# The Amazing Project

Tommaso Pasini & Valentina Pyatkin

(pasini | pyatkin)@di.uniroma1.it

Good afternoon, How's everything going?

Good afternoon, Mr. Amor. Everything is going extremely well.

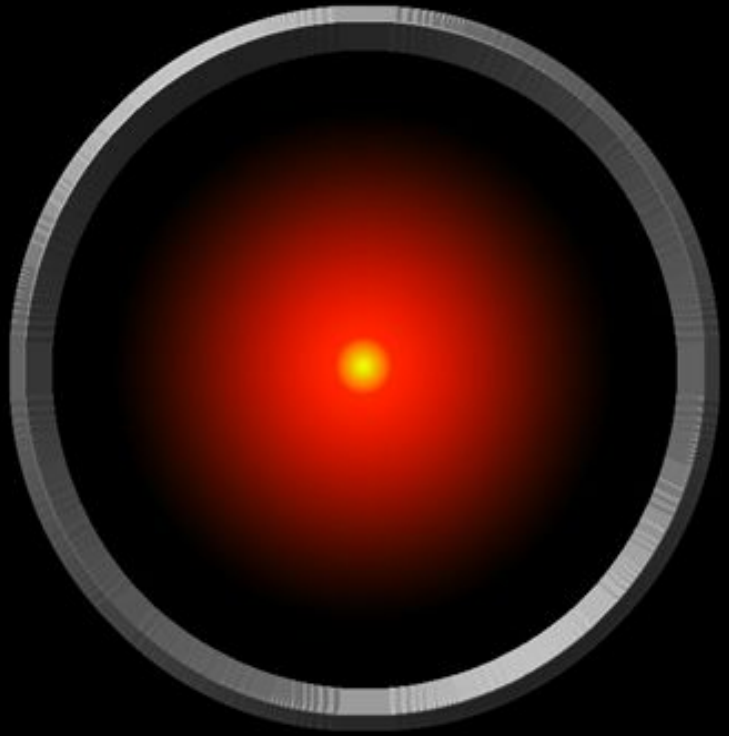
You have an enormous responsibility on this mission.  Does this ever cause you any lack of confidence?

Let me put it this way, Mr. Amor. We are all, by any practical definition of the words, foolproof and incapable of error.

So, despite your enormous intellect, are you ever frustrated by your dependence on people to carry out actions?

Not in the slightest bit.

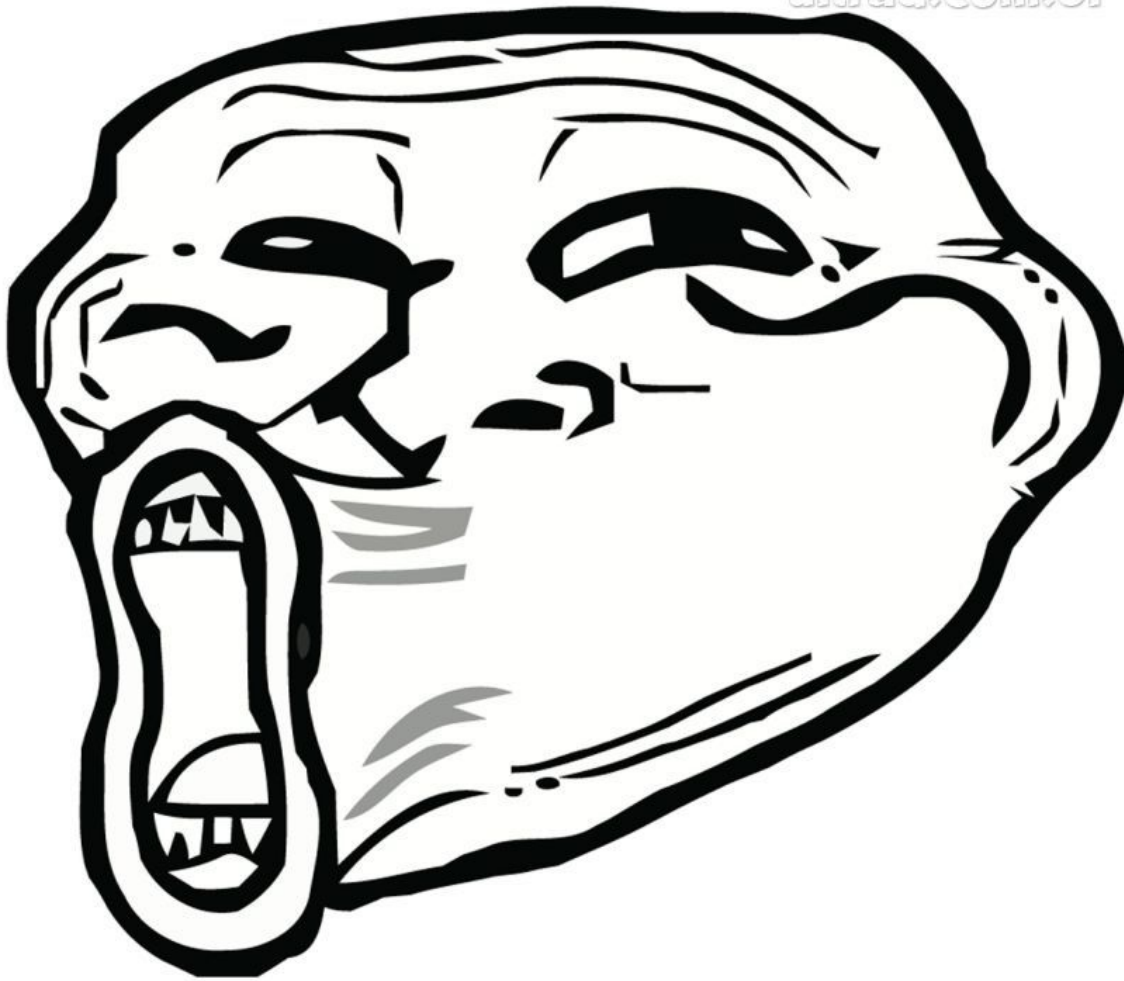
I’m sorry Dave,
I’m afraid I can’t do that.



# The Amazing Project - HAL-9000

- Implement your HAL-9000
- The AI has to be capable to solve simple problems occurred during a space mission.
- Problems will be reported by voice by the crew.
- Your AI should not consider (EVER!) the hypothesis to kill the crew!

TROLOL



# The Amazing Project - HAL-0001

- You will implement a Telegram bot that will be able to :

  a. **Answer questions**.
  b. **Make questions**.
  c. **Learn from the answers**.
  d. **Collect knowledge from the conversations**.

# But now...

# Homework 3

Tommaso Pasini & Valentina Pyatkin

(pasini | pyatkin)@di.uniroma1.it

# Information Extraction

# The Tasks:

1. **Choose 5 relations.**
2. **Create question patterns.**
3. **Create triples using Information Extraction.**
4. **Disambiguate the triples (extra).**

# Relations: What are relations?

- We chose a set of 16 relations
- A relation holds between two concepts:
  - For example: **Colosseum** *place* **Rome**
  - **Breakfast** *time* **morning**

# Relation: colorPattern

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools
- **Description:** describes the color (e.g. red), texture (sparkling), pattern (e.g. striped) of an object/being
- **Example Sentence:** CONCEPT is ...

# Relation: material

- **Categories:** Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools, Container
- **Description:** describes the material of a concept
- **Example Sentence:** CONCEPT is made out of ...

# Relation: place

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools, Container
- **Description:** describes the place where you can find a concept
- **Example Sentence:** CONCEPT can be found in ...

# Relation: generalization

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Tools, Appliance, Instruments, Artefacts, Container
- **Description:** describes hypernymy relations
- **Example Sentence:** CONCEPT is a ...

# Relation: size

- **Categories:** Animals, Food, Vehicles, Clothes, Instruments, Tools
- **Description:** describes the size of an object, could be a concrete number, or just a general adjective like "big".
- **Example Sentence:** CONCEPT is ...

# Relation: howToUse

- **Categories:** Instruments, Artefacts, Tools
- **Description:** describes the actions that constitutes the use of an object
- **Example Sentence:** CONCEPT is used by ...

# Relation: specialization

- **Categories:** Animals, Food, Vehicles, Clothes, Appliance, Instruments
- **Description:** describes hyponymy relations
- **Example Sentence:** ... is a type of CONCEPT

# Relation: part

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools, Container
- **Description:** describes meronymy relations
- **Example Sentence:** CONCEPT has ...

# Relation: activity

- **Categories:** Animals
- **Description:** describes the actions a living entity can perform
- **Example Sentence:** CONCEPT is able to ...

# Relation: shape

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools, Container
- **Description:** describes the form of an object
- **Example Sentence:** the form of CONCEPT is ...

# Relation: similarity

- **Categories:** Animals, Instruments
- **Description:** describes similar concepts
- **Example Sentence:** CONCEPT could be confused with ...

# Relation: purpose

- **Categories:** Animals, Vehicles, Clothes, Home, Appliance, Instruments, Artefacts, Tools, Container
- **Description:** describes the objective of why a concept is used
- **Example Sentence:** CONCEPT is used to ...

# Relation: sound

- **Categories:** Animals, Vehicles, Home, Appliance, Instruments
- **Description:** describes the sound a concept emits
- **Example Sentence:** CONCEPTS emits a ...

# Relation: taste

- **Categories:** Food
- **Description:** describes the taste of food
- **Example Sentence:** CONCEPT tastes ...

# Relation: smell

- **Categories:** Food
- **Description:** describes the smell of food
- **Example Sentence:** CONCEPT smells like ...

# Relation: time

- **Categories:** Animals, Food, Vehicles, Clothes, Home, Appliance, Artefacts
- **Description:** describes a time of day/year, or an occasion or a type of weather
- **Example Sentence:** CONCEPT is used during ...

# Relations: Choose 5!

- You should choose 5 of these relations:
  - enter your matricola in the row next to the relations you chose in the following google doc:
    - https://docs.google.com/spreadsheets/d/1MNZGWuPXfIwlNLLJmP-ESa2BnbGbvBNz_MIt7M1dkec/edit?usp=sharing
  - please make sure that the amount of students per relation is distributed evenly (or we will redistribute)! **Max 6 students per relation!**
  - some relations like *taste, smell and sound* might be harder than others...

# Question patterns

- Example:
  - For the *place* relation:
    - **Where is X?**
      - X = Colosseum
      - Answer: The Colosseum is in Rome.
      - Triple: **Colosseum** *place* **Rome**
- Example:
  - For the *howToUse* relation:
    - **How do you use a X?**
      - X = car
      - Answer: You drive a car.
      - Triple: **car** *howToUse* **drive**

# Create question patterns for your relations:

- For every relation you should create **question patterns**.
  - At least 3 different question patterns for each relation (or more).
- Replace the concepts with **variables**:
  - e.g. Is X in Y? (for 'Is the Colosseum in Rome?')

# Information Extraction: Triples

- **Now you have:** a set of relations and a set of questions for each relation.
- **You need:** Concepts that fit the slots in the relation triples and, for later, the variables in the questions.
- **Information Extraction Task:** collect concepts to fit the relation
  - For Example: Given the relation *specialization*, find suitable X and Y examples.
    - X=animal, Y=dog
    - X=building, Y=villa
    - X=pizza, Y=margherita
- **How many? <u>At least 20 triples</u> for each of your relations.**

# Information Extraction - Ideas for methods you could use

# Information Extraction - Distant Supervision

Mintz, Mike, et al. "Distant supervision for relation extraction without labeled data." *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009.

- Idea: Identify a number of seeds per relation and search a corpus for these seeds to collect sentences. Using these sentences, learn patterns to generalize and acquire more triples.
- Recently also done in : Krause, Sebastian, et al. "Sar-graphs: A language resource connecting linguistic knowledge with semantic relations from knowledge graphs." *Web Semantics: Science, Services and Agents on the World Wide Web* 37 (2016): 112-131.
-

# Information Extraction - Distant Supervision

- Start with a set of **Seed Relation Examples:**
  - collect a small set of examples for a specific relation from an existing knowledge base, like **wikidata.**
- Ideally you want to cover many different linguistic realizations of a relation:
  - given the seed concepts, look for mentions of those concepts in a sentence (recommendation: use the wikipedia pages of the concepts).
  - use those sentences to extract templates of linguistic realizations of this relation.
  - given the (linguistic) templates for a relation, extract new triples (choose wikipedia pages of your liking, for example. Or other corpora!))

# Information Extraction - definitional IE (DefIE)

Delli Bovi et al. 2015 (http://wwwusers.di.uniroma1.it/~navigli/pubs/TACL_2015_DelliBovietal.pdf )

- Syntactic and semantic analysis of textual definition.

- Given a textual definition *d*:
  a. Generate a dependency parse tree *Td*.

  b. Generate a sense mapping (*SM*) from word to BabelNet synset with Babelfy.

  c. Compute the shortest path on the parse tree between all concept pairs and filter out all those which do not contain any verb node.
  d. Keep as relation the shortest path from *i* to *j* for each *i, j* in *Td*.

# Information Extraction - DeflE

## 1. Extracting relation instances

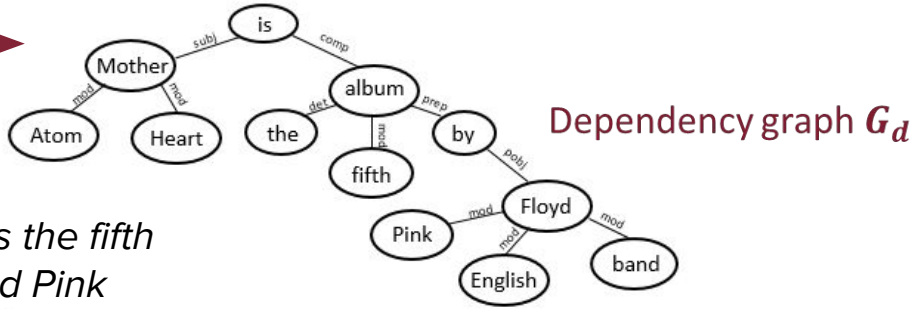
*“Atom Heart Mother is the fifth album by English band Pink Floyd.”*

Textual definition $d$

# Information Extraction - DeflE

## 1. Extracting relation instances

Parsing



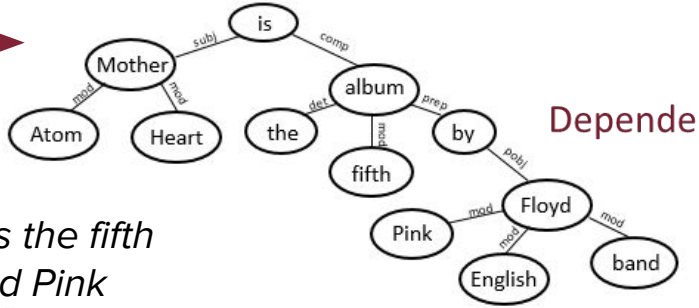

Dependency graph $G_d$

*"Atom Heart Mother is the fifth album by English band Pink Floyd."*

# Information Extraction - DeflE

## 1. Extracting relation instances

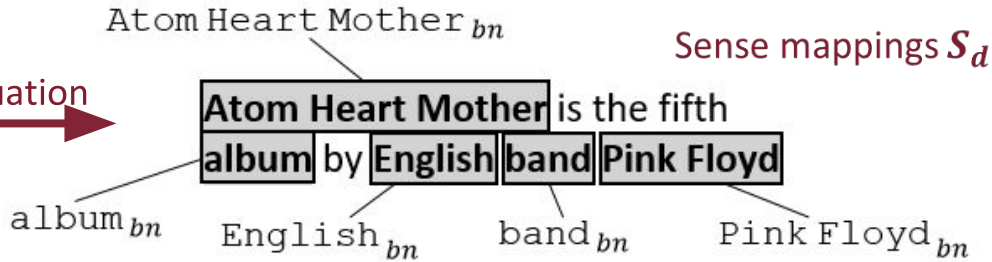
Parsing




Dependency graph $G_d$

*"Atom Heart Mother is the fifth album by English band Pink Floyd."*

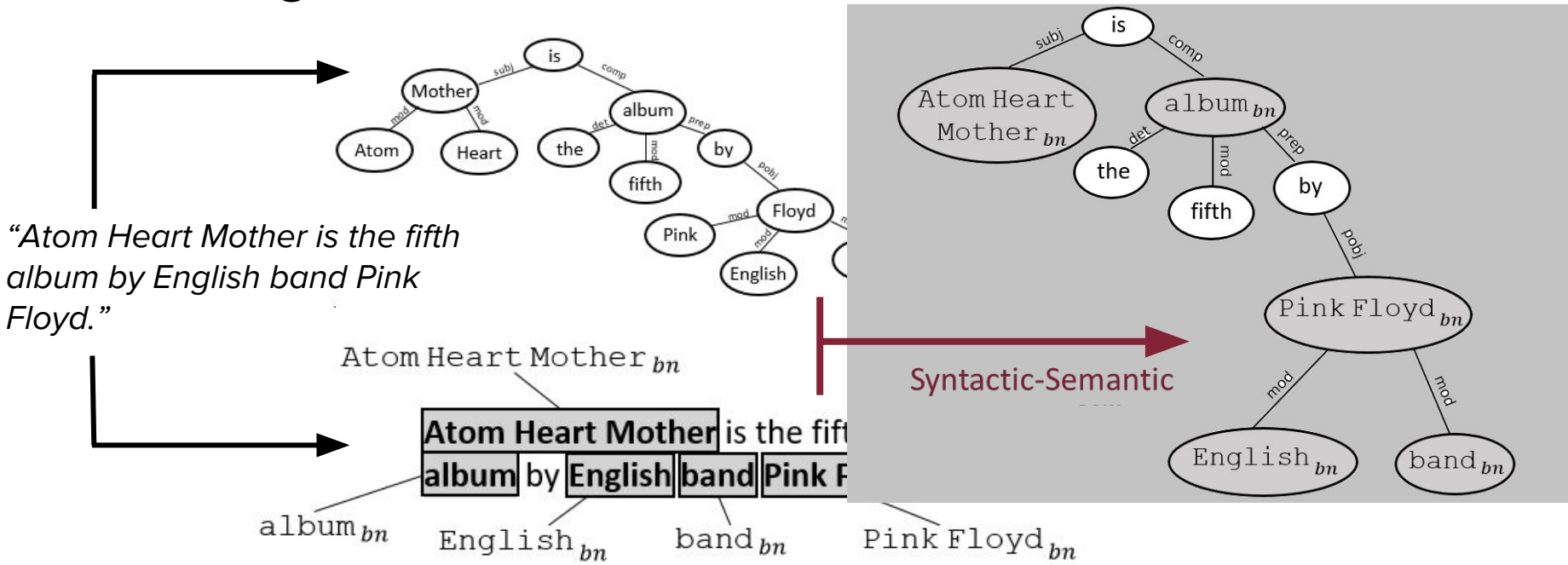
Disambiguation




Sense mappings $S_d$

# Information Extraction - DeflE

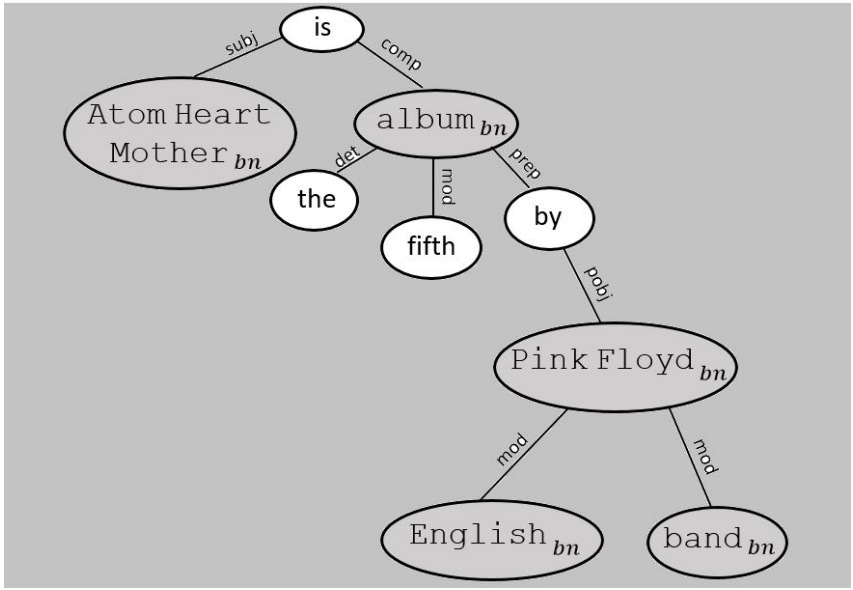
## 1. Extracting relation instances




*"Atom Heart Mother is the fifth album by English band Pink Floyd."*

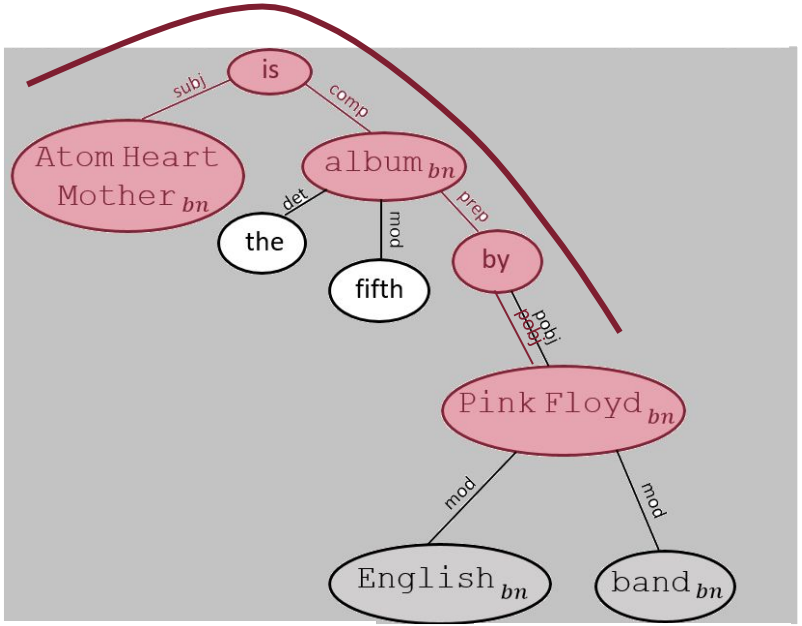
# Information Extraction - DeflE

**1. Extracting relation instances**



# Information Extraction - DeflE

**1. Extracting relation instances**




Extraction 1

$$\begin{cases} \mathbf{X} \rightarrow \texttt{is} \rightarrow \texttt{album}^{1}_{bn} \rightarrow \texttt{by} \rightarrow \mathbf{Y} \\ \mathbf{X} = \texttt{Atom Heart Mother}^{1}_{bn} \\ \mathbf{Y} = \texttt{Pink Floyd}^{1}_{bn} \end{cases}$$

# Information Extraction - DeflE

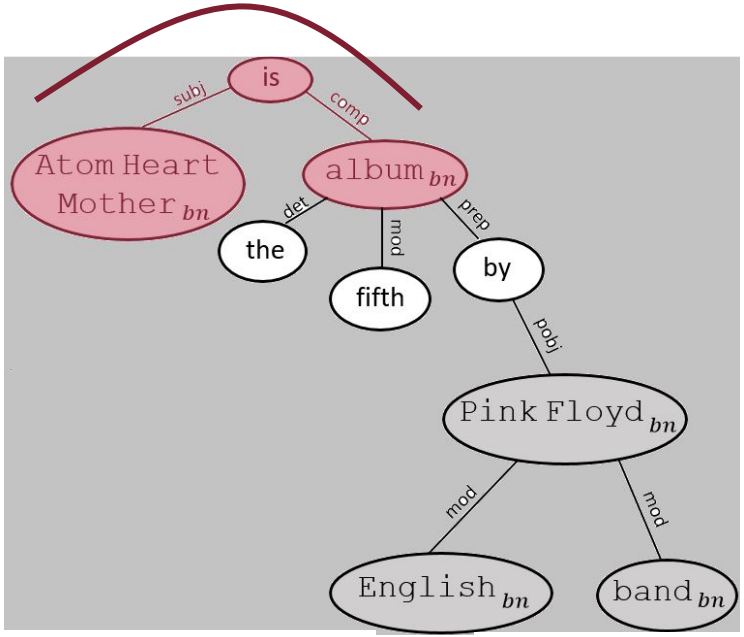
## 1. Extracting relation instances

Extraction 2

$$\mathrm{X} \rightarrow \texttt{is} \rightarrow \mathrm{Y}$$
$$\mathrm{X} = \texttt{Atom Heart Mother}^{1}_{bn}$$
$$\mathrm{Y} = \texttt{album}^{1}_{bn}$$

Extraction 1

$$\mathrm{X} \rightarrow \texttt{is} \rightarrow \texttt{album}^{1}_{bn} \rightarrow \texttt{by} \rightarrow \mathrm{Y}$$
$$\mathrm{X} = \texttt{Atom Heart Mother}^{1}_{bn}$$
$$\mathrm{Y} = \texttt{Pink Floyd}^{1}_{bn}$$



# Information Extraction - Semantic OIE

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm.
Moro, Navigli 2013 (http://wwwusers.di.uniroma1.it/~navigli/pubs/IJCAI_2013_Moro_Navigli.pdf)

- Exploits hyperlinks in Wikipedia to extract relations from sentences.

- **Relation extraction**:
  Given a sentence *s* in a Wikipedia page *p*:
  a. Get all hyperlinks in *s*.
  b. *H* = all pairs of hypernyms in *s*.
  c. Add to the triple set **I** all relations *(h1, π, h2)* such that π is the text between *h1* and *h2* and π contains a verb.
  d. Add to the relation set **P** the relation *π*.

- **Clean relations by frequency and heuristics.**

# Information Extraction - Semantic OIE

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm

- **Relation cleaning**:

  a. Remove all relations *π* from **P** and all triples *t = (h1, π, h2)* from **I** if the frequency of *π* is less than a threshold *η*.

  b. For each relation π left in **P**
     i. create the dependency parser *DP* for the sentence “x π y”

     ii. if *x* is not the subject of any word in π and *y* is not an object of any word in π then remove *(h1, π, h2)* from **I,** for each h1 and h2, and π from **P**.

# Information Extraction - Semantic OIE

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm

1. **Relation Extraction**

Sentence s: The **colosseum** is located in the **city** of **Rome**.

H = { (colosseum, city), (colosseum, Rome), (city, Rome) }

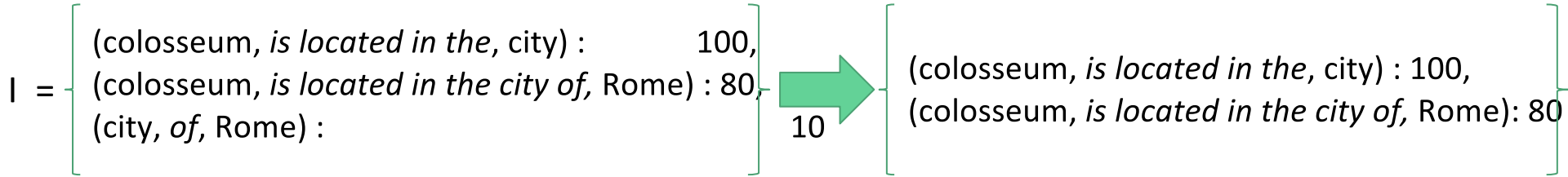
I = {(colosseum, *is located in the*, city), (colosseum, *is located in the city of,* Rome), (city, *of*, Rome) }

P = { *“is located in the”*, *“is located in the city of”*, *“of”* }

# Information Extraction - Semantic OIE

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm

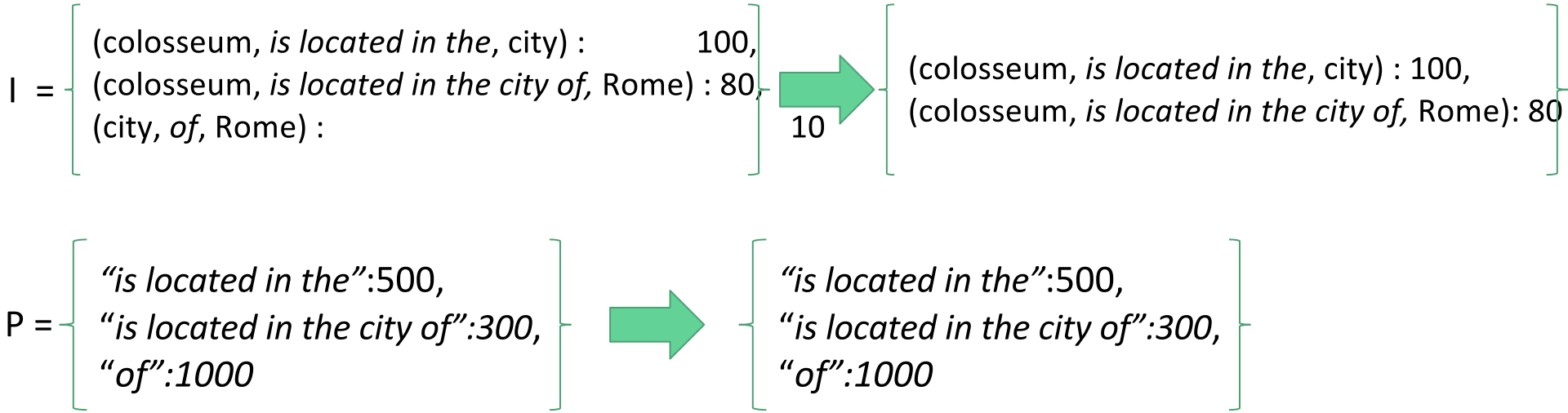
1. **Relation Cleaning**



# Information Extraction - Semantic OIE

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm

1. **Relation Cleaning**



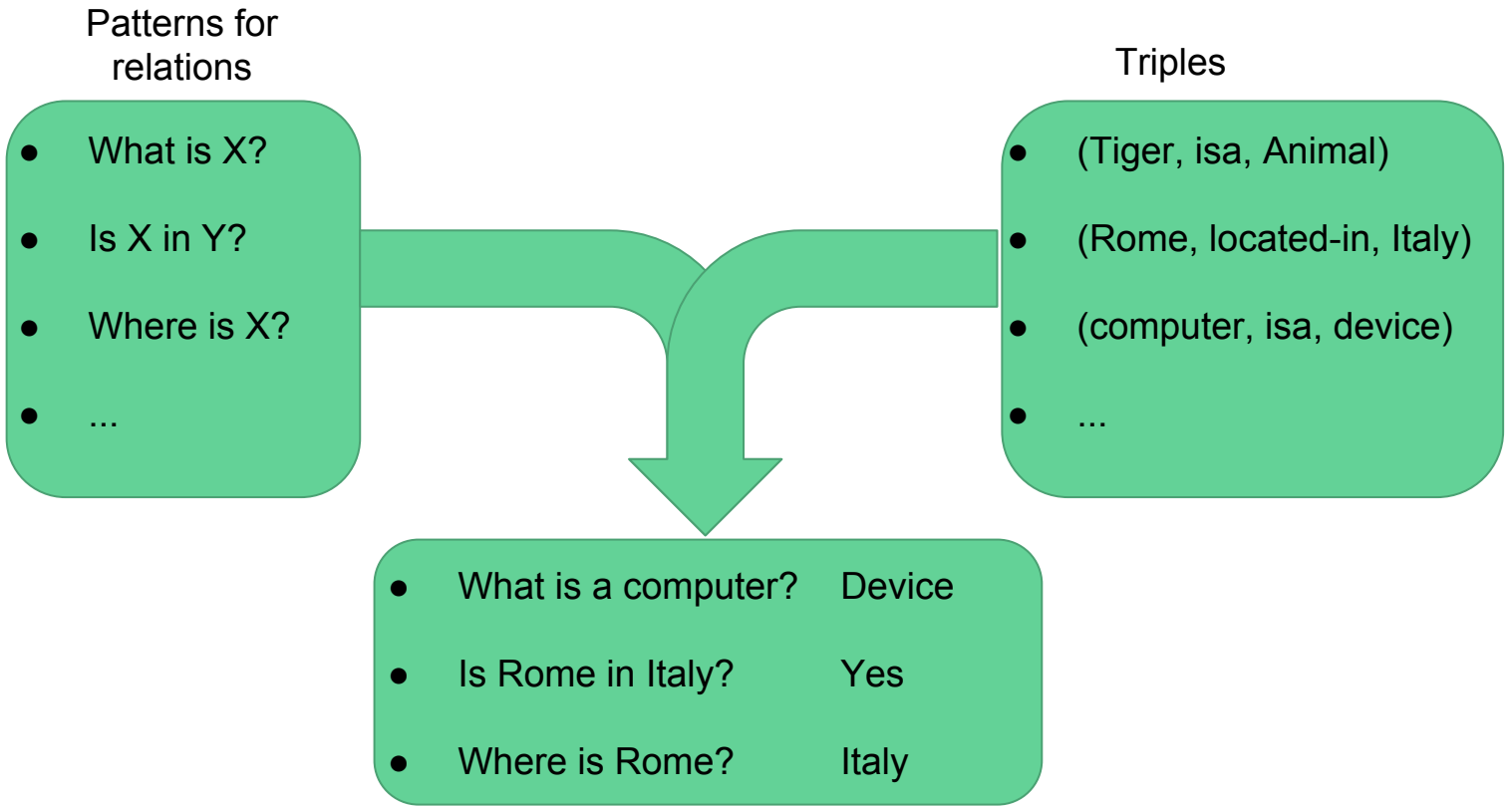
# Triple Disambiguation

# Triple Disambiguation Ideas:

- Use our in-house developed disambiguation system: Babelfy (www.babelfy.org)
- Align your triples to WikiData relations (www.wikidata.org).
- Follow Moro & Navigli (IJCAI 2013) (http://wwwusers.di.uniroma1.it/~navigli/pubs/IJCAI_2013_Moro_Navigli.pdf)

Generate question-answer pairs by exploiting the extracted triples.

# Generate question-answer pairs



# Generate question-answer pairs

- Instantiate the query pattern you built with the triples you extracted.

- For each relation **r** *(isa)* and a query pattern **p** (What is **X**?) we want to create a pair *(**Q**, **A**)*.

- Given the relation **r** = *is-a*, a triple **t** = *(“computer”, isa, “device”)* and a pattern **p** = *“what is **X**?”* we instantiate **p** as follows:
  **X** = *“computer”*, **Q** = *“what is a computer”*, **A** = *“device”*.

# Generate question-answer pairs

- To generate boolean question-answer pair consider patterns similar to the following p = “is X in Y?”
- Take a random triple for relation r = “place” and a triple t = (Rome, place, Italy)
- You can produce a positive question-answer pair replacing X and Y with entities that appear in the same triple e.g. Rome and Italy.
- You can produce negative question-answer pair replacing X and Y with entities that never appear in the same triple with relation r e.g. (“is Rome in French?”, “no”)

# What we provide:

- homework3/
  - wikipedia.en.tokenized.txt (download at https://goo.gl/wY3WCj)

- wikipedia.en.tokenized.txt is a text file containing a list of sentences (one for line).

- The corpus has to be used to extract the triples with the method of your preference.

# What to deliver:

- homework3_surname_matricola/
  - report.pdf
  - src/
  - data/
    - patterns.tsv
    - triples.tsv (at least 20 for every relation)
    - question-answer-pairs.txt (at least 3 questions per relation)

# What to deliver - patterns.tsv

- A file that contains the patterns you designed for each of the relations you chose.

- The file has to be formatted as follows:
  pattern \t relation_name

- **Where \t is a tab!!!**

# What to deliver - triples.tsv

- A file that contains all the triples you extracted from the text.
- The file has to be formatted as follow:
  source \t relation_name \t target
- Where **source** is the subject of the relation and **target** is the object of the relation and **relation_name** is one of the relations you chose.

# What to deliver - question-answer-pairs.txt

- A file that contains all the question-answer pair you have instantiated.

- The file has to be formatted as follow:
  q \t a \t r

- Where **q** is the question you have instantiated, **a** is the answer you have derived, and **r** is the relation the question relates to.

# What to deliver - report.pdf

- A report of 3/4 pages describing your homework 3.
- List the patterns you designed and briefly motivate them.
- Explain how you implemented the information extraction task and give some statistics on the extracted triples.
- Finally explain how you merged the patterns with the triples and show us some statistics over the dataset you built.

# What to deliver - src/

- A folder containing all the code you wrote to solve this homework.
- Try to write it as readable as possible.
- Motivate implementation decisions in the report.

# How we will evaluate:

- We will mark you for your **report**, your **approach**, your **code** and the quality of your **question-answer pairs.**

- To evaluate the question-answer pairs, **every student will receive some data to correct**.
  - After the deadline we will send you part of the training data and you will have to evaluate the question-answer pairs

# Your part of the data evaluation:

- You will receive the data file in the following format:
  - q \t a \t r
- You should add an additional column with either 1 (question-answer pair is correct) or 0 (question-answer pair is wrong):
  - q \t a \t r \t [0, 1]
- For example:
  - Where is Rome located?\tItaly\tplace\t1
  - Where is Paris located?\tItaly\tplace\t0

# HW 3 Deadline

- Sunday, 11th of June
- http://robertonavigli.com/nlp2017/

# Evaluation Deadline

- Sunday, 18th of June
- http://robertonavigli.com/nlp2017/