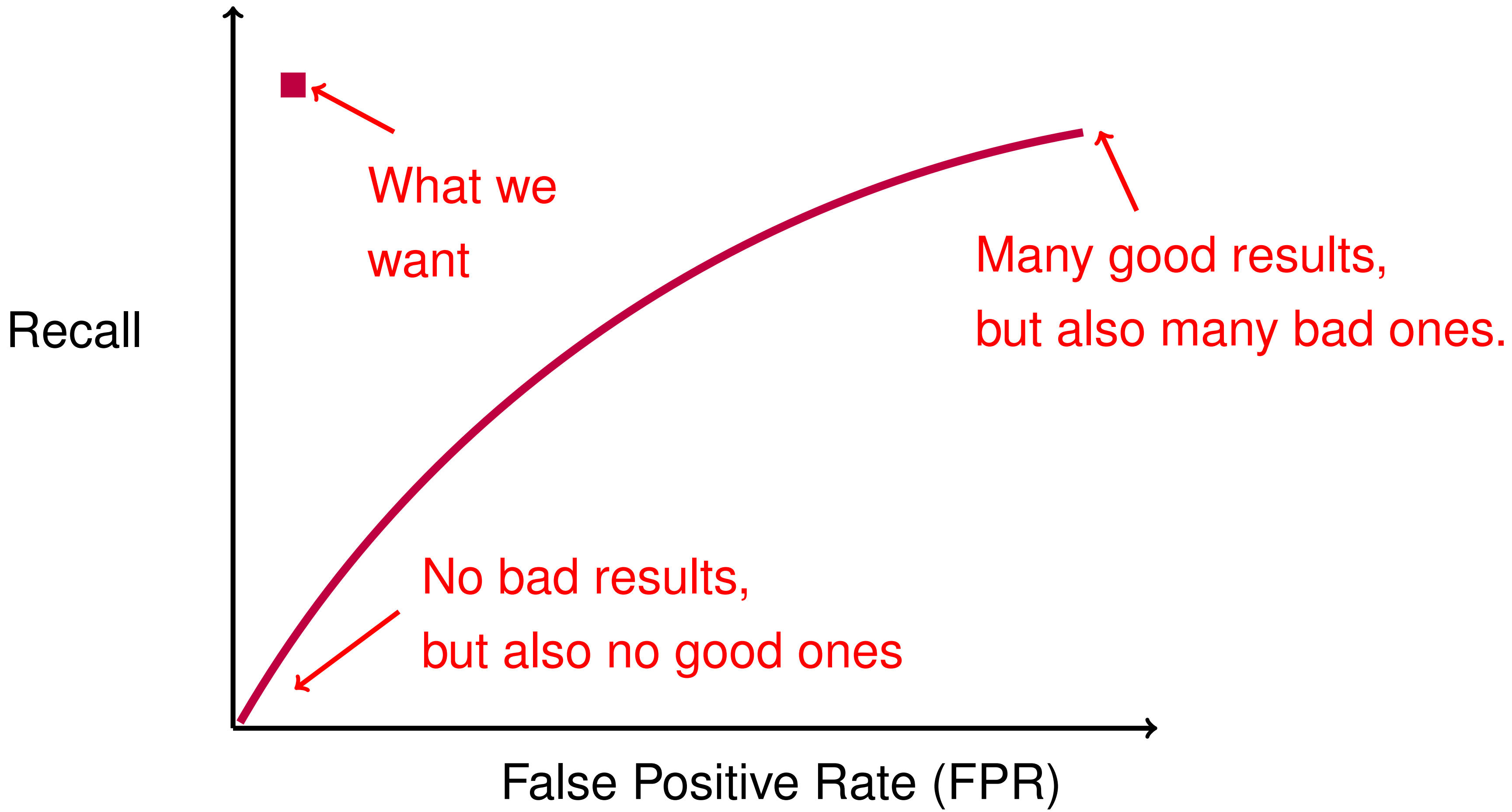
informedness = recall + TNR - 1 = recall - FPR = 100% + 0% - 1 = 0

(It's kind of the F1 measure in the case of a limited population.)

