Improving the Machine Interpretation of Internet Posts

Part 2

Extraction of a lightweight, domain independent semantic network from the Wikipedia categorization system

Università degli Studi di Pavia

CVMLab

Giacomo Parigi

14.02.14

Computer Vision and Multimedia Lab

Objectives

1. Overview of the topic
2. Identification of the specific set of problems
3. Show the reasons of the chosen approach
4. Illustrate the solutions implemented (or designed)

Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

1. Knowledge Representations I

"A subarea of Artificial Intelligence concerned with understanding, designing, and implementing ways of representing information in computers so that programs (agents) can use this information

- *to derive information that is implied by it,*
- *to converse with people in natural languages,*
- *to decide what to do next*
- *to plan future activities,*
- ***to solve problems in areas that normally require human expertise."***

(Stuart C. Shapiro)

- How is it made?
  - A *Knowledge Base*

    The machine-readable codification of the information
  - A set of *Inference Rules*

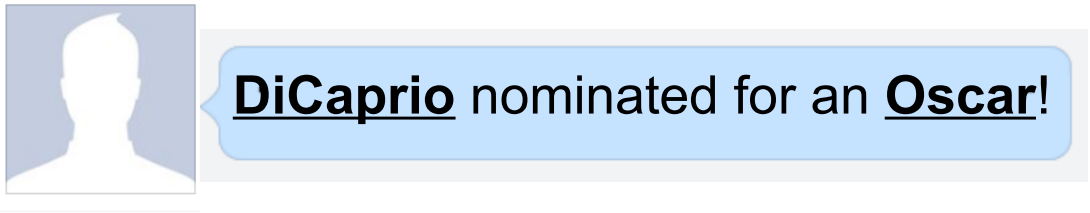
    The *reasoning* methods, determined by the purpose

1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)

1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)



1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)

DiCaprio nominated for an **Oscar**!

Leonardo DiCaprio

"DiCaprio" redirects here. For other uses, see DiCaprio (disambiguation).

Leonardo DiCaprio

DiCaprio at the premiere of *Shutter Island* at the 60th Berlin Film Festival

Born	Leonardo Wilhelm DiCaprio November 11, 1974 (age 39) Los Angeles, California, U.S.
Occupation	Actor, film producer

Cinema of the United States

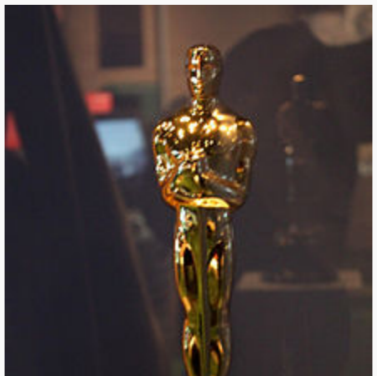
Cinema of the United States

Academy Award

"Oscars" and "The Oscar" redirect here. For the film, see The Oscar (film). For other uses of the word "Oscar", see Oscar (disambiguation).

The Oscars

86th Academy Awards



1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)

DiCaprio nominated for an **Oscar**!

George DiCaprio

George Paul DiCaprio (born November 30, 1943) is an American comic book writer, editor and major West Coast underground comic book distributor.

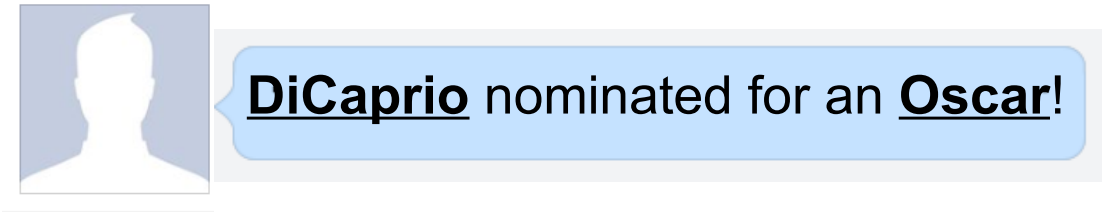
DiCaprio was born to George Leon DiCaprio (May 14, 1902- November 18, 1965),[1] and a German-born mother, Olga Anne Jacobs (December 23, 1904- May 16, 1984).[2][*verification needed*][3][4][5][*unreliable source?*]

George DiCaprio's grandfather named Salvatore DiCaprio (February 18, 1866- August 1966), was from Alife, Caserta, Campania, Italy. He immigrated to the United States in 1882 and married Rosina Cassella in 1893 in Manhattan, New York. They had six children, one of them

- Oscar Wilde
- Oscars
- Oscar Hammerstein II
- OSCAR
- Oscar Peterson
- Oscar II of Sweden
- Oscar
- Oscar De La Hoya
- Óscar Hernández (tennis)

1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)



- What kind of inference rules do we need?
  - Word Sense Disambiguation

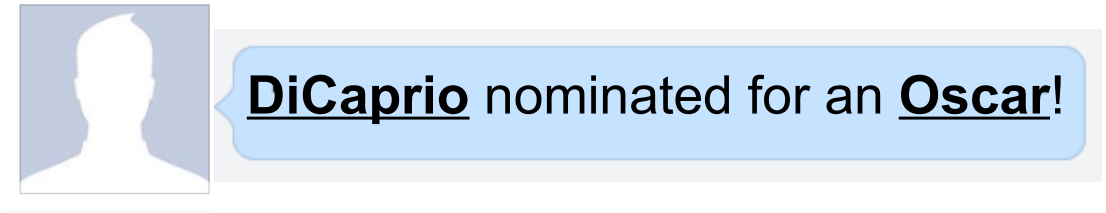
    Disambiguate the meaning of words in context in a computational manner

    AI-Complete problem
  - Named Entity Recognition & Classification

    Identify atomic units in test and classify them into predefined categories

1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)



- What kind of inference rules do we need?
  - Word Sense Disambiguation

    Disambiguate the meaning of words in context in a computational manner

    AI-Complete problem
  - Named Entity Recognition & Classification

    Identify atomic units in test and classify them into predefined categories

...Spoiler:

  - Neither of them is enough!

1. Knowledge Representation III: WSD & NERC Methods

- Knowledge Bases
  - Structured

    Thesauri, machine-readable dictionaries, taxonomies and ontologies
  - Unstructured

    Raw or sense-annotated corpora, lists, other...

1. Knowledge Representation III: WSD & NERC Methods

- Knowledge Bases
  - Structured

    Thesauri, machine-readable dictionaries, taxonomies and ontologies
  - Unstructured

    Raw or sense-annotated corpora, lists, other...

- Methods
  - Supervised and semi-supervised

    Hand-crafted rules or bootstrapping methods

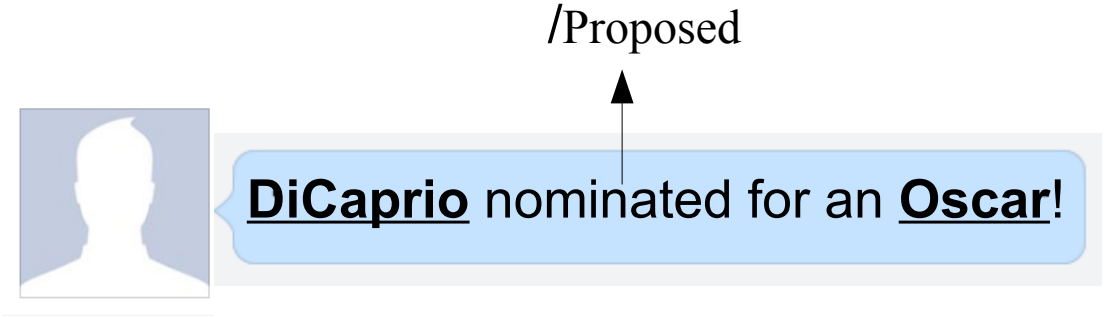
    Knowledge Acquisition Bottleneck
  - Unsupervised

    Avoid the bottleneck, but provide only word clustering
  - Knowledge based

    Wide semantic knowledge for context extraction, word statistics

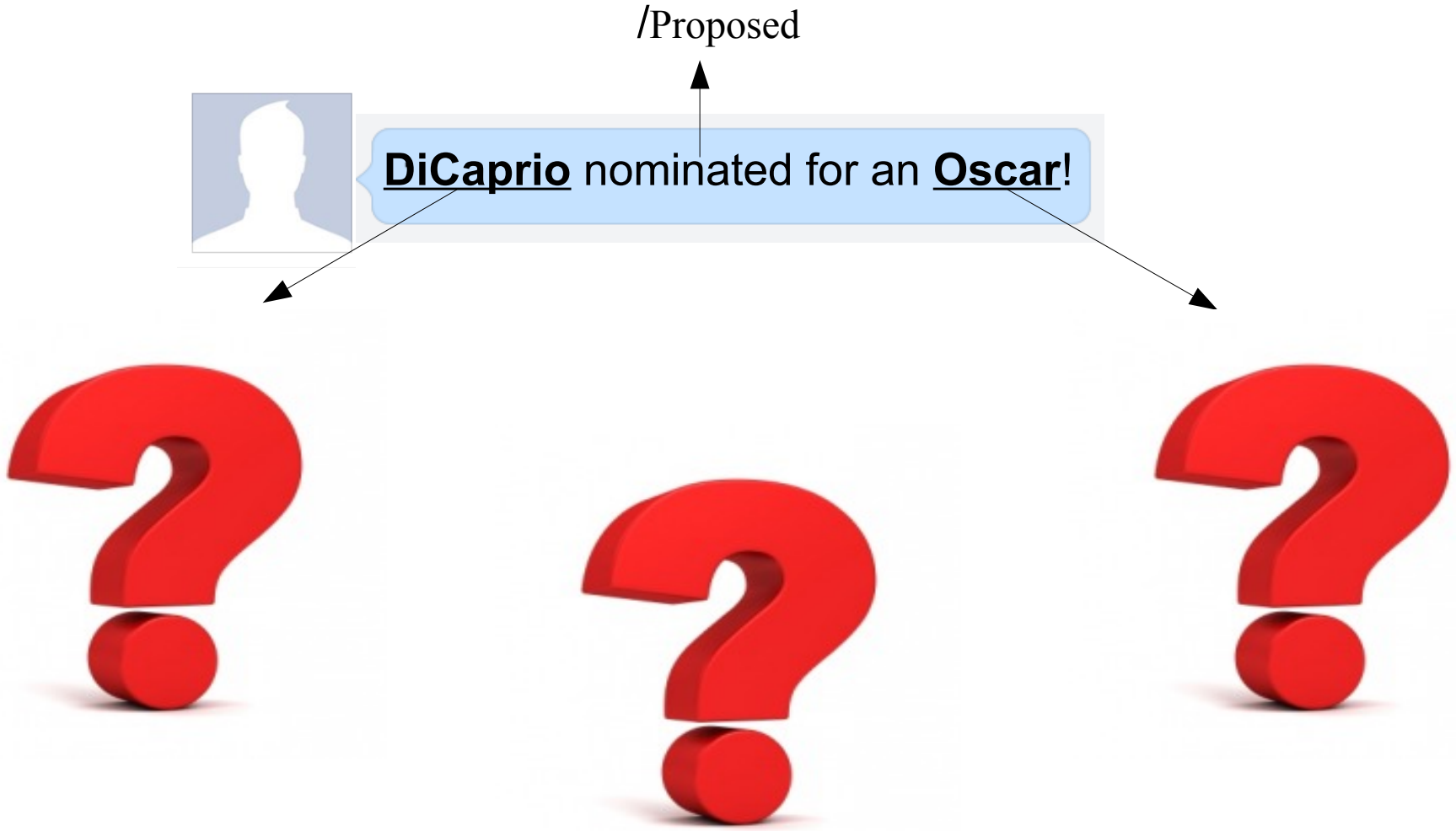
1. KR IIII: WSD & NERC, What about our purpose?

- Word Sense Disambiguation



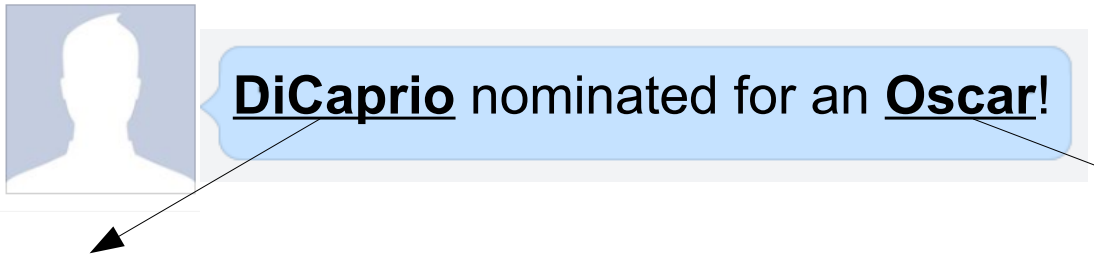
1. KR IIII: WSD & NERC, What about our purpose?

- Word Sense Disambiguation



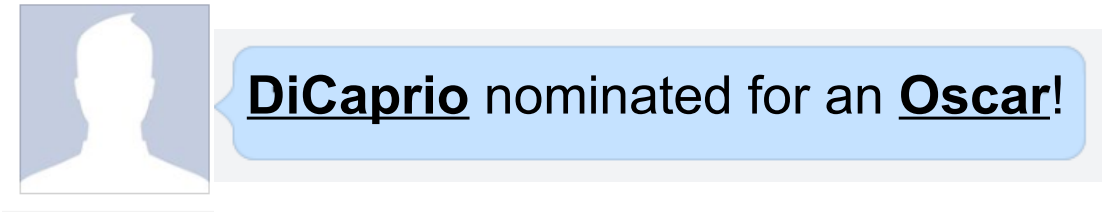
1. KR IIII: WSD & NERC, What about our purpose?

- Named Entity Recognition & Classification



1. Knowledge Representations II

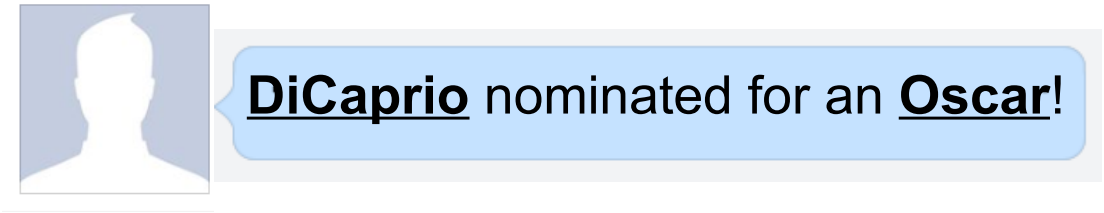
- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)



- What does it imply?
  - Reduced and human-readable information
  - Little or no *near-words context*
  - Extremely wide possible domain
  - Typing errors and carelessness
  - Omitted information about shared or collective background

1. Knowledge Representations II

- Purpose of our system
  - Identify context and entities in user-generated contents (posts, forums...)



- Proposed solution
  - Reduced and human-readable information, little or no *near-words context,* Extremely wide possible domain

    Knowledge-based system
  - Typing errors and carelessness

    Mixed NERC and WSD methods
  - Omitted information about shared or collective background

    Integration with *a priori* information (e.g. word frequency)

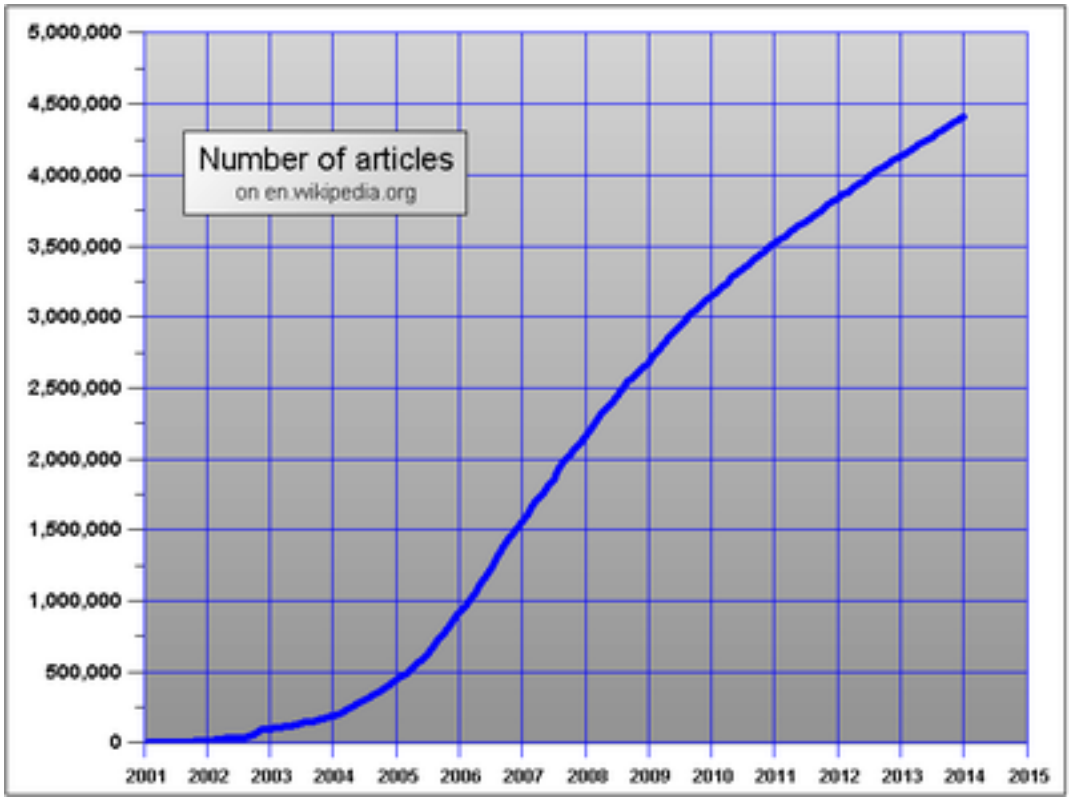
Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

5. Wikipedia I: Why Wikipedia?

- Main competitors
  - WordNet → Hand-crafted lexical database, no named entities
  - ResearchCyc → Hand-crafted ontology, multi-domain breadth, out-of-date
  - Wikipedia → Crowd-crafted database, domain-independent, multilingual, captures “common sense”



2. Wikipedia II: The Categorization System

- Guidelines
  - Group similar articles
  - Balance category breadth

    n° of sub-categories/sub-pages w.r.t. hierarchic level
  - Avoid cycles
  - Every article should belong to at least one category
  - Article inclusions should be based only on *defining* characteristics

2. Wikipedia II: The Categorization System

- Guidelines
  - Group similar articles
  - Balance category breadth

    n° of sub-categories/sub-pages w.r.t. hierarchic level
  - Avoid cycles
  - Every article should belong to at least one category
  - Article inclusions should be based only on *defining* characteristics

- Folk taxonomy (or *Folksonomy)*
  - Not a strict and logically grounded ontology
  - Inconsistencies
  - Loose definition of relationships

2. Wikipedia II: The Categorization System

- Guidelines
  - Group similar articles
  - Balance category breadth

    n° of sub-categories/sub-pages w.r.t. hierarchic level
  - Avoid cycles
  - Every article should belong to at least one category
  - Article inclusions should be based only on *defining* characteristics

- Folk taxonomy (or *Folksonomy)*
  - Not a strict and logically grounded ontology
  - Inconsistencies
  - Loose definition of relationships
  - **Reflects our intuitions about classification and organization**

2. Wikipedia III: The Category Tree Organization

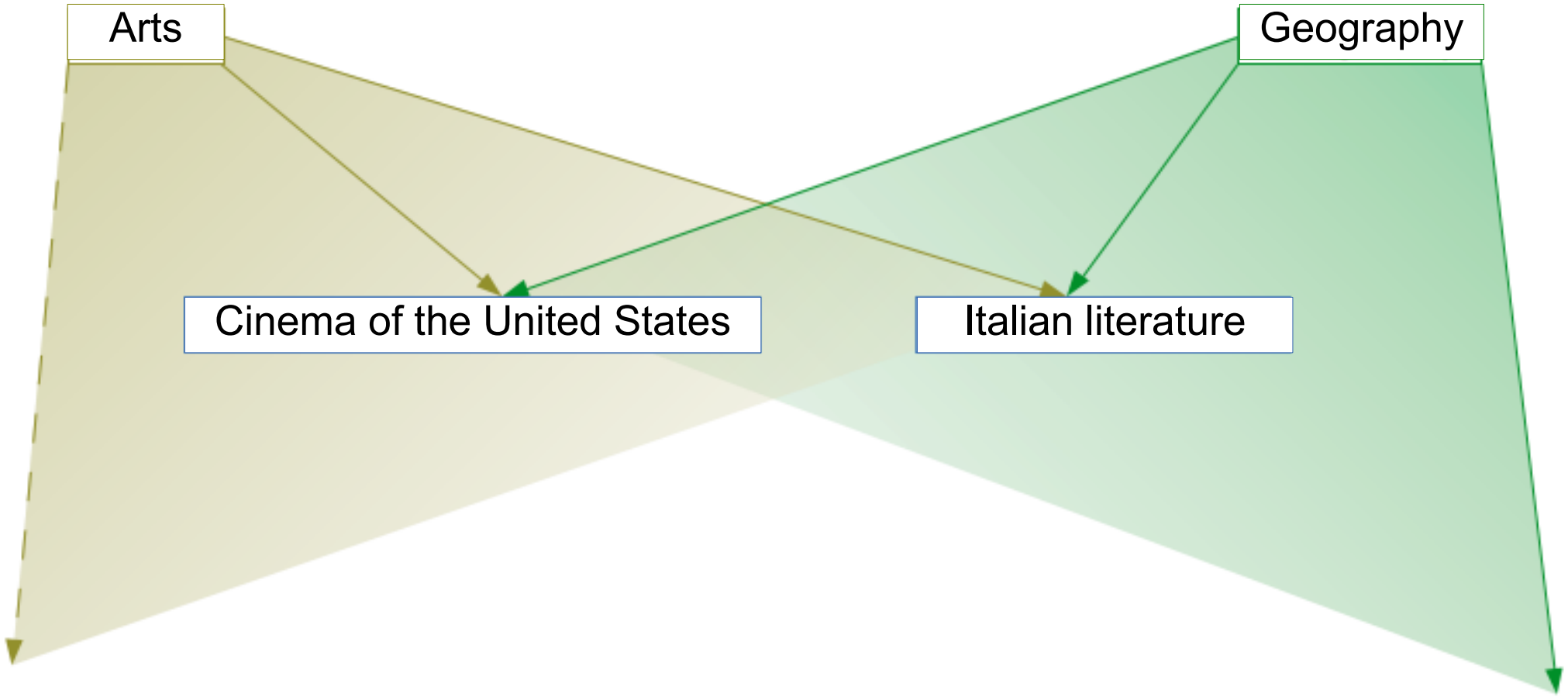
- Many overlapping trees

Arts

Geography

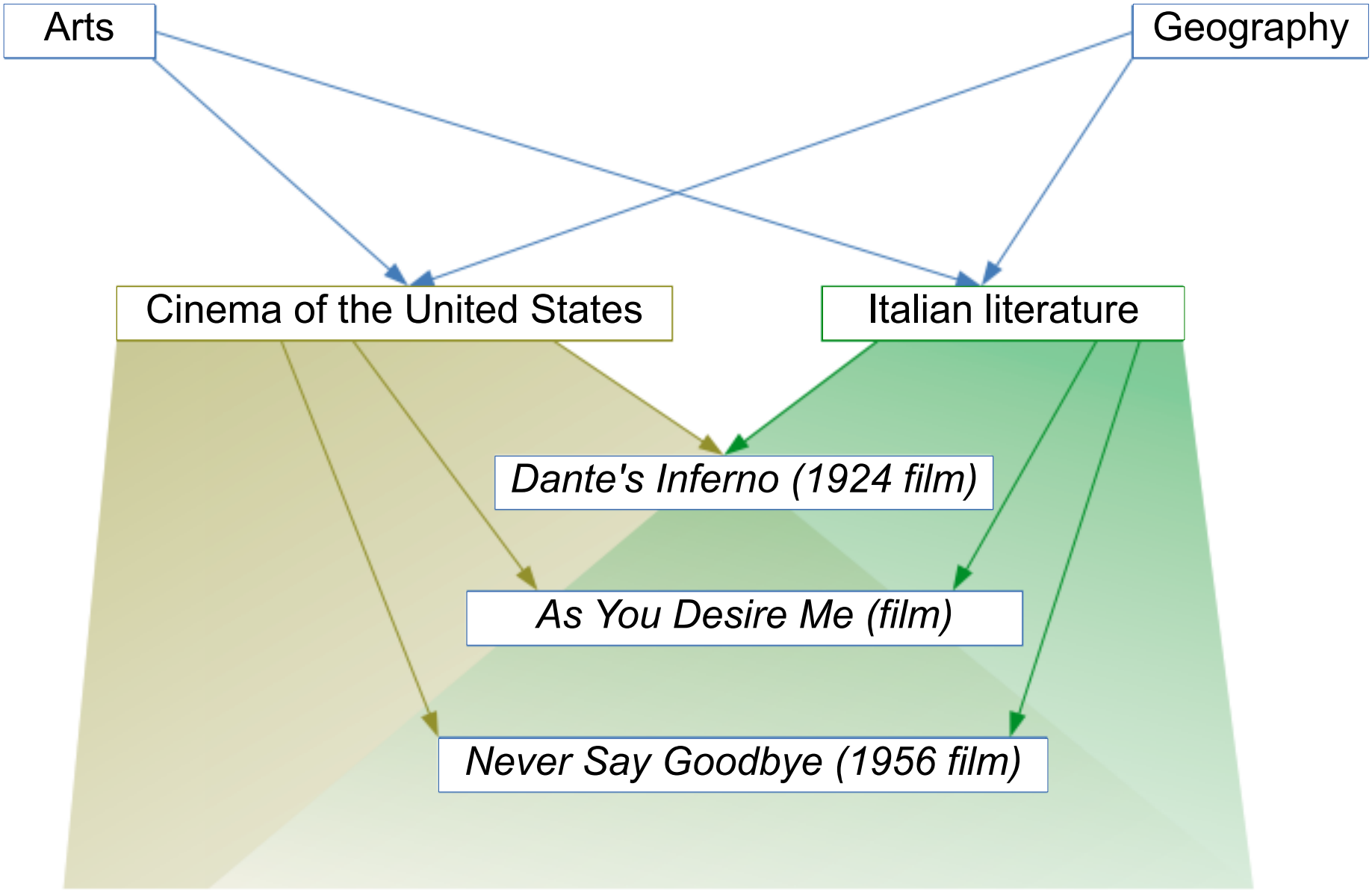
2. Wikipedia III: The Category Tree Organization

- Many overlapping trees



2. Wikipedia III: The Category Tree Organization

- Many overlapping trees (at each hierarchical level)



2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

- Diffuse large categories

  Albums → Albums by artist → *Artistname* albums

2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

- Diffuse large categories

  Albums → Albums by artist → *Artistname* albums

- Non-diffusing categories

  Film actors → Best Actor Academy Awards winners

2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

- Diffuse large categories

  Albums → Albums by artist → *Artistname* albums

- Non-diffusing categories

  Film actors → Best Actor Academy Awards winners

- Eponymous categories

  France/cat → France/article

2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

- Diffuse large categories

  Albums → Albums by artist → *Artistname* albums

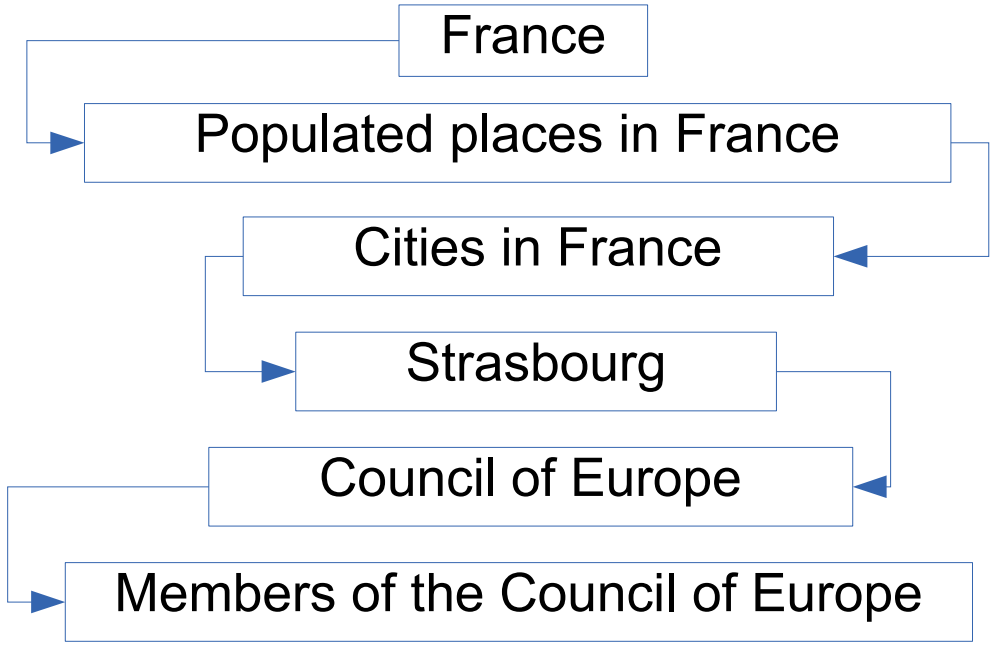
- Non-diffusing categories

  Film actors → Best Actor Academy Awards winners

- Eponymous categories

  France/cat → France/article

  Systematic Error



2. Wikipedia IV: Subcategorization

- Two main kinds

  Topic (Opera) and **Set** (Operas) categories

- Diffuse large categories

  Albums → Albums by artist → *Artistname* albums

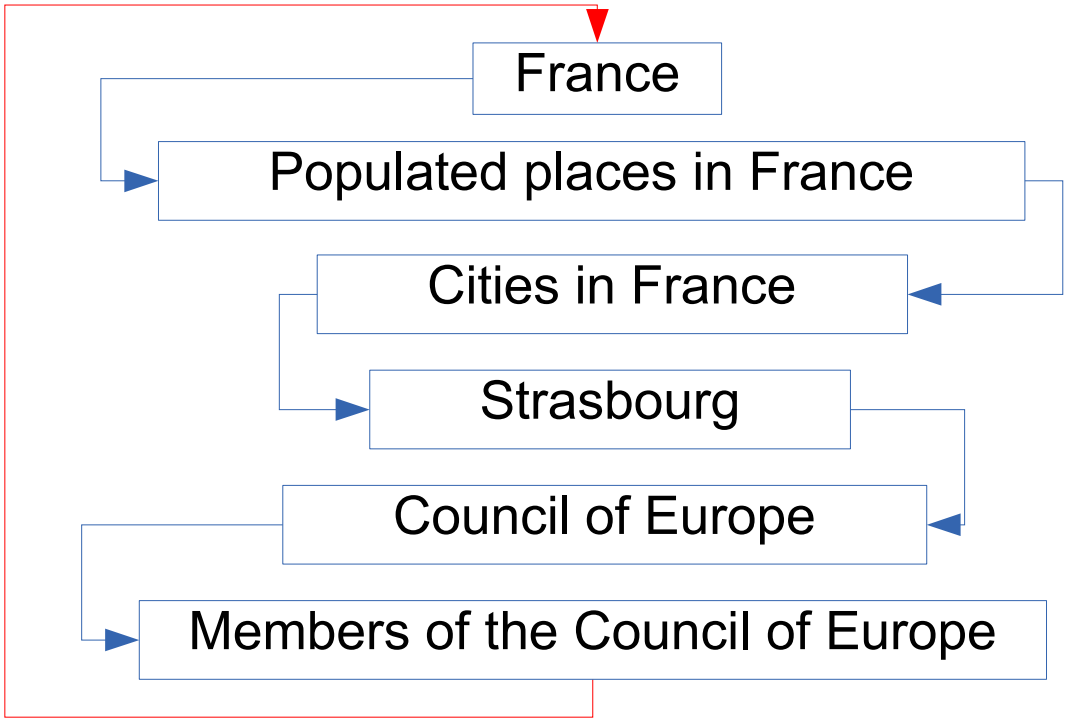
- Non-diffusing categories

  Film actors → Best Actor Academy Awards winners

- Eponymous categories

  France/cat → France/article

  Systematic Error



2. Wikipedia V: Knowledge Extraction Methods

- Natural Language Processing (NLP) methods
  - Based on category and page names

    Part-Of-Speech patterns, word matching
  - Build a new graph from scratch (usually)

    Classes and instances are made by copying or splitting Wikipedia categories

    Links are made anew from the relations found

2. Wikipedia V: Knowledge Extraction Methods

- Natural Language Processing (NLP) methods
  - Based on category and page names

    Part-Of-Speech patterns, word matching
  - Build a new graph from scratch (usually)

    Classes and instances are made by copying or splitting Wikipedia categories

    Links are made anew from the relations found

- Connectivity-based methods
  - Based on properties and “habits” of Wikipedia categorization

    instance and redundant categorization
  - Propagate relations found to sub-categories and sub-pages

Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories

     “Members_of...”, “Presidents_of...”

     [VBN IN] patterns: “...*directed_by...”*, “...*located_in...”*

3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories

  2) Partly explicit relation categories

     Prepositions: *“Villages_in_Brandeburg”*, *“Conflicts_in_2000”*

     Need super-categories to identify the relation (*“Geography”/“Years”)*

3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories

  2) Partly explicit relation categories

  3) Implicit relation categories

     “Mixed martial arts television programs”

3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories

  2) Partly explicit relation categories

  3) Implicit relation categories

  4) “Class attribute” or “Diffusing” categories

     “X_by_Y” patterns

     Grouping of instances of *X* by attribute *Y*

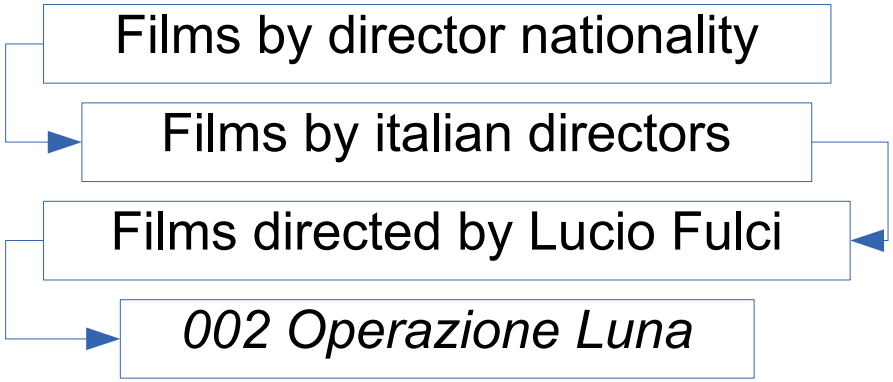
3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories
  2) Partly explicit relation categories
  3) Implicit relation categories
  4) “Class attribute” or “Diffusing” categories
- Relations found propagate to sub-categories and sub-pages

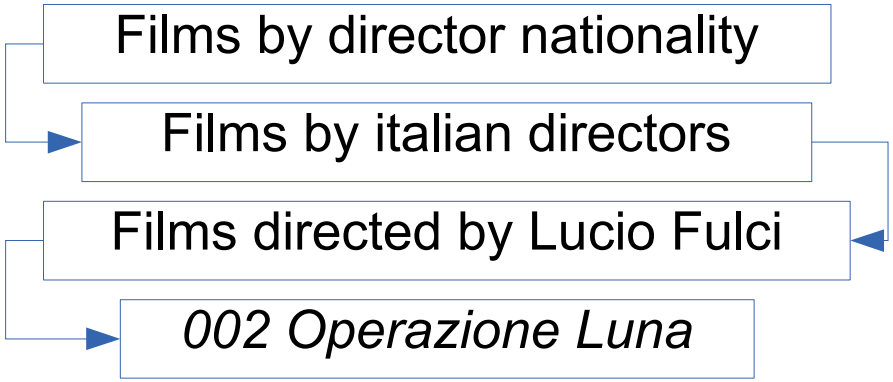
3. Knowledge Extraction Example I: NLP & Connectivity

- Application of hand-crafted rules to category names
  1) Explicit relation categories
  2) Partly explicit relation categories
  3) Implicit relation categories
  4) “Class attribute” or “Diffusing” categories
- Relations found propagate to sub-categories and sub-pages
  - Problem: *“Extinct_cephalopods”* is a subcategory of *“Fashion”* !

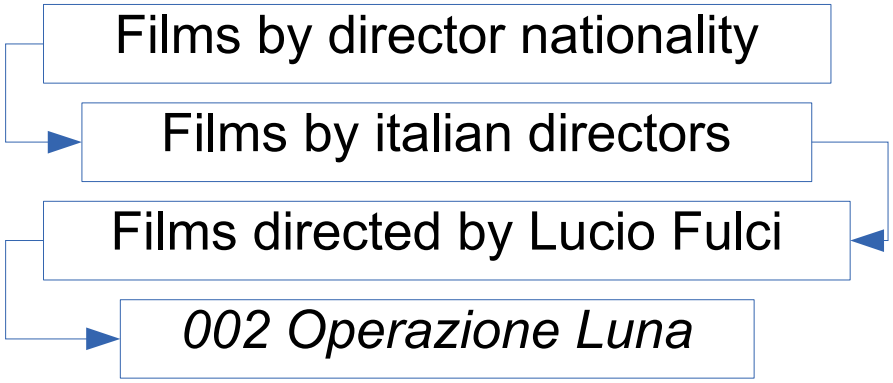
3. Knowledge Extraction Example II: Limitations



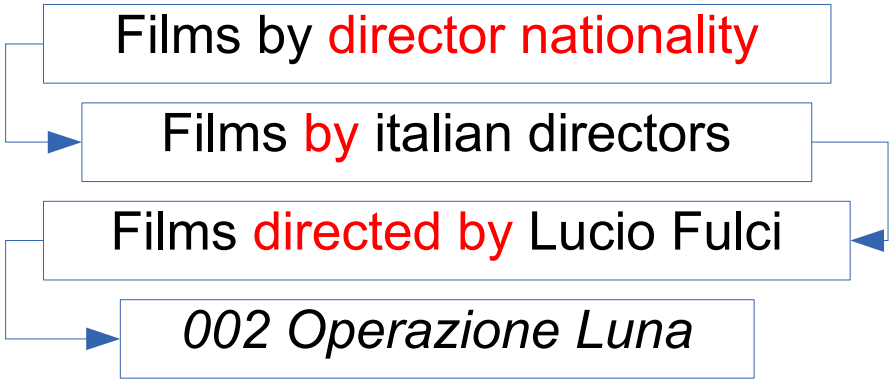
3. Knowledge Extraction Example II: Limitations



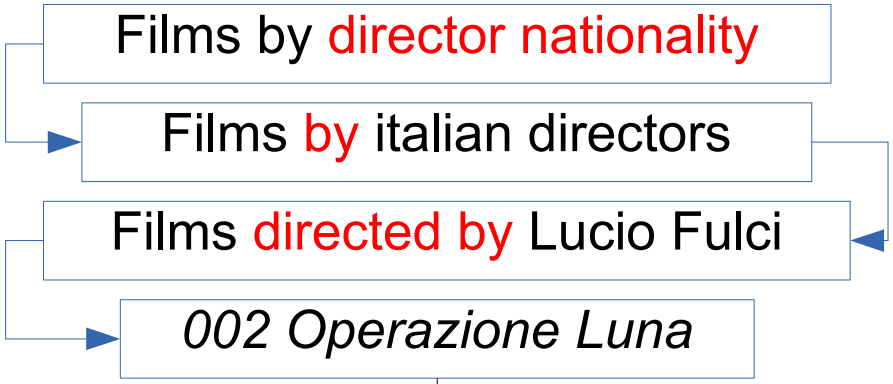
3. Knowledge Extraction Example II: Limitations



3. Knowledge Extraction Example II: Limitations

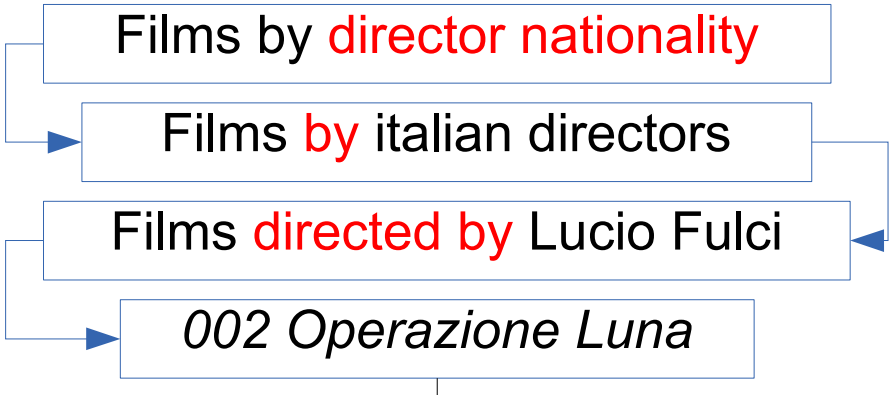


3. Knowledge Extraction Example II: Limitations



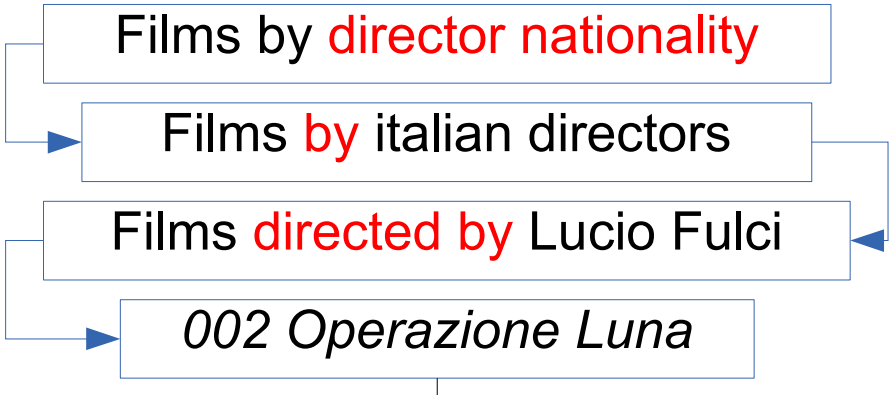
"002_Operazione_Luna", *IS_A*, Director

3. Knowledge Extraction Example II: Limitations



"002_Operazione_Luna", *IS_A*, Director

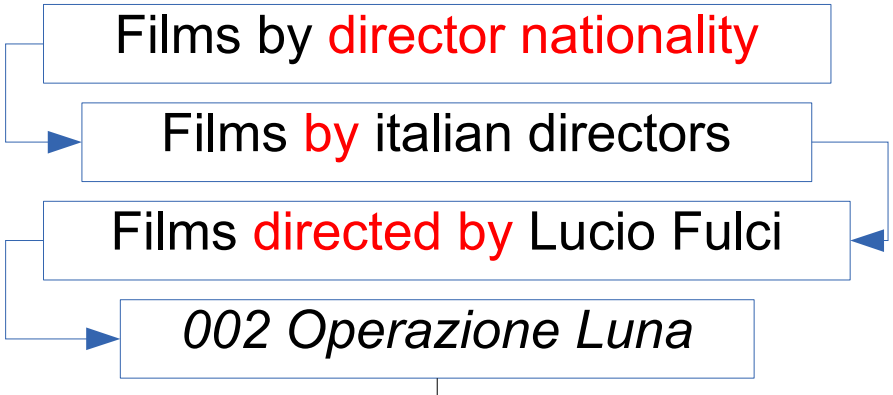
3. Knowledge Extraction Example II: Limitations



"002_Operazione_Luna", *IS_A,* Director

"002_Operazione_Luna", *IS_A,* Cinema

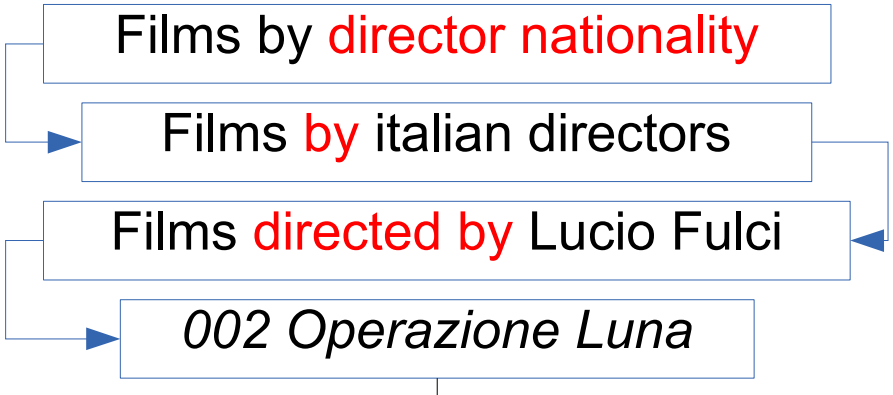
3. Knowledge Extraction Example II: Limitations



"002_Operazione_Luna", *IS_A,* Director

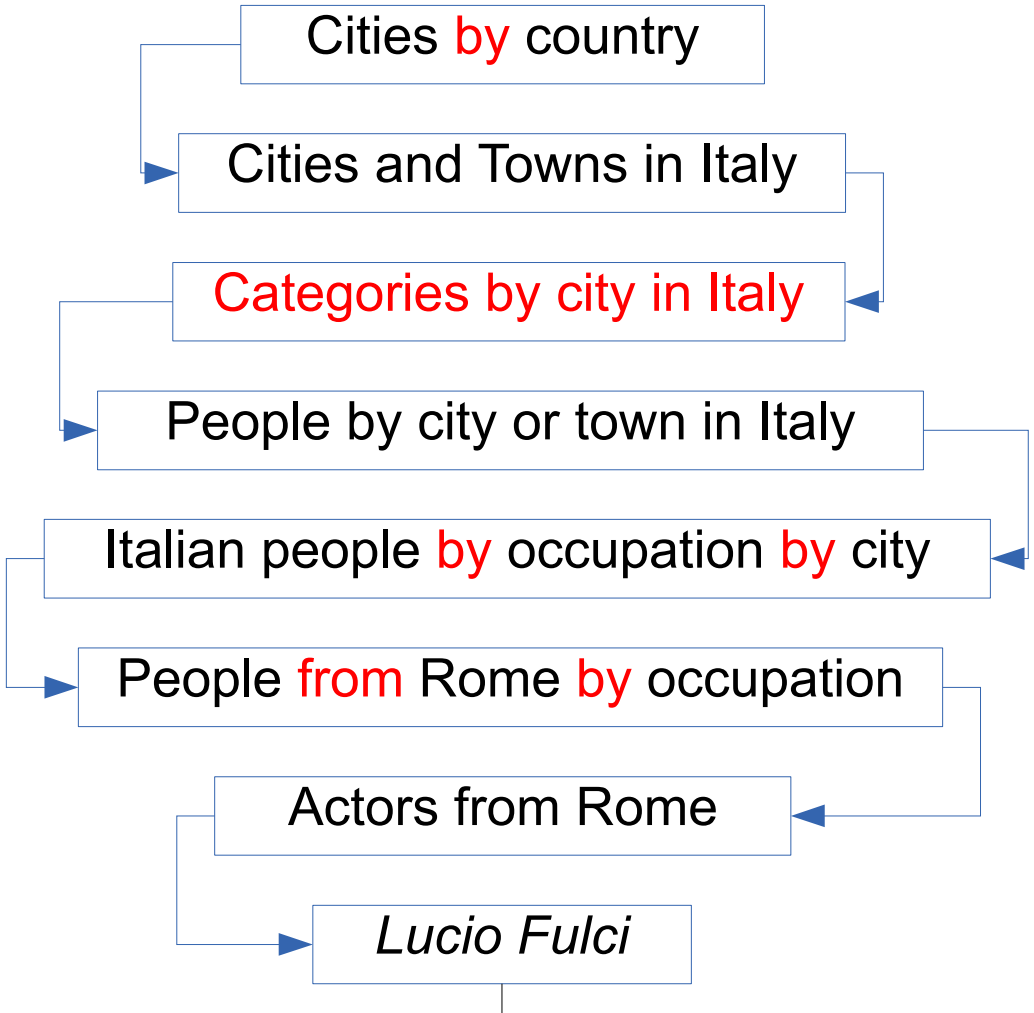
"002_Operazione_Luna", *IS_A,* Cinema

3. Knowledge Extraction Example II: Limitations



"002_Operazione_Luna", IS_A, Director

"002_Operazione_Luna", IS_A, Cinema



"Lucio_Fulci", IS_PART_OF, Cities

Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

“Films directed by Steven Spielberg”

Films → [directed by] → *Steven Spielberg*

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

2) Keep only the most specific links

remove from *“Footwear”* pages and categories shared with *“Shoes”*

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

2) Keep only the most specific links

3) Human-made links are good (Unless otherwise proven)

Use a pruning strategy instead of a build-from-scratch one

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

2) Keep only the most specific links

3) Human-made links are good (Unless otherwise proven)

4) Human-made chains of links are (usually) bad

   Trust connectivity only for a few levels

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

2) Keep only the most specific links

3) Human-made links are good (Unless otherwise proven)

4) Human-made chains of links are (usually) bad

5) Don't impose strict rules automatically

   such as *“X_of_the_Y”* : band names using that pattern do exist!

   Cases must be distinguished with mixed methods

4. Implementation I: Key concepts

1) Atomic entities and meaningful relations

2) Keep only the most specific links

3) Human-made links are good (Unless otherwise proven)

4) Human-made chains of links are (usually) bad

5) Don't impose strict rules automatically

6) KISS (Keep It Simple Stupid)

4. Implementation II: Methods

- NLP methods
  - Give more importance to word matching
  - Part-Of-Speech patterns are used mainly as trigger for further controls

4. Implementation II: Methods

- NLP methods
  - Give more importance to word matching
  - Part-Of-Speech patterns are used mainly as trigger for further controls
- Connectivity methods
  - Short range
  - Mainly used as constraints for other types of method

4. Implementation II: Methods

- NLP methods
  - Give more importance to word matching
  - Part-Of-Speech patterns are used mainly as trigger for further controls
- Connectivity methods
  - Short range
  - Mainly used as constraints for other types of method
- Statistical methods
  - Aim to reconstruct a (natural?) hierarchical structure
  - Hypothesis: pyramidal structure composed by several overlapping pyramids
  - Problem: Statistical values change greatly between different topics
  - Better applied separately to sub-trees (such as *“Clothing”* or *“Music”*)

4. Implementation III: Process

- Step 1: cleanup

a) Wikipedