

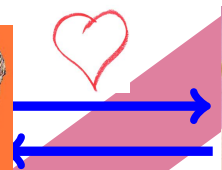
Information Extraction

Lecture 3: Disambiguation & Instance Extraction

Fabian M. Suchanek

Semantic IE

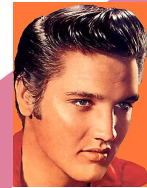
Reasoning



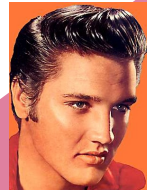
Fact Extraction



Is-A Extraction



→ singer



Entity Disambiguation

singer Elvis

Entity Recognition

Source Selection and Preparation

Overview

- Evaluation
- Disambiguation
- Is-A Extraction

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.



Pixelbazar.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



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)



Pixelart.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. $[A-Z][a-z]^+ [A-Z][a-z]^+$

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)

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

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

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.

E.g., for NER:

The set of names that we wish to extract from a given corpus.

Example: Gold Standard

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.

Example: Gold Standard

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}



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}



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}



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}



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

=> Precision: $2/3 = 66\%$

Def: Recall

The recall 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 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 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 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 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 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: $2/4 = 50\%$

examples>

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 1: [A-Z][a-z]+ Simpson

Output: {The Simpson, Homer Simpson,
Bart Simpson, Lisa Simpson}

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 1: [A-Z][a-z]+ Simpson

Output: {The Simpson, Homer Simpson,
Bart Simpson, Lisa Simpson}

Precision: $3/4=75\%$

Recall: $3/3=100\%$

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 2: $[A-Z][a-z]^+$ $[A-Z][a-z]^+$

Output: {The Simpson, Homer Simpson,
Bart Simpson, Lisa Simpson, Matt Groening}

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 2: $[A-Z][a-z]^+$ $[A-Z][a-z]^+$

Output: {The Simpson, Homer Simpson,
Bart Simpson, Lisa Simpson, Matt Groening}

Precision: $3/5=60\%$

Recall: $3/3=100\%$

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 3: “Homer Simpson”

Output: {Homer Simpson}

Example: Precision & Recall

Gold Standard: {Homer Simpson,
Bart Simpson, Lisa Simpson}

Algorithm 3: “Homer Simpson”

Output: {Homer Simpson}

Precision: $1/1=100\%$

Recall: $1/4=25\%$

Precision & Recall Trade-Off

Algorithm 1: [A-Z][a-z]+ Simpson

Finds all Simpsons, but also
one bad name

High Recall, Low Precision

Algorithm 3: Homer Simpson

Finds only one Simpson,
but this one is correct.

High Precision, Low Recall

Def: Precision & Recall Trade-Off

Explorative algorithms

Extract the gold standard, but also some bad items.

High Recall, Low Precision

Conservative algorithms

Extract only few items, but these are correct.

High Precision, Low Recall

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.
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.
3. On Elvis Radio TM, 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?

How not to design an IE algorithm

Task: Find Simpson pets



Corpus: 

How not to design an IE algorithm

Task: Find Simpson pets



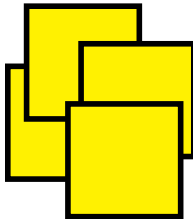
Corpus: 

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: 

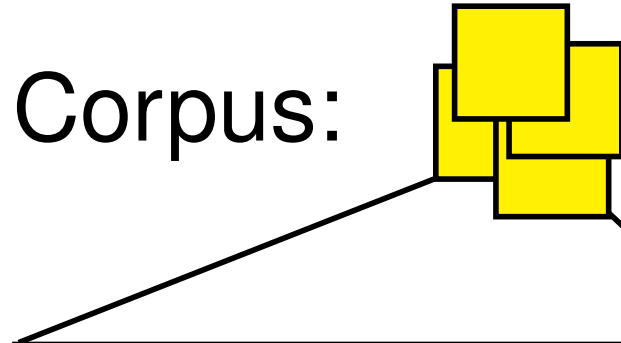
Algorithm: Regex: “Snowball (I|V)*”

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

Is this algorithm good?

How to design an IE algorithm

Task: Find Simpson pets



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: 



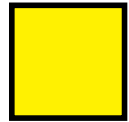
Consider only
the sample corpus.

How to design an IE algorithm

Task: Find Simpson pets



Corpus:



Consider only
the sample corpus.

Gold Standard:

{Coltrane, Snowball I, ...}

Manually make
a gold standard

How to design an IE algorithm

Task: Find Simpson pets



Corpus: 

Gold Standard:

{Coltrane, Snowball I, ...}



Algorithm

How to design an IE algorithm

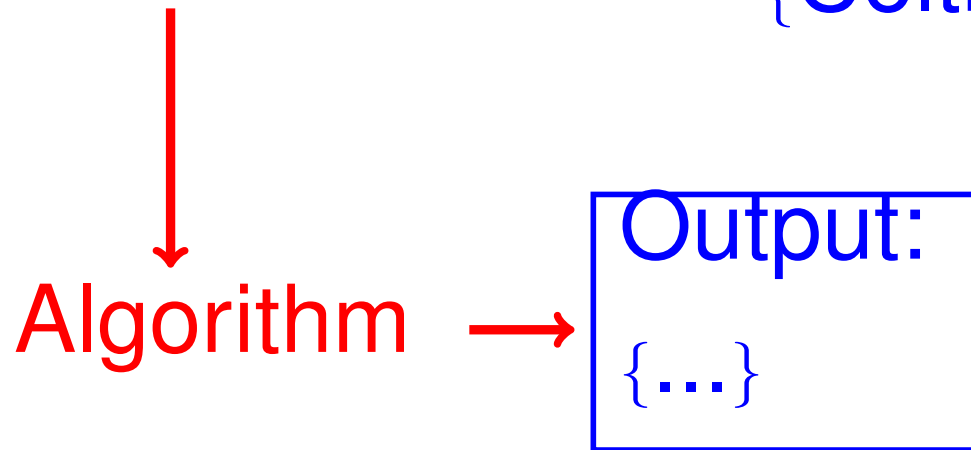
Task: Find Simpson pets



Corpus: 

Gold Standard:

{Coltrane, Snowball I, ...}



How to design an IE algorithm

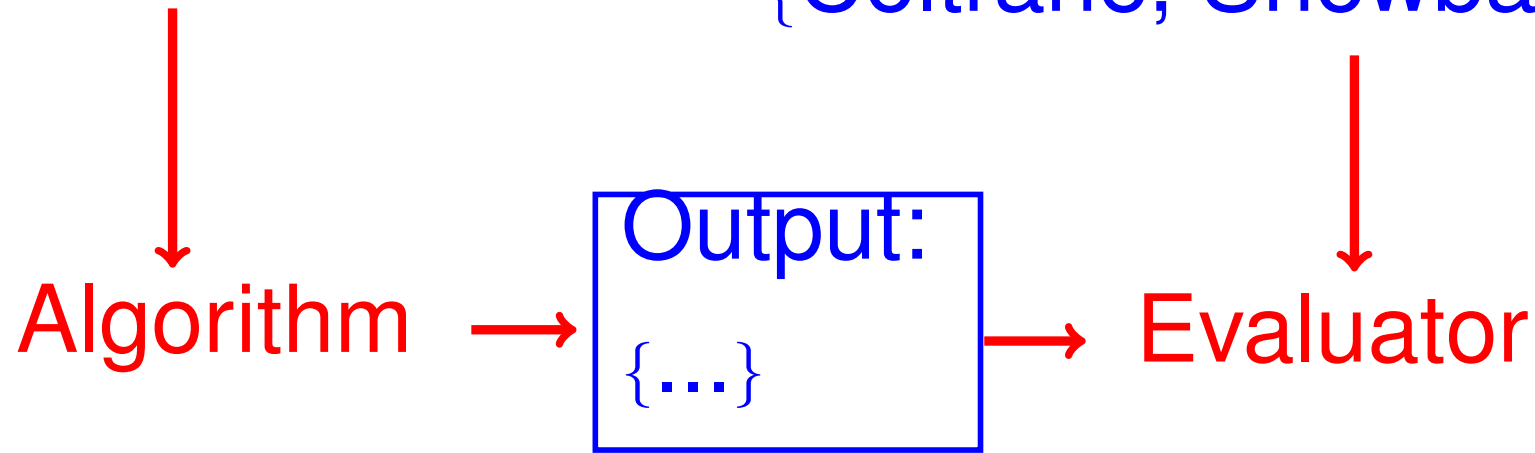
Task: Find Simpson pets



Corpus: 

Gold Standard:

{Coltrane, Snowball I, ...}



How to design an IE algorithm

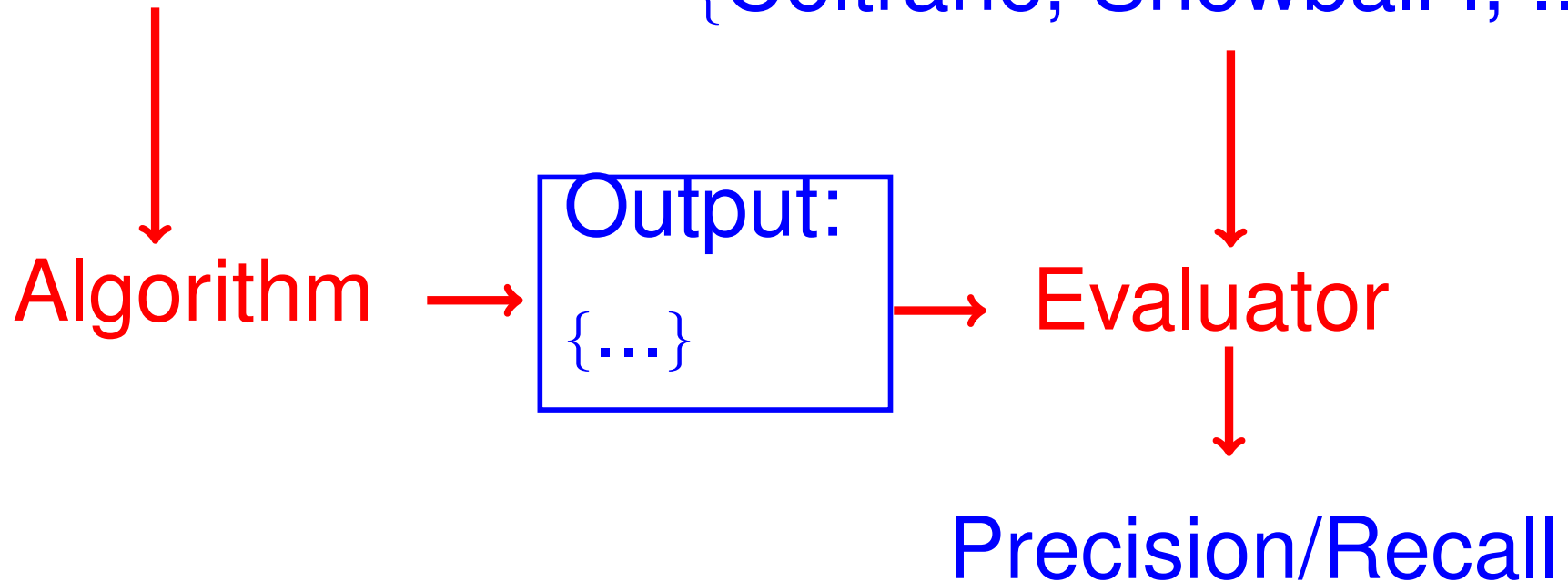
Task: Find Simpson pets



Corpus: 

Gold Standard:

{Coltrane, Snowball I, ...}



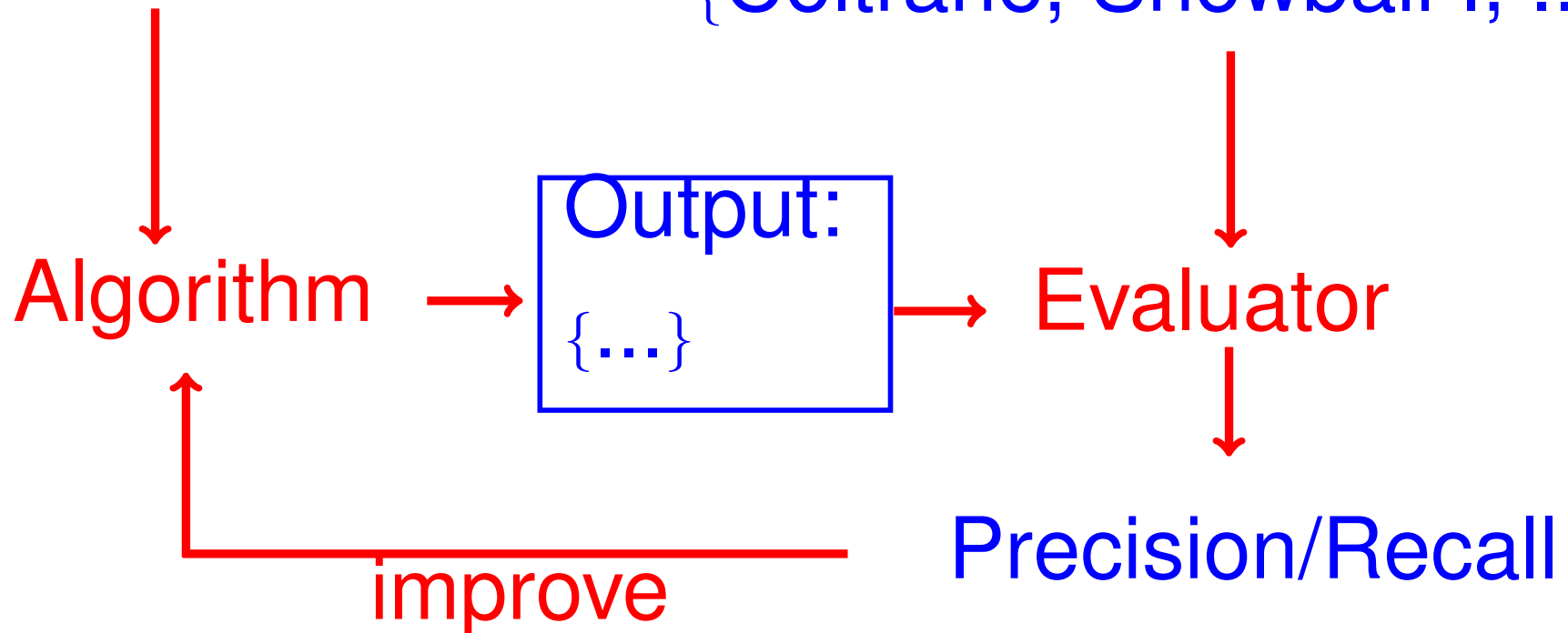
How to design an IE algorithm

Task: Find Simpson pets



Corpus: 

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

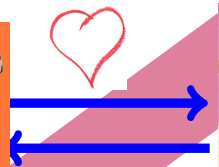
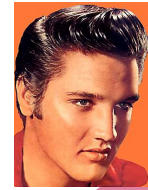


Overview

- Evaluation
- Disambiguation
- Instance Extraction

Semantic IE

Reasoning



Fact Extraction

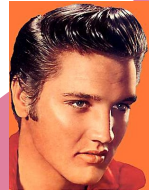


Instance Extraction



→ singer

You
are still
here



Entity Disambiguation

singer Elvis

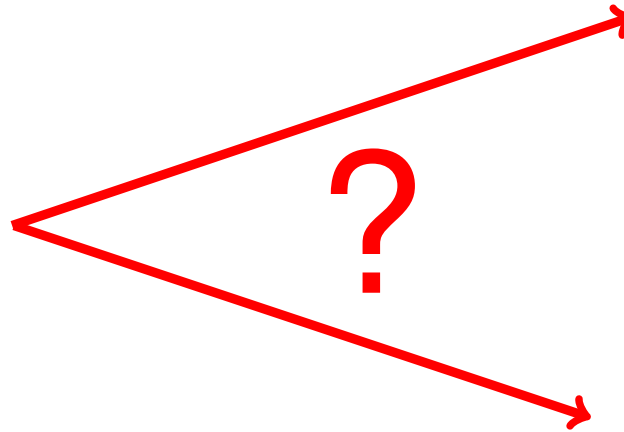
Entity Recognition

Source Selection and Preparation

Def: Disambiguation

Given an ambiguous name in a corpus and its meanings, disambiguation is the task of determining the intended meaning.

Homer eats
a doughnut.



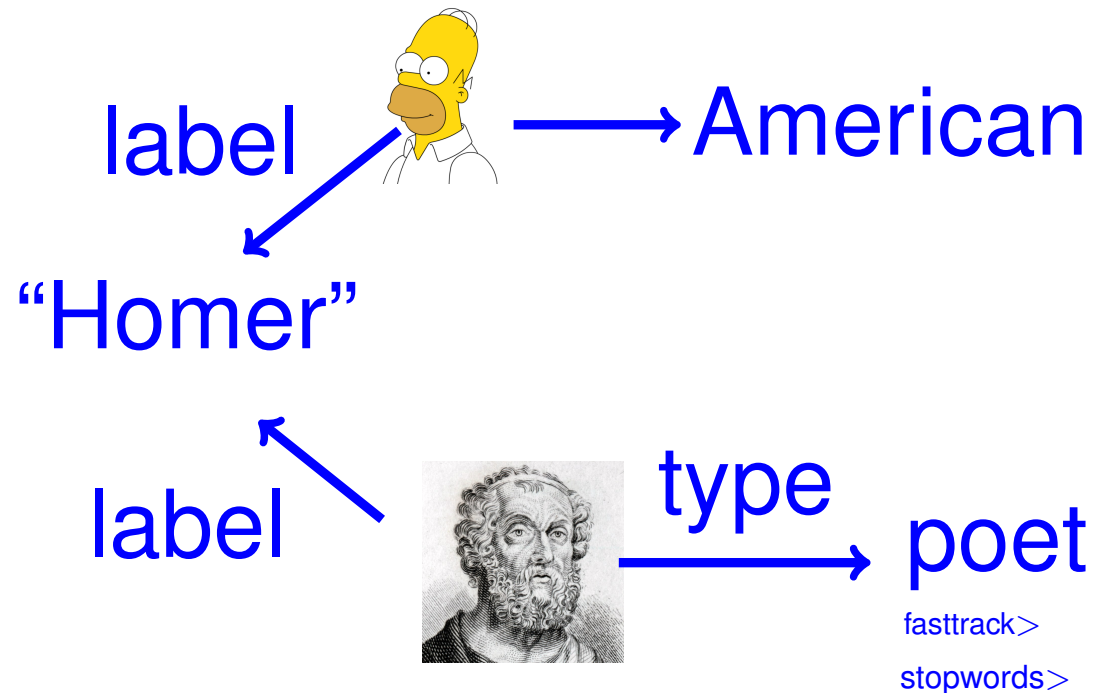
Disambiguation Setting

Usually NER runs first, and the goal is to map the names to entities in a KB.

NER'ed
corpus

Homer eats
a doughnut.

Knowledge Base



Def: Stopword

A stopwords is a word that is common in a corpus but has little value for searching.

e.g., in Postgres

(The definition of stopwords is application-dependent)

Homer eats
a doughnut.

Def: Stopword

A stopwords is a word that is common in a corpus but has little value for searching.

e.g., in Postgres

(The definition of stopwords is application-dependent)

Usually all words except nouns, adjectives, and verbs are stopwords. Auxiliary verbs are stopwords.

Example

Homer eats
a doughnut.

Stopword Rationale

Imagine we search for

How many cats
do the Simpsons have?

Here we do explain
how many teeth
chicken have.

List of
Simpson
cats:

Stopword Rationale

Imagine we search for

How many cats
do the Simpsons have?

Here we do explain
how many teeth
chicken have.

Overlap: 5

List of
Simpson
cats:

Overlap: 2

Stopword Rationale

Imagine we search for

How many cats
do the Simpsons have?

Here we do explain
how many teeth
chicken have.

Overlap: 5

List of
Simpson
cats:

Overlap: 2

Result!



Stopword Rationale

Imagine we search for

cats

Simpsons

Here we do explain
how many teeth
chicken have.

List of
Simpson
cats:

Stopword Rationale

Imagine we search for

cats

Simpsons

Here we do explain
how many teeth
chicken have.

Overlap: 0

List of
Simpson
cats:

Overlap: 2

Stopword Rationale

Imagine we search for

cats

Simpsons

Here we do explain
how many teeth
chicken have.

Overlap: 0

List of
Simpson
cats:

Overlap: 2

Result!

Task: Stopwords

Remove the stopwords from the following sentences.

Don't come here!

Homer was hit by Marge.

Homer ate a few doughnuts.

Task: Stopwords

Remove the stopwords from the following sentences.

Come!

Homer hit Marge.

Homer ate few doughnuts.

(These are fun examples where the meaning of the sentence changes. Usually, applications assume that the meaning of the sentence stay the same.)

Def: Context of a word

The context of a word in a corpus is the multi-set of the words in its vicinity without the stopwords.

(The definition may vary depending on the application)




Homer eats
a doughnut.

Def: Context of a word

The context of a word in a corpus is the multi-set of the words in its vicinity without the stopwords.

(The definition may vary depending on the application)



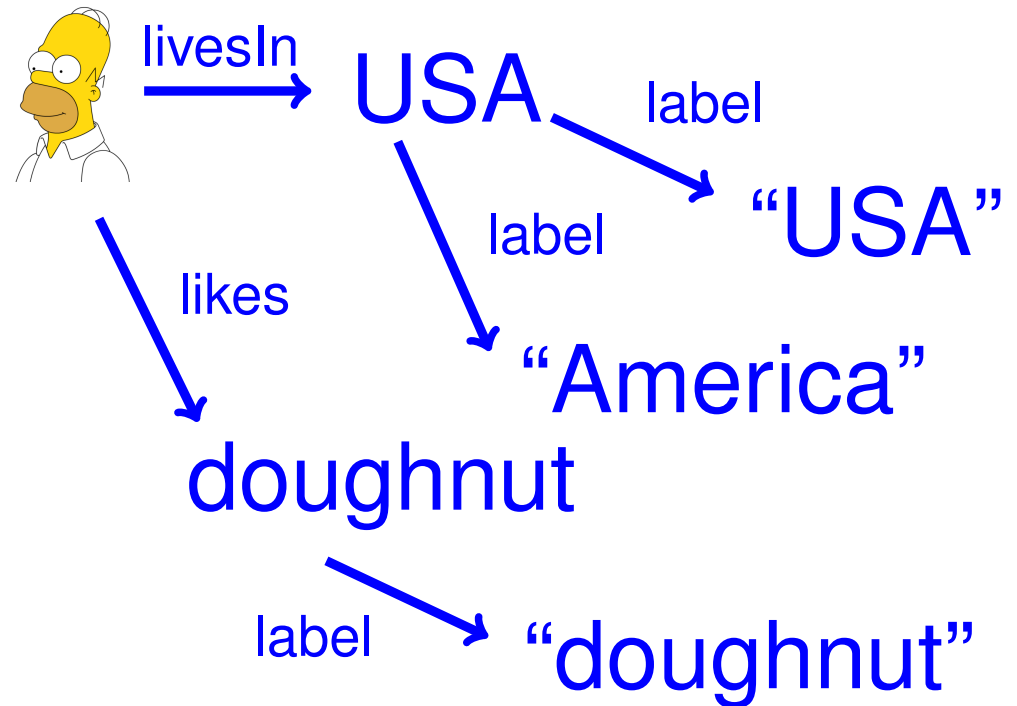
Homer eats
a doughnut.

Context of “Homer”:
{eats, doughnut}

Def: Context of an entity

The context of an entity in a KB
is the set of all labels of all entities
in its vicinity.

(The definition may vary depending on the application)

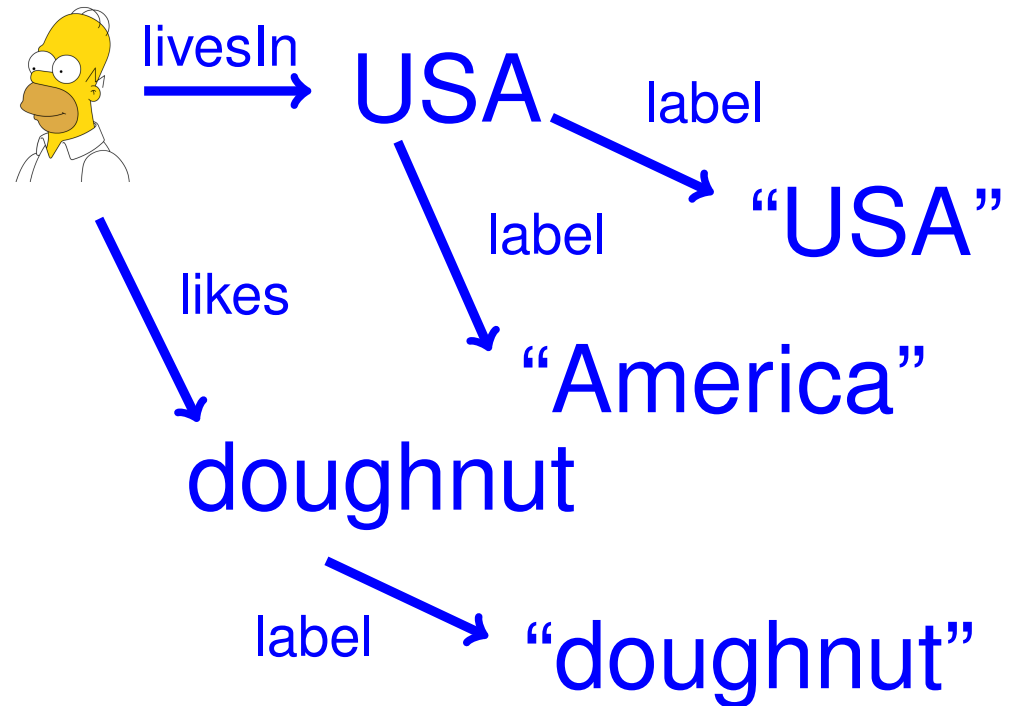


Def: Context of an entity

The context of an entity in a KB
is the set of all labels of all entities
in its vicinity.

(The definition may vary depending on the application)

Context
of Homer:
{doughnut,
USA, America}



Def: Context-based disambiguation

Context-based disambiguation (also: bag of words disambiguation) maps a name in a corpus to the entity in the KB whose context has the highest overlap to the context of the name.

(The definition may vary depending on the application)

Example: Context-based disamb.

For USA Today, Homer is among the top 25 most influential people of the past 25 years.

Example: Context-based disamb.

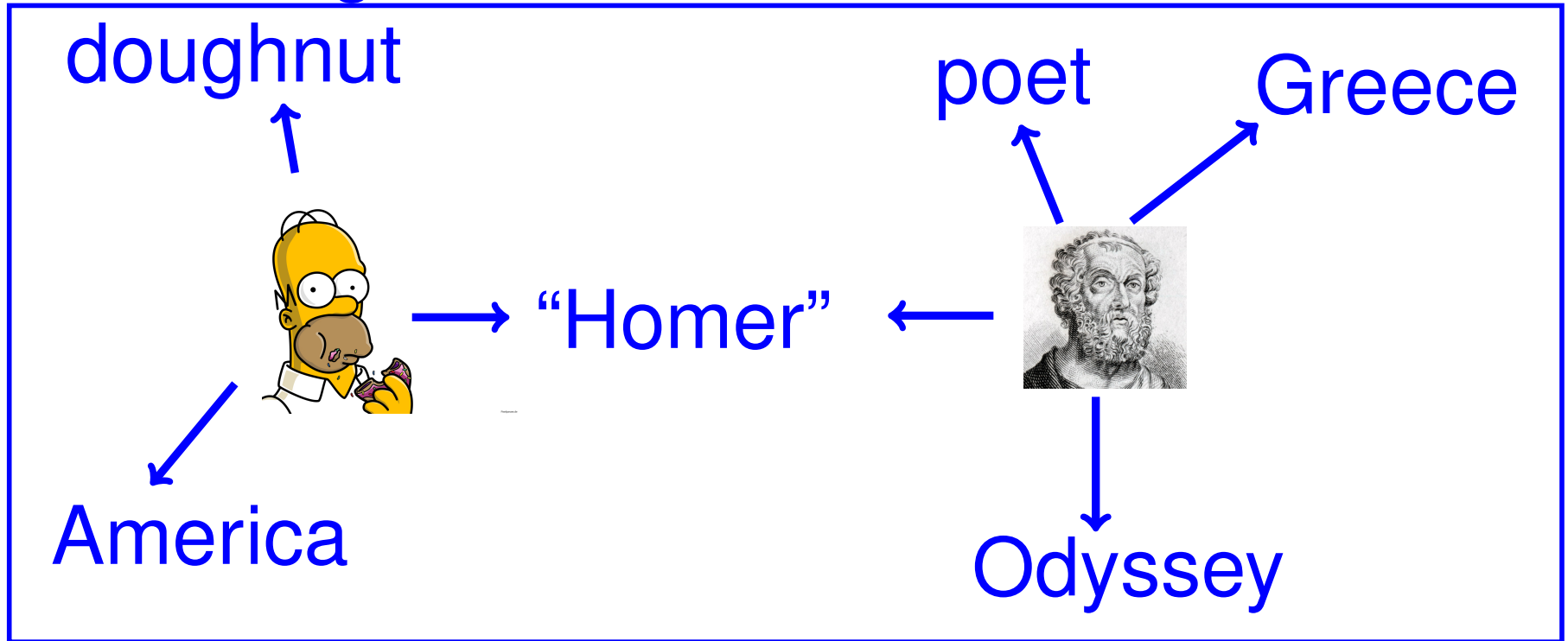
For USA Today, Homer is among the top 25 most influential people of the past 25 years.

Context of “Homer” in corpus:

{USA, Today, top, influential,
people, past, years}

Example: Context-based disamb.

Knowledge Base



Context of "Homer" in corpus:

{USA, Today, top, influential,
people, past, years}

Example: Context-based disamb.

Contexts in the Knowledge Base



{doughnut,
America,
USA,...}



{poet,
Greece,
Odyssey,...}

Context of “Homer” in corpus:

{USA, Today, top, influential,
people, past, years}

Example: Context-based disamb.

Contexts in the Knowledge Base



overlap
with corpus
context=1

{doughnut,
America,
USA,...}



overlap
with corpus
context=0

{poet,
Greece,
Odyssey,...}

Context of “Homer” in corpus:

{USA, Today, top, influential,
people, past, years}

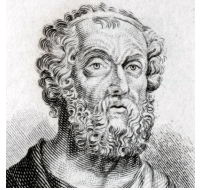
Example: Context-based disamb.

Contexts in the Knowledge Base



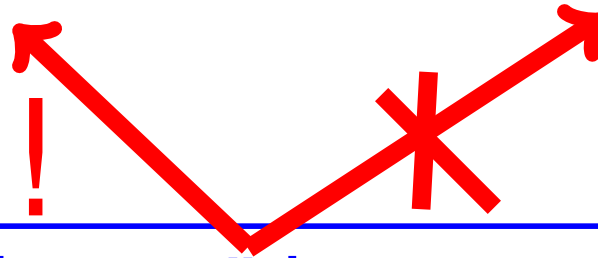
overlap
with corpus
context=1

{doughnut,
America,
USA,...}



overlap
with corpus
context=0

{poet,
Greece,
Odyssey,...}



Context of “Homer” in corpus:

{USA, Today, top, influential,
people, past, years}

Example: Disambiguation by AIDA

AIDA is a system for the disambiguation of entity names, based on YAGO.



Try it out

Example: Disambiguation by AIDA

Disambiguation Method:

prior

prior+sim

prior+sim+coherence

Parameters: (defaults should be OK)

Prior-Similarity-Coherence balancing ratio:

prior VS. sim. balance = 0.4

(prior+sim.) VS. coh. balance 0.6



Ambiguity degree 7



Coherence robustness test threshold: 0.9



Entities Type Filters:

Enter the types her

Mention Extraction:

Stanford NER

Manual

You can manually tag the mentions by putting them between [[and]].
HTML Tables are automatically disambiguated in the manual mode.



Lisa, Bart, and Homer all love the
mother of the house, Marge.

Input Type:TEXT Overall runtime:43s, 78ms

Types list

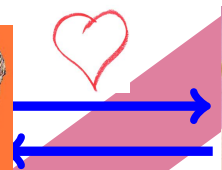
Types tag cloud

Focused T

[[Lisa Simpson](#)]**Lisa**, [[Bart Simpson](#)]**Bart**, and Homer all love the mother of the house, [[Marge Simpson](#)]**Marge**.

Semantic IE

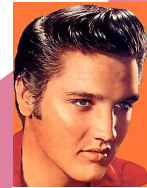
Reasoning



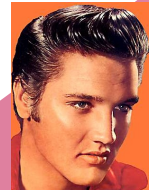
Fact Extraction



Is-A Extraction



singer



Entity Disambiguation

singer Elvis

Entity Recognition

Source Selection and Preparation

Def: Is-A

Is-A (also: hyponymy) is the relation that holds between words X and Y, if X names a subclass or an instance of the class named by Y.

(Strictly speaking, “hyponymy” refers only to the relation of subclass name/superclass name. However, it is often used synonymously with “is-a”)

is-a(“Lisa”, “girl”)

is-a(“girl”, “person”)

is-a(“Jaguar”, “car brand”)

Task: Is-A

Which of the following are true?

1. is-a("Lisa", "girl") ✓
2. is-a("Homer", "person")
3. is-a("Homer", "Marge")
4. is-a("Homer", "poet")
5. is-a("poet", "person")
6. is-a("person", "poet")
7. is-a("person", "Homer")
8. is-a("tree", "data structure")
9. is-a("tree", "plant")

(i.e., are in the KB made from reality plus the facts from The Simpsons)

Is-A needs disambiguation

An is-a fact holds between words,
not between entities.

is-a("Jaguar", "car brand")



Disambiguation

type(JaguarBrand, carBrand)

Is-A Extraction

Is-A Extraction is the task of extracting Is-A facts from a corpus.

(Different from NEA, the class names are not given upfront.)

Lisa is a girl in “The Simpsons”.



is-a(“Lisa”, “girl”)

Example: Instance Extraction

In the Simpson episode "HOMR", Doctor Monson discovers a crayon in Homer's brain and removes it. His IQ goes up from 55 to 105, but he feels uncomfortable and wants it back. Moe, who is not only a bartender but also an unlicensed physician, puts the crayon back, returning Homer to the idiot.

Example: Instance Extraction

In the Simpson episode “HOMR”, Doctor Monson discovers a crayon in Homer’s brain and removes it. His IQ goes up from 55 to 105, but he feels uncomfortable and wants it back. Moe, who is not only a bartender but also an unlicensed physician, puts the crayon back, returning Homer to the idiot.



HOMR	is-a	Simpson episode
------	------	-----------------

Monson	is-a	Doctor
--------	------	--------

Homer	is-a	idiot
-------	------	-------

Moe	is-a	bartender
-----	------	-----------

Moe	is-a	unlicensed physician
-----	------	----------------------

Def: Hearst Patterns

A Hearst pattern is a simple textual pattern that indicates an is-a fact.

“Y such as X”

An idiot such as Homer.



is-a(“Homer”, “idiot”)

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

...some animals, such as dogs...

...some scientists, such as computer scientists...

...some plants, such as nuclear power plants....

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

is-a(“Lisa”, “activists”)

...some animals, such as dogs...

...some scientists, such as computer scientists...

...some plants, such as nuclear power plants....

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

is-a(“Lisa”, “activists”)

...some animals, such as dogs...

is-a(“dogs”, “animals”)

...some scientists, such as computer scientists...

...some plants, such as nuclear power plants....

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

is-a(“Lisa”, “activists”)

...some animals, such as dogs...

is-a(“dogs”, “animals”)

...some scientists, such as computer scientists...

is-a(“computer”, “scientists”) ?

...some plants, such as nuclear power plants....

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

is-a(“Lisa”, “activists”)

...some animals, such as dogs...

is-a(“dogs”, “animals”)

...some scientists, such as computer scientists...

is-a(“computer”, “scientists”) ?

...some plants, such as nuclear power plants....

is-a(“nuc.Pow.Plants”, “plants”) ?

Example: Hearst Patterns

“Y such as X”

...many activists, such as Lisa...

is-a(“Lisa”, “activists”)

...some animals, such as dogs...

is-a(“dogs”, “animals”)

...some scientists, such as computer scientists...

is-a(“computer”, “scientists”) ?

...some plants, such as nuclear power plants....

is-a(“nuc.Pow.Plants”, “plants”) ?

=> Hearst patterns have to be combined with
NER and disambiguation to yield entity facts.

Def: Classical Hearst Patterns

The classical Hearst Patterns are

Y such as X+

such Y as X+

X+ and other Y

Y including X+

Y, especially X+

...where X+ is a list of
names of the form
X[1],...,X[n-1] (and|or)? X[n].

(In the original paper, the X[i] are noun phrases)

These imply is-a(X[i],Y).

Task: Classical Hearst Patterns

Apply

1. Y such as X+
2. such Y as X+
3. X+ and other Y
4. Y including X+
5. Y, especially X+

(you should know these)

I lived in such countries as Germany, France, and Saarland.

He wrote about fictional entities such as Homer, Lisa, and Bielefeld.

The election was won by two clowns, including Grillo.

I love people that are not genies, especially Homer.

Example: Hearst on the Web

"cities such as"

Web

Images

Maps

Shopping

More ▾

Search tools

About 79,800,000 results (0.19 seconds)

[These 12 Hellholes Are Examples Of What The Rest Of America Wi...](#)
[theeconomiccollapseblog.com/.../these-12-hellholes-are-examples-of-wh...](#) ▾

Jul 15, 2012 – The reality is that most of the country has been experiencing a slow decline for a very long time and once thriving **cities such as** Gary, Indiana ...

[City - Wikipedia, the free encyclopedia](#)

[en.wikipedia.org/wiki/City](#) ▾

Every city expansion would imply a new circle (canals together with town walls). In **cities such as** Amsterdam, Haarlem, and also Moscow, this pattern is still ...

try it out

set expansion>
finish>

Def: Set Expansion

Set Expansion is the task of, given names of instances of a class (“seeds”), extracting more such instance names from a corpus.

cities: {“Springfield”, “Seattle”}



Set Expansion

cities: {“Springfield”, “Seattle”,
“Washington”, “Chicago”, ...}

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

1. Find the pattern X, Y, and Z

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

1. Find the pattern $X, Y, \text{ and } Z$
2. If 2 variables match known instance names, add the match of the 3rd.

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

1. Find the pattern $X, Y, \text{ and } Z$
2. If 2 variables match known instance names, add the match of the 3rd.

Seattle, Chicago, and Austin

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

1. Find the pattern $X, Y, \text{ and } Z$
2. If 2 variables match known instance names, add the match of the 3rd.

Seattle, Chicago, and Austin

=> add Chicago

Def: Recursive Pattern Application

Recursive Pattern Application is the following algorithm for set expansion:

cities: {"Austin", "Seattle"}

1. Find the pattern $X, Y, \text{ and } Z$
2. If 2 variables match known instance names, add the match of the 3rd.

Seattle, Chicago, and Austin

=> add Chicago

3. Go to 1

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

cities: {"Springfield", "Austin", "Seattle", "Houston"}

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

cities: {"Springfield", "Austin", "Seattle", "Houston"}

... Houston, Chicago, and Springfield...

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

cities: {"Springfield", "Austin", "Seattle", "Houston"}

... Houston, Chicago, and Springfield...

cities: {"Spr.", "Aust.", "Seattle", "Houston", "Chicago"}

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

cities: {"Springfield", "Austin", "Seattle", "Houston"}

... Houston, Chicago, and Springfield...

cities: {"Spr.", "Aust.", "Seattle", "Houston", "Chicago"}

... Austin, Texas, and Seattle, Washington...

Task: Recursive Pattern Appl.

cities: {"Springfield", "Austin", "Seattle"}

... Austin, Seattle, and Houston...

cities: {"Springfield", "Austin", "Seattle", "Houston"}

... Houston, Chicago, and Springfield...

cities: {"Spr.", "Aust.", "Seattle", "Houston", "Chicago"}

... Austin, Texas, and Seattle, Washington...

Precision may suffer over time

Def: Semantic Drift

Semantic Drift is the problem in Set Expansion that names of instances of other classes get into the set.

cities: {"Chicago", "Seattle", "Texas"}

Def: Table Set Expansion

Table Set Expansion is the following algorithm for set expansion:





Def: Table Set Expansion

Table Set Expansion is the following algorithm for set expansion:

1. Find HTML tables where one column contains 2 known instance names

Largest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	Country	Total Area (sq km)
1.	 Russia	17,098,242
2.	 Canada	9,984,670
3.	 United States	9,826,675
4.	 China	9,596,961





Def: Table Set Expansion

Table Set Expansion is the following algorithm for set expansion:

1. Find HTML tables where one column contains 2 known instance names

Largest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	<u>Country</u>	<u>Total Area (sq km)</u>
1.	 Russia	17,098,242
2.	 Canada	9,984,670
3.	 United States	9,826,675
4.	 China	9,596,961

2. Add all column entries to the set





Def: Table Set Expansion

Table Set Expansion is the following algorithm for set expansion:

1. Find HTML tables where one column contains 2 known instance names

Largest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	Country	Total Area (sq km)
1.	 Russia	17,098,242
2.	 Canada	9,984,670
3.	 United States	9,826,675
4.	 China	9,596,961

2. Add all column entries to the set
3. Go to 1

Ex: Table Set Expansion







countries: {"Russia", "China", "Brazil"}

Ex: Table Set Expansion

countries: {"Russia", "China", "Brazil"}

Richest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)







▲	<u>Country</u>	<u>GDP</u>
1.	 United States	\$15,290,000,000,000
2.	 China	\$11,440,000,000,000
3.	 India	\$4,515,000,000,000
4.	 Japan	\$4,497,000,000,000
5.	 Germany	\$3,139,000,000,000
6.	 Russia	\$2,414,000,000,000

Ex: Table Set Expansion

countries: {"Russia", "China", "Brazil"}

Richest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	<u>Country</u>	<u>GDP</u>
1.	 United States	\$15,290,000,000,000
2.	 China	\$11,440,000,000,000
3.	 India	\$4,515,000,000,000
4.	 Japan	\$4,497,000,000,000
5.	 Germany	\$3,139,000,000,000
6.	 Russia	\$2,414,000,000,000

countries: {"Russia", "China", "Brazil",
"United States", "Japan", "India", "Germ."}

Ex: Table Set Expansion








countries: {"Russia", "China", "Brazil",
"United States", "Japan", "India", "Germ."}

Ex: Table Set Expansion

countries: {"Russia", "China", "Brazil",
"United States", "Japan", "India", "Germ."}

Countries with the Largest Armed Forces in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	<u>Country</u>	<u>Total armed forces</u>
1.	 China	2,255,000
2.	 United States	1,456,850
3.	 India	1,325,000
4.	 Russia	1,058,000
5.	 Korea, South	687,000
6.	 Pakistan	620,000
7.	 Iran	540,000

Ex: Table Set Expansion

countries: {"Russia", "China", "Brazil",
"United States", "Japan", "India", "Germ."}

Countries with the Largest Armed Forces in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	Country	Total armed forces
1.	 China	2,255,000
2.	 United States	1,456,850
3.	 India	1,325,000
4.	 Russia	1,058,000
5.	 Korea, South	687,000
6.	 Pakistan	620,000
7.	 Iran	540,000

countries: {"Russia", ..., "Germany",
"Korea, South", "Pakistan", "Iran"}

Ex: Table Set Expansion

countries: {“Russia”, ..., “Germany”,
“Korea, South”, “Pakistan”, “Iran”}

Countries and dependencies

Rank ↕	Country ↕	To km
—	<i>World</i>	51 (196
1	 Russia	1 (6
—	<i>Antarctica</i>	1 (5
2	 Canada	(3
3	 China	(3
4	 America	(3

Ex: Table Set Expansion

countries: {"Russia", ..., "Germany",
"Korea, South", "Pakistan", "Iran"}

Countries and dependencies

Rank ↕	Country ↕	To km
—	<i>World</i>	51 (196
1	 Russia	1 (€
—	<i>Antarctica</i>	1 (€
2	 Canada	(€
3	 China	(€
4	 America	(€



countries: {
"Russia", ...,
"World",
"Antarctica",
"America"}

Ex: Table Set Expansion

countries: {
“Russia”,...,
“World”,
“Antarctica”,
“America”}

Ex: Table Set Expansion

countries: {
“Russia”,...,
“World”,
“Antarctica”,
“America”}

All continents:
Antarctica
Africa
Asia
America
Australia
Europe

Ex: Table Set Expansion

countries: {
“Russia”,...,
“World”,
“Antarctica”,
“America”}

All continents:
Antarctica
Africa
Asia
America
Australia
Europe

Semantic Drift may occur

Summary: Set Expansion

Set Expansion extends a set of instance names. We saw 2 methods:







1. Recursively applied patterns

X, Y, and Z

2. Table Set Expansion

Richest Countries in the World

view as: [list](#) / [slideshow](#) / [map](#)

▲	Country	GDP
1.	 United States	\$15,290,000,000,000
2.	 China	\$11,440,000,000,000
3.	 India	\$4,515,000,000,000
4.	 Japan	\$4,497,000,000,000
5.	 Germany	\$3,139,000,000,000
6.	 Russia	\$2,414,000,000,000

Summary: Is-A Extraction

Is-a finds names of instance/class or subclass/superclass pairs.

We saw 2 methods:

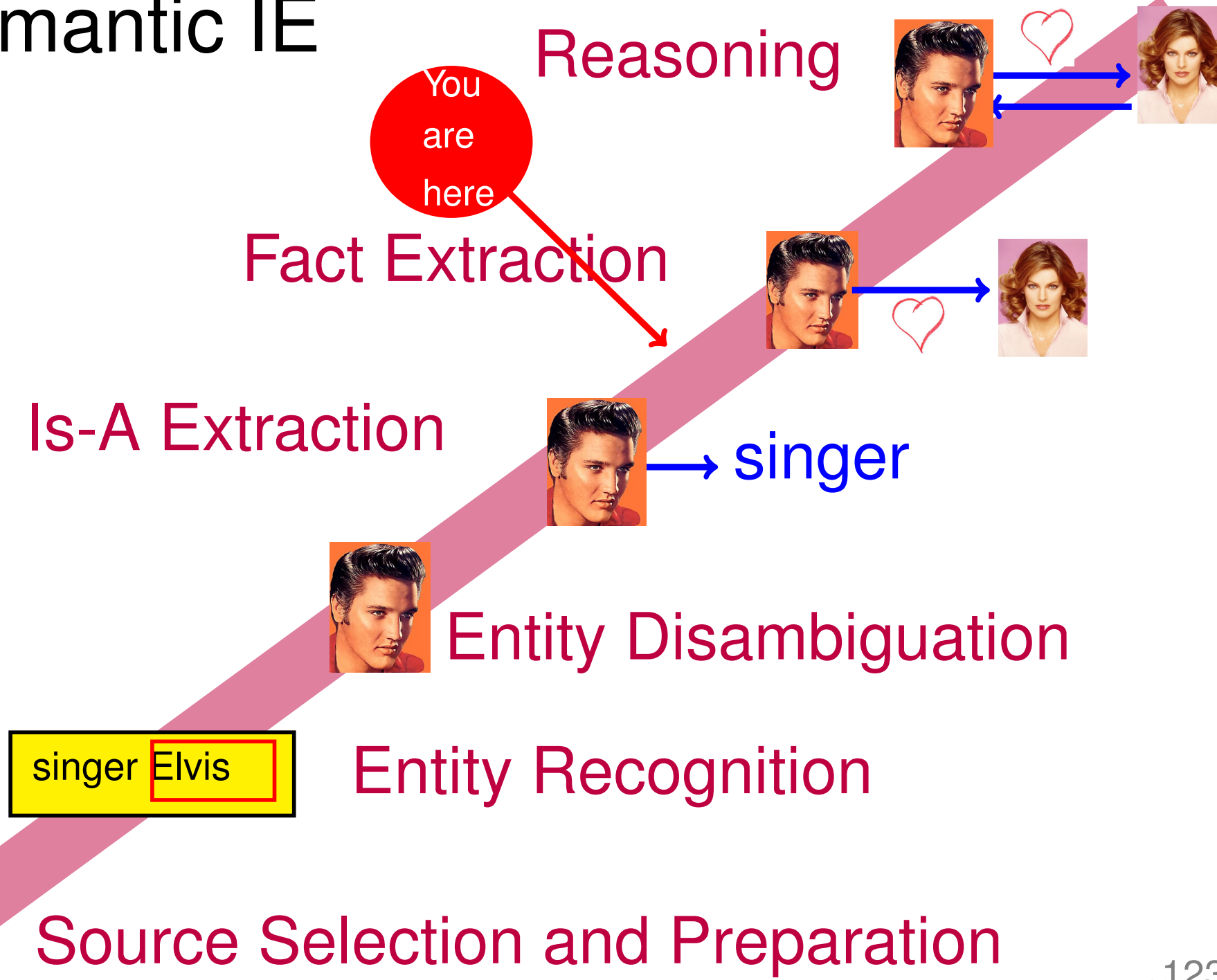
1. Hearst Patterns

vegetarians such as Lisa

2. Set Expansion

cities: {"Chicago", "Springfield"}

Semantic IE



References

AIDA: An Online Tool for Accurate Disambiguation

Marti Hearst: Automatic Acquisition of Hyponyms

Learning Arguments and Supertypes of Semantic Relations

Knowledge Harvesting from Text and Web Sources