Extraction of Entailed Semantic Relations Through Syntax-based Comma Resolution

Vivek Srikumar   Roi Reichart   Mark Sammons
Ari Rappoport   Dan Roth

University of Illinois, Urbana-Champaign

Hebrew University of Jerusalem

The City of Chicago’s OEMC and IBM launch Advanced Video Surveillance System, part of Operation Virtual Shield.

- The City of Chicago possesses OEMC.
- The City of Chicago’s OEMC, IBM form a conjunction
- Advanced Video Surveillance System is part of Operation Virtual Shield.

Motivation

- Sentences can be decomposed into smaller ones
    - Smaller sentences are easier to process

- Syntax gives us cues for decomposition

Along the lines of (Chandrasekar and Srinivas, ’96)

Outline

1. Task: Comma Resolution
2. Learning to Transform Sentences
3. Evaluation

Outline

1. Task: Comma Resolution
2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure
3. Evaluation
   - The Comma Data Set
   - Experiments

Commas tell us *something*

- Authorities have arrested John Smith, a police officer.

  $\Rightarrow$ John Smith is a police officer.

Commas tell us *something*

- Authorities have arrested John Smith, a police officer.

  $\Rightarrow$ John Smith is a police officer.

- Authorities have arrested John Smith, a police officer and his brother today.

  $\Rightarrow$ John Smith, a police officer, his brother are elements of a list.

Commas tell us *something*

- Authorities have arrested John Smith, a police officer.

  $\Rightarrow$ John Smith is a police officer.

- Authorities have arrested John Smith, a police officer and his brother today.

  $\Rightarrow$ John Smith, a police officer, his brother are elements of a list.

- They live in Chicago, IL.

  $\Rightarrow$ Chicago is located in IL.

Commas tell us *something*

Commas indicate several syntactic phenomena

- Appositives
- Lists
- Clausal modifiers
- Locations
- Many others...

Each interpretation implies different relationships .

(van Delden and Gomez, 2002)

(Bayraktar et al., 1998)

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE
- LOCATION
- LIST
- OTHER

Commas come in different flavors

- SUBSTITUTE: An IS-A relation between the arguments

  John Smith, a police officer, was arrested.

  $\Rightarrow$ John Smith *is* a police officer.

  John Smith was arrested.

  A police officer was arrested.

- ATTRIBUTE
- LOCATION
- LIST
- OTHER

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE: One argument is an attribute of the other

  John Smith, 61, was arrested.

  $\Rightarrow$ John Smith *is* 61.

  John Smith was arrested.

- LOCATION
- LIST
- OTHER

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE
- LOCATION: A located-in relation

  Chicago, Illinois saw some snow today.

  $\Rightarrow$ Chicago *is located in* Illinois.

- LIST
- OTHER

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE
- LOCATION
- LIST: A list of entities, adjectives, actions, etc.

  John, James and Kelly left last week.

  $\Rightarrow$ { John, James, Kelly } form a group.
- OTHER

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE
- LOCATION
- LIST
- OTHER: Everything else

However, he cheered up quickly.

"So what if I can't spell pesticde," he said.

$\Rightarrow$ Discourse information, pauses, etc.

Commas come in different flavors

- SUBSTITUTE
- ATTRIBUTE
- LOCATION
- LIST
- OTHER

Comma Resolution

Given a sentence, Comma resolution consists of:

- Interpreting the type of each comma

- Decomposing the sentence based on the interpretation

  - Meaning is preserved

Why Comma Resolution?

- Shorter sentences can be analyzed better
- Decomposition helps other tasks involving text understanding

Why Comma Resolution?

- Shorter sentences can be analyzed better
- Decomposition helps other tasks involving text understanding

  For example, think about textual entailment.

  Given a sentence T, is H true?

Outline

1. Task: Comma Resolution

2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure

3. Evaluation
   - The Comma Data Set
   - Experiments

Outline

1 Task: Comma Resolution

2 Learning to Transform Sentences
- What are we learning from?
- The Learning Procedure

3 Evaluation
- The Comma Data Set
- Experiments

Representation of Sentences

Example:

Both are produced by the same company, Macmillan-McGraw-Hill, a joint venture of McGraw-Hill Inc. and Macmillan's parent, Maxwell Communication Corp.

.

Representation of Sentences

Example:

Both are produced by the same company, Macmillan-McGraw-Hill, a joint venture of McGraw-Hill Inc. and Macmillan's parent, Maxwell Communication Corp.