>ROC

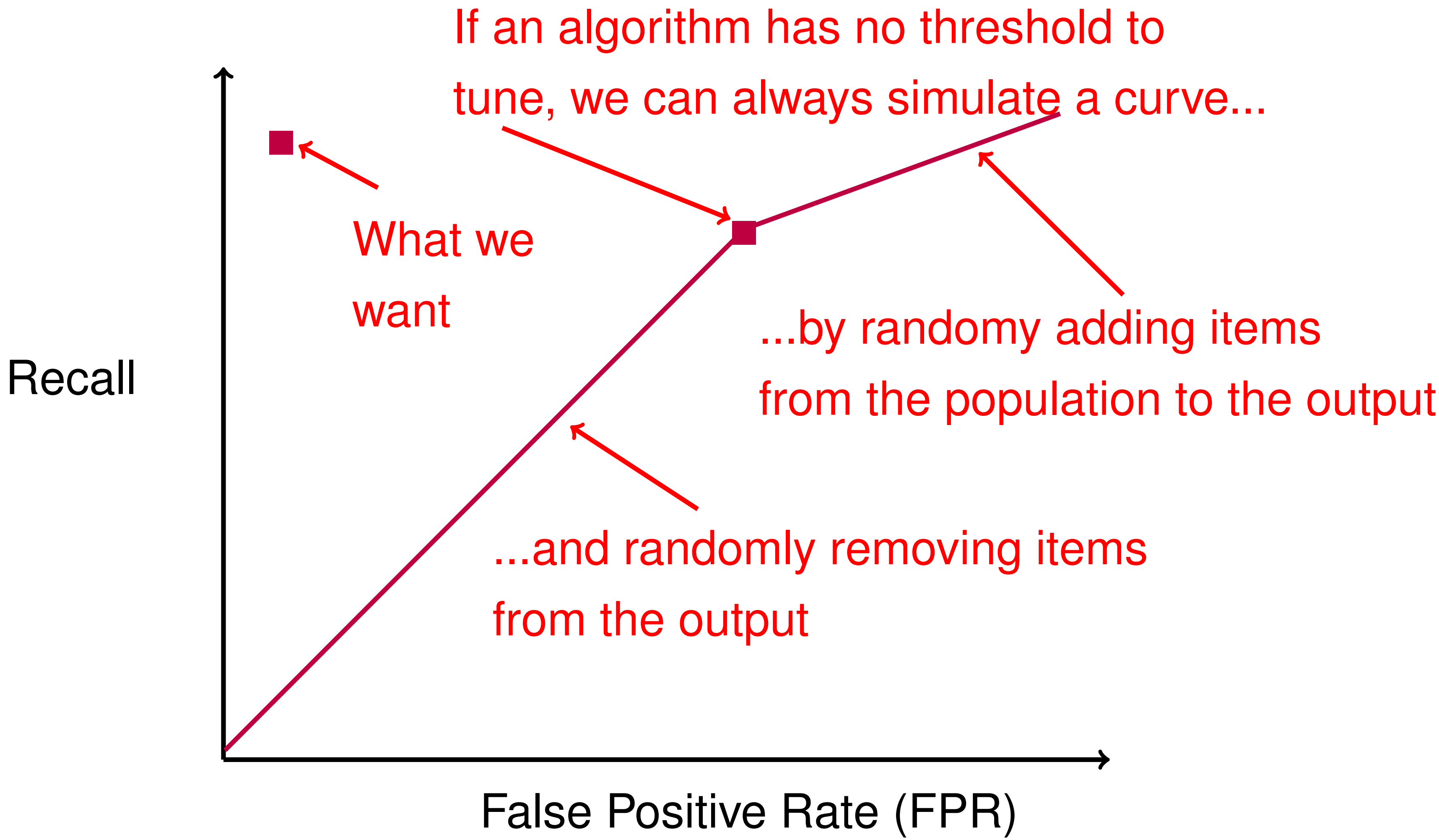
Def: ROC

The ROC (receiver operating characteristic) curve plots recall against the FPR for different thresholds of the algorithm. It guards against imbalanced classes, and is applicable when the population is finite.



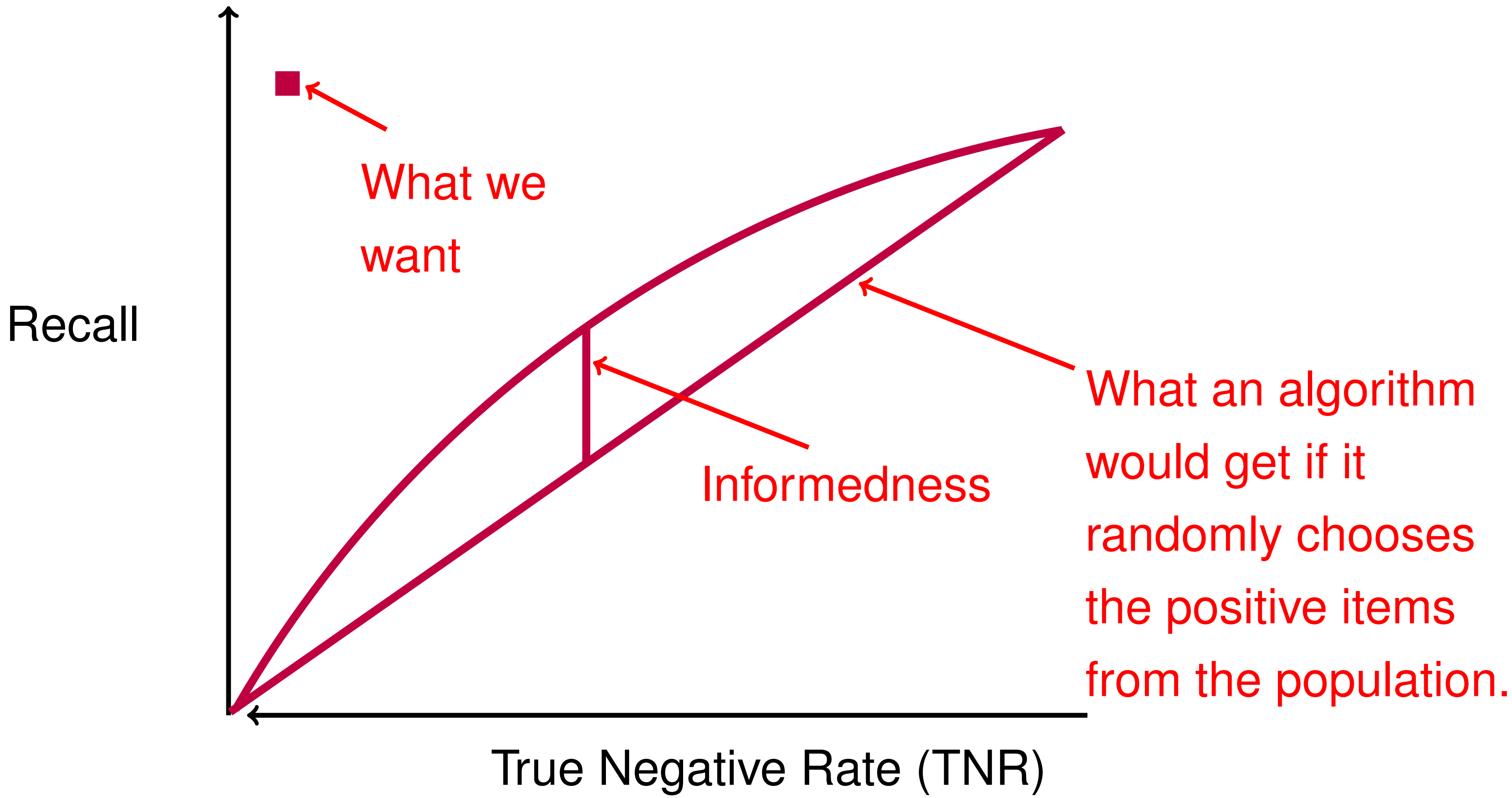
Def: ROC

The ROC (receiver operating characteristic) curve plots recall against the FPR for different thresholds of the algorithm. It guards against imbalanced classes, and is applicable when the population is finite.



Def: ROC

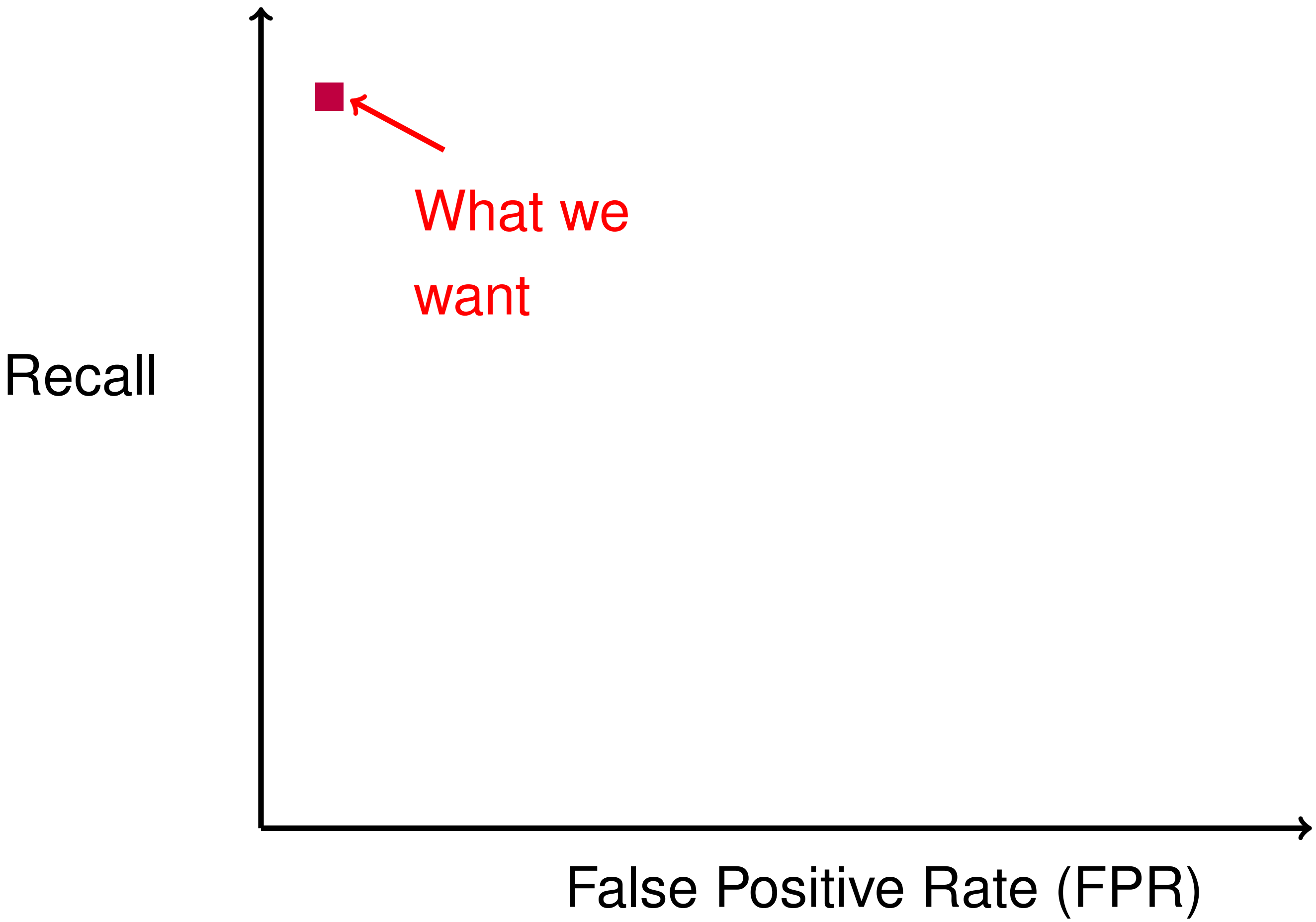
The ROC (receiver operating characteristic) curve plots recall against the FPR for different thresholds of the algorithm. It guards against imbalanced classes, and is applicable when the population is finite.



Def: AUC

The AUC (area under curve) is the area under the ROC curve.

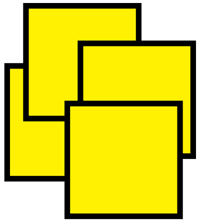
The AUC is between 0 and 1. A high AUC is good. Like F1, AUC combines recall and “correctness”. AUC can be used when the population is known and finite. It guards againts unbalanced classes.



How not to design an IE algorithm

Task: Find Simpson pets

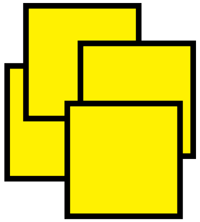
Corpus:



Algorithm: Regex ”Snowball I*”

How not to design an IE algorithm

Task: Find Simpson pets

Corpus: 

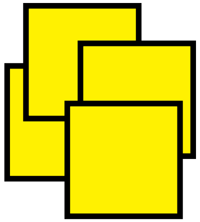
Algorithm: Regex "Snowball I*"

Output: {Snowball I, Snowball II}

How not to design an IE algorithm

Task: Find Simpson pets

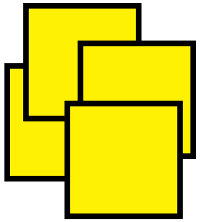
Corpus:



Algorithm: Regex: ”Snowball (I|V)*”

How not to design an IE algorithm

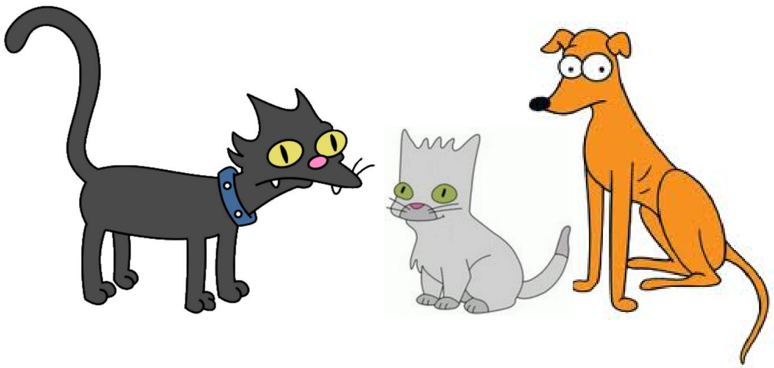
Task: Find Simpson pets

Corpus: 

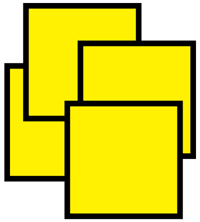
Algorithm: Regex: ”Snowball (I|V)*”

Output: {Snowball I,Snowball II,Snowball IV}

How not to design an IE algorithm



Task: Find Simpson pets

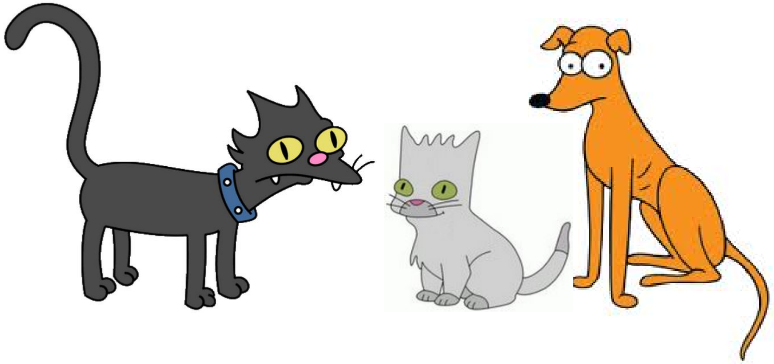
Corpus: 

Algorithm: Regex: ”Snowball (I|V)*”

Output: {Snowball I,Snowball II,Snowball IV}

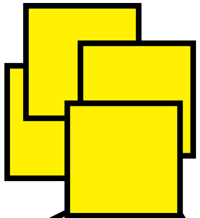
Is this algorithm good?

How to design an IE algorithm



Task: Find Simpson pets

Corpus:



Take only a sample of the corpus

Lisa decides to play music on her saxophone for Coltrane, but the noise frightens him and he commits suicide. As Gil swerves to avoid hitting Snowball V, his car hits a tree and bursts into flames. Since the cat is unhurt, Lisa takes it as a sign of good luck and adopts her. [...]

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Consider only
the sample corpus.

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Gold Standard:
{Coltrane, Snowball I, ...}

Manually make
a gold standard

Consider only
the sample corpus.

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Gold Standard:
{Coltrane, Snowball I, ...}

Algorithm

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Gold Standard:
{Coltrane, Snowball I, ...}

Algorithm

Output:
{...}

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Gold Standard:
{Coltrane, Snowball I, ...}

Algorithm → Output: {...} → Evaluator

How to design an IE algorithm

Task: Find Simpson pets

Corpus:

Gold Standard:
{Coltrane, Snowball I, ...}

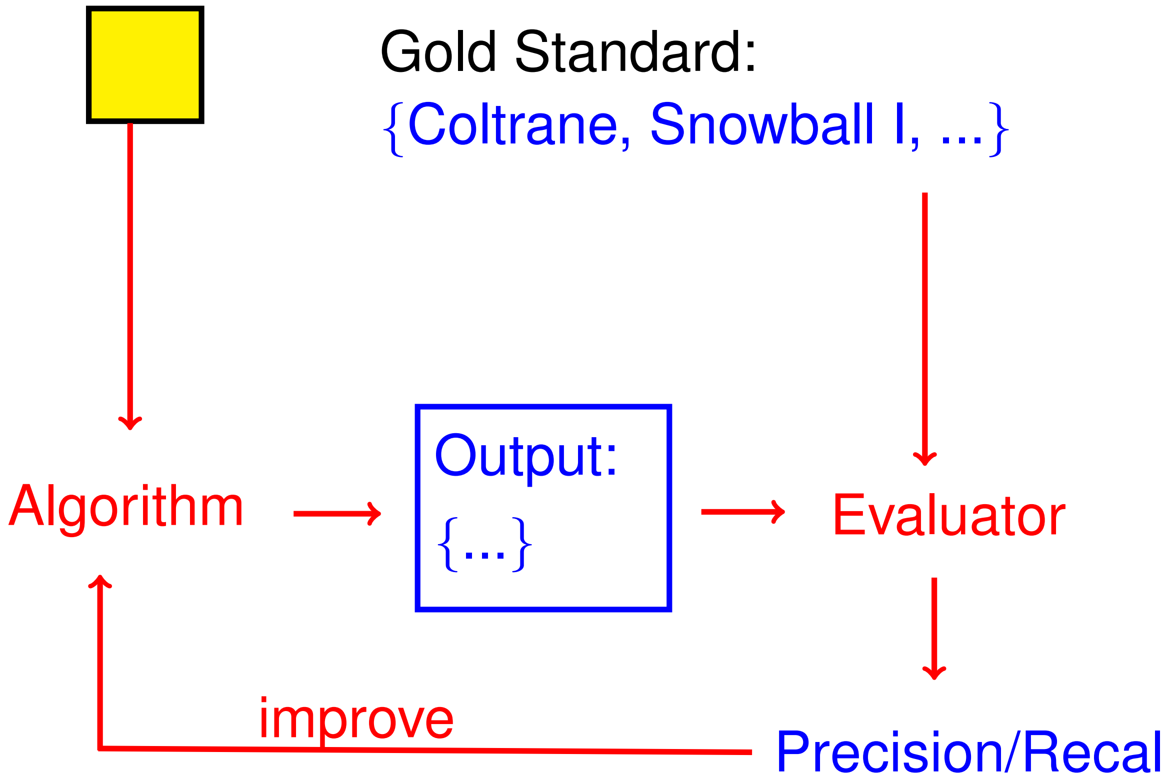
Algorithm → Output: {...} → Evaluator

Precision/Recall

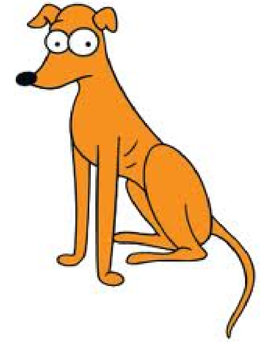
How to design an IE algorithm

Task: Find Simpson pets

Corpus:



Evaluation on a Sample



Corpus:

1: ...A...B...
2: ...C....
3: ..D.....E...
4: .....H.....
5: ..I...J...K...

Sample:

3: ..D.....E...
4: .....H.....

Gold Standard:
{D, E, H}

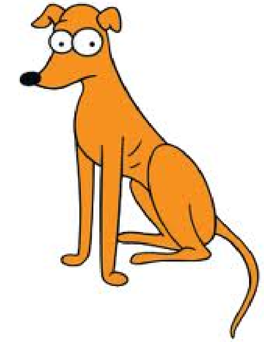
Algorithm:
{1: A, Z
2: C
3: D, E, K
4: L,
5: I, K, X}

Sample:
{
3: D, E, K
4: L
}

Precision: 2/4
Recall: 2/3

A, B, etc. can be entities, but also facts

Evaluation on a Sample



Corpus:

1: ...A...B...

2: ...C....

3: ..D.....E...

4: .....H.....

5: ..I...J...K...

Sample:

3: ..D.....E...

4: .....H.....

Gold Standard:

{D, E, H}

Algorithm:

{1: A, Z

2: C

3: D, E, K

4: L,

5: I, K, X}

Sample:

{

3: D, E, K

4: L

}

Precision: 2/4

Recall: 2/3

Document 5 not considered for computing recall!

Simple case: 1 target per document

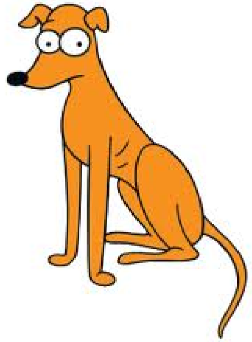
Corpus:

A: ...A'...
B: ...B'...
C: ...C'...
D: ...D'...
E: ...E'...

Sample:

C: ...C'...
D: ...D'...
E: ...E'...

Gold Standard on sample:
{C:C', D:D', E:E'}

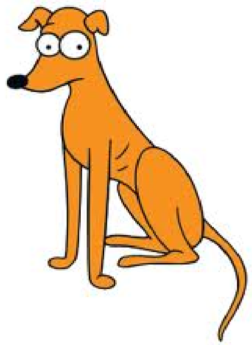


Algorithm:
{A: A'
B: X
C: Z
D: D'
}

Sample output:
{
C: Z,
D: D'
}

Precision: 1/ 2
Recall: 1/3

Simple case: 1 target per document



Corpus:

A: ...A'...
B: ...B'...
C: ...C'...
D: ...D'...
E: ...E'...

Sample:

C: ...C'...
D: ...D'...
E: ...E'...

Gold Standard on sample:
{C:C', D:D', E:E'}

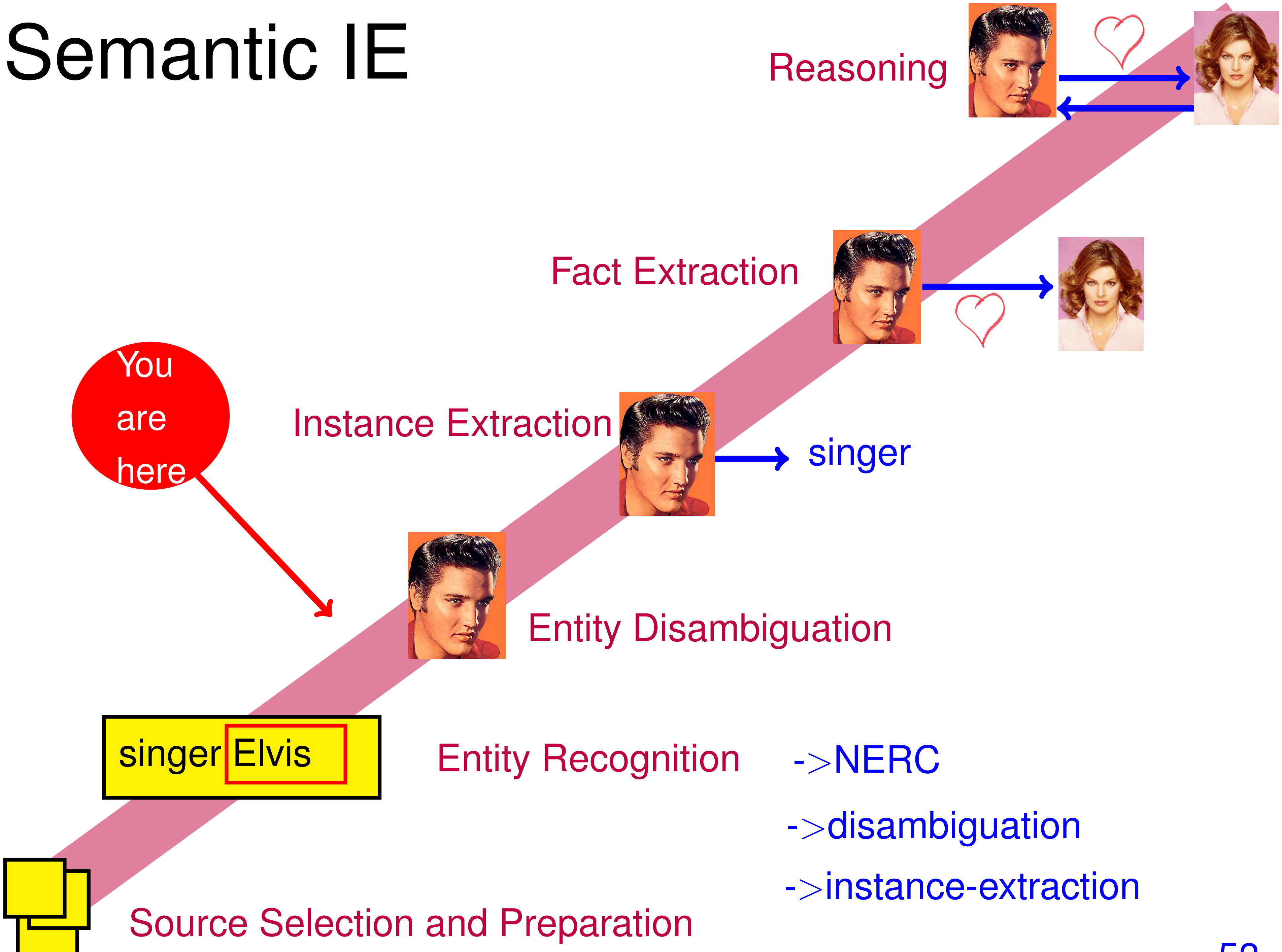
Algorithm:
{A: A'
B: X
C: Z
D: D'
E: K }

Sample output:
{
C: Z,
D: D',
E: K
}

Precision: 1/3
Recall: 1/3

If the algorithm produces one output per input, prec=rec.

Semantic IE


Reasoning
Fact Extraction
You are here
Instance Extraction
singer
Entity Disambiguation
singer Elvis
Entity Recognition
->NERC
->disambiguation
->instance-extraction
Source Selection and Preparation