- Macmillan-McGraw-Hill is a joint venture of ...

Representation of Sentences

Example:

Both are produced by the same company, Macmillan-McGraw-Hill, a joint venture of McGraw-Hill Inc. and Macmillan's parent, Maxwell Communication Corp.

- Macmillan-McGraw-Hill is a joint venture of ...
- Macmillan's parent is Maxwell Communication Corp.

Representation of Sentences

Example:

Both are produced by the same company, Macmillan-McGraw-Hill, a joint venture of McGraw-Hill Inc. and Macmillan's parent, Maxwell Communication Corp.

- Macmillan-McGraw-Hill is a joint venture of ...
- Macmillan's parent is Maxwell Communication Corp.

- Relations might be nested
- We need hierarchical information.
- Parse trees encode this

Sentence Transformation Rules

We want to do two things –

- Look for a pattern in the parse tree of a sentence
- If we find the pattern, then we generate new sentences using the matched parts.

A Sentence Transformation Rule (STR) does these.

More on STRs later $\cdots$

Outline

1. Task: Comma Resolution

2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure

3. Evaluation
   - The Comma Data Set
   - Experiments

An Algorithm Outline

For every example –

- Learn a Sentence Transformation Rule from the example
- Refine it with statistics taken over the entire dataset
- Remove all covered examples

An Algorithm Outline

For every example –

- Learn the most general STR from the example
- Refine it with statistics taken over the entire dataset
- Remove all covered examples

An Algorithm Outline

For every example –

- Learn the most general STR from the example
- Specialize it with statistics taken over the entire dataset
- Remove all covered examples

An Algorithm Outline

For every example –

- Learn the most general STR from the example
- Specialize it with statistics taken over the entire dataset
- Remove all covered examples

This is A Sentence Transformation Rule Learner (ASTRL)

Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.

- But Fujitsu is n’t alone.
- But Japan ’s No. 1 computer maker is n’t alone.
- Fujitsu is Japan ’s No. 1 computer maker.

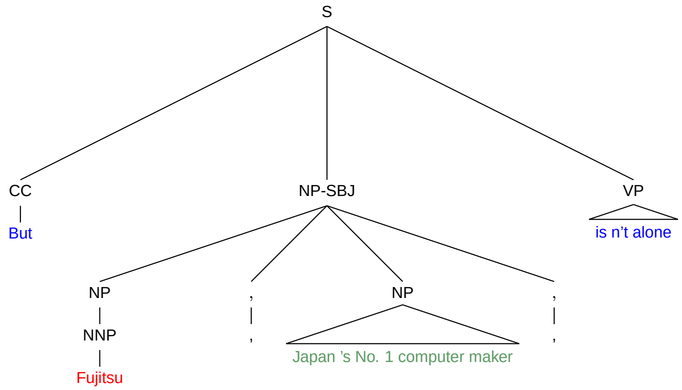
Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.

- But Fujitsu is n’t alone.
- But Japan ’s No. 1 computer maker is n’t alone.
- Fujitsu *is* Japan ’s No. 1 computer maker.

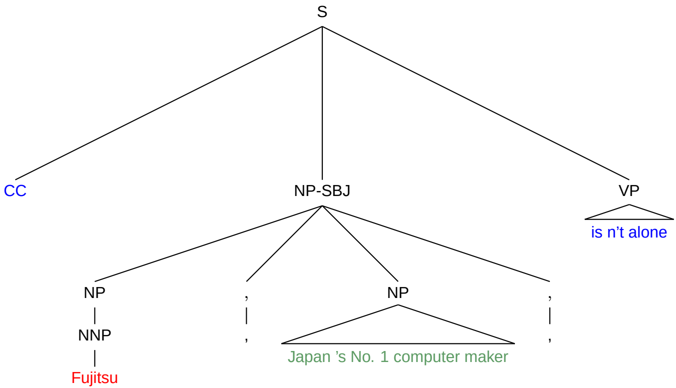
Learning from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



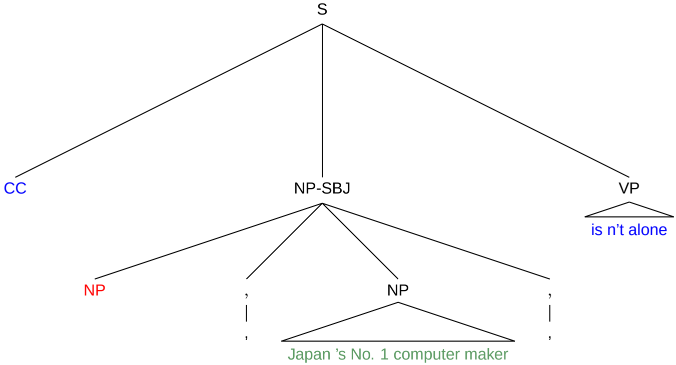
Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



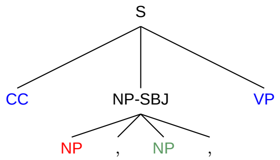
Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



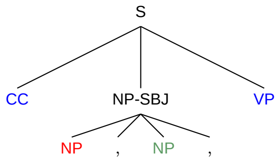
Learning from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



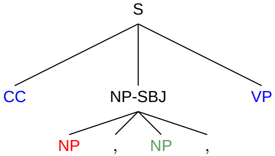
. But Fujitsu is n’t alone.

. But Japan ’s No. 1 computer maker is n’t alone.

. Fujitsu *is* Japan ’s No. 1 computer maker.

Learning from a Single Example

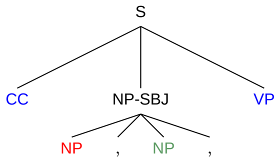
But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



- CC NP VP.
- But Japan ’s No. 1 computer maker is n’t alone.
- Fujitsu *is* Japan ’s No. 1 computer maker.

Learning from a Single Example

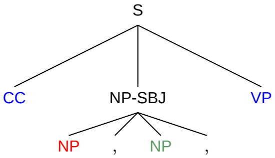
But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



- CC NP VP.
- CC NP VP.
- NP *is* NP.

Learning from a Single Example

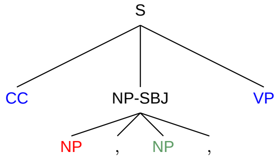
But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



- CC NP VP.
- CC NP VP.
- NP *is* NP.

Learning from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



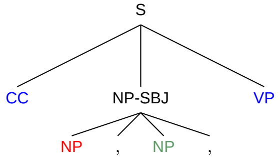
- CC NP VP.
- CC NP VP.
- NP *is* NP.

Abstracted away some details from parse tree.

Can we get a smaller pattern?

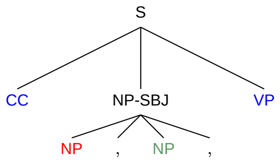
Learning more from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



Learning more from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



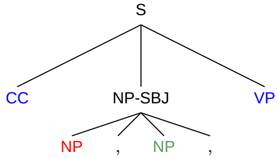
Leaves of this pattern tree:

CC NP , NP , VP

- CC NP VP
- CC NP VP
- NP *is* NP

Learning more from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



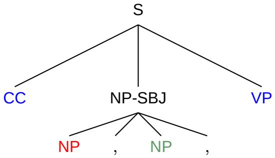
Leaves of this pattern tree:

CC NP , NP , VP

CC NP VP

Learning more from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



Leaves of this pattern tree:

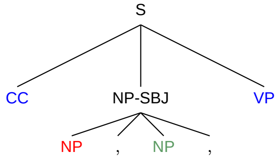
CC NP , NP , VP

CC NP VP

NP has substituted NP, NP,

Learning more from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



Leaves of this pattern tree:

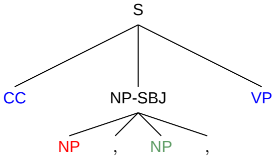
CC $\boxed{\text{NP , NP ,}}$ VP

CC $\boxed{\text{NP}}$ VP

NP has substituted *NP-SBJ*

Learning more from a Single Example

But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



Leaves of this pattern tree:

CC $\boxed{\text{NP , NP ,}}$ VP

CC $\boxed{\text{NP}}$ VP

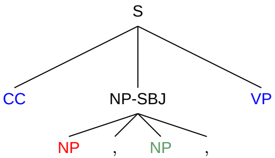
But Fujitsu is n't alone

NP has substituted *NP-SBJ*

Fujitsu has substituted *Fujitsu, Japan 's No. 1 computer maker,*

Learning from a Single Example

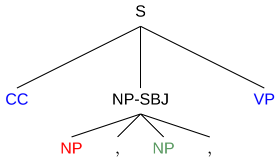
But Fujitsu, Japan 's No. 1 computer maker, is n't alone.



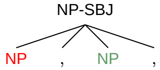
- CC NP VP.
- CC NP VP.
- NP *is* NP.

Learning from a Single Example

But Fujitsu, Japan ’s No. 1 computer maker, is n’t alone.



- CC NP VP.
- CC NP VP.
- NP *is* NP.



- NP (Substitute)
- NP (Substitute)
- NP *is* NP. (Introduce)

Substitute: NP substitutes root of pattern (NP-SBJ)

Introduce: The relation is independent of the context in which the pattern is found.

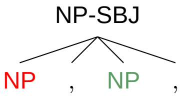
Definition: Sentence Transformation Rule

$$\mathbf{L} \longrightarrow \mathbf{R}$$

Definition: Sentence Transformation Rule

$$\mathbf{L} \longrightarrow \mathbf{R}$$

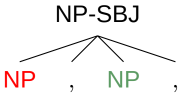
- *L*: A tree fragment (think part of a parse tree)



Definition: Sentence Transformation Rule

$$\mathbf{L} \rightarrow \mathbf{R}$$

- *L*: A tree fragment (think part of a parse tree)

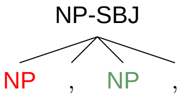


- *R*: Set of combinations of nodes of *L*, possibly with new tokens
  - NP
  - NP
  - NP *is* NP.

Definition: Sentence Transformation Rule

$$\mathbf{L} \rightarrow \mathbf{R}$$

- *L*: A tree fragment (think part of a parse tree)



- *R*: Set of combinations of nodes of *L*, possibly with new tokens
  - NP (Substitute)
  - NP (Substitute)
  - NP *is* NP. (Introduce)
- Each relation is marked as *Introduce* or *Substitute*

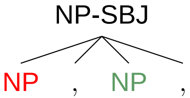
Sentence Transformation Rules

Applying $L \rightarrow R$ to a sentence, whose parse tree is p

1. Look for matches of L in p
2. For every match:

   Generate new sentences, as specified by R

Learning from a Single Example



. NP (Substitute)

. NP (Substitute)

. NP *is* NP. (Introduce)

- This is the smallest pattern tree that is possible.

  Easy to check

- This is the most general STR for the example.

An Algorithm Outline

For every example –

- Learn the most general STR from the example
- Specialize it with statistics taken over the entire dataset
- Remove all covered examples

This is A Sentence Transformation Rule Learner (ASTRL)

An Algorithm Outline

For every example –

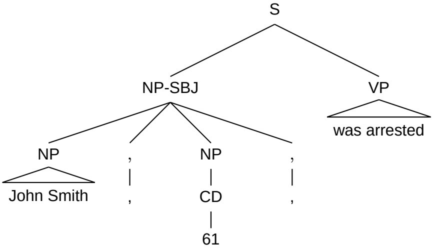
✓ Learn the most general STR from the example

- Specialize it with statistics taken over the entire dataset
- Remove all covered examples

This is A Sentence Transformation Rule Learner (ASTRL)

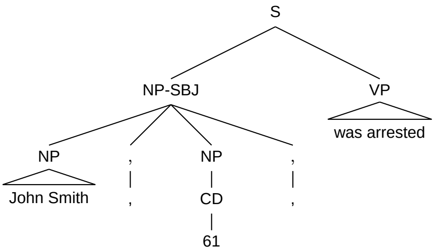
Why Refinement?

John Smith, 61, was arrested.

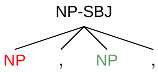


Why Refinement?

John Smith, 61, was arrested.

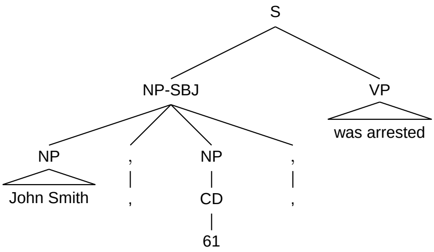


generates

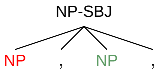


Why Refinement?

John Smith, 61, was arrested.



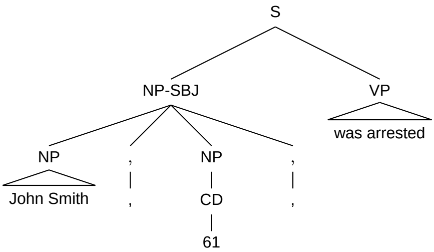
generates



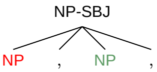
But this is not an apposition. **CD** will help disambiguate.

Why Refinement?

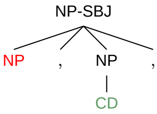
John Smith, 61, was arrested.



generates



But this is not an apposition. **CD** will help disambiguate.



Why Refinement?

Problem:

- The most general STR is generated from a single example
- It might cover other phenomena too

    i.e. it might be too general

Solution:

Specialize the STR to maximize performance over the entire dataset.

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type t:

1. p = All examples of type t

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type t:

1. p = All examples of type t
2. For each example in p:
    1. r = Most general STR that covers this example

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type t:

1. p = All examples of type t
2. For each example in p:
   1. r = Most general STR that covers this example
   2. Compute score of r

      Score = fraction of positive examples covered − fraction of negative examples covered

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type t:

1. p = All examples of type t
2. For each example in p:
   1. r = Most general STR that covers this example
   2. Compute score of r
   3. Get all neighbors of the pattern from the parse tree

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type *t*:

1. *p* = All examples of type *t*
2. For each example in *p*:
   1. *r* = Most general STR that covers this example
   2. Compute score of *r*
   3. Get all neighbors of the pattern from the parse tree
   4. For every neighbor
      1. Add it to the pattern and recompute score
      2. If the score is better, keep it and recalculate the neighbors

A Sentence Transformation Rule Learner (ASTRL)

For a given comma type *t*:

1. *p* = All examples of type *t*
2. For each example in *p*:
    1. *r* = Most general STR that covers this example
    2. Compute score of *r*
    3. Get all neighbors of the pattern from the parse tree
    4. For every neighbor
        1. Add it to the pattern and recompute score
        2. If the score is better, keep it and recalculate the neighbors
    5. Add *r* to list of STRs
    6. Remove all covered examples from *p*

Outline

1. Task: Comma Resolution
2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure
3. Evaluation
   - The Comma Data Set
   - Experiments

Outline

1. Task: Comma Resolution
2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure
3. Evaluation
   - The Comma Data Set
   - Experiments

Data Set

- 1000 sentences from WSJ corpus, all with commas
- Manually annotated with comma type and transformed sentences
  - Four annotators, high agreement

Data available for download at `http://L2R.cs.uiuc.edu/~cogcomp/data.php`

Data: Example Annotation

But Fujitsu [1], Japan’s No. 1 computer maker [1], isn’t alone.

[1] SUBSTITUTE

- But Fujitsu is n’t alone.
- But Japan ’s No. 1 computer maker is n’t alone.
- Fujitsu *is* Japan ’s No. 1 computer maker.

Data: Comma types

- SUBSTITUTE: An IS-A relation between the arguments

  John Smith, a police officer, was arrested.
- ATTRIBUTE: One argument is an attribute of the other

  John Smith, 61, was arrested.
- LOCATION: A located-in relation

  Chicago, Illinois saw some snow today.
- LIST: A list of entities, adjectives, actions, etc.

  John, James and Kelly left last week.
- OTHER: Everything else

  However, he cheered up soon.

Outline

1. Task: Comma Resolution
2. Learning to Transform Sentences
   - What are we learning from?
   - The Learning Procedure
3. Evaluation
   - The Comma Data Set
   - Experiments

Experimental Setup

1. GOLD-GOLD

   Train and test using gold standard trees

2. GOLD-CHARNIAK

   Train with gold standard trees and test with trees generated by a statistical parser (Charniak and Johnson, 2005)

3. CHARNIAK-CHARNIAK

   Train and test using generated parses

Results: Extracted Sentences

- Most relevant for entailment applications
- Different comma types used for learning STRs
- Only relations scored during evaluation, irrespective of type.

Setting	P	R	F
Gold-Gold	86.1	75.4	**80.2**
Gold-Charniak	77.3	60.1	**68.1**
Charniak-Charniak	77.2	64.8	**70.4**

Results

- Performance of ASTRL is close to human agreement

- Even the most general STR is often quite good

  Specialization disambiguates SUBSTITUTE and ATTRIBUTE.

  ATTRIBUTE F-score gain of $\approx$ 14 %

- F-score for identifying OTHER $\approx$ 80%

Conclusion

- Defined *Comma Resolution*
  - Extracting relations based on commas
- Developed an annotation scheme and released an annotated dataset
- Developed an efficient learning algorithm
  - Uses syntax to learn and generate relations
- Same technique could be used to draw inferences from other other phenomena (like possessives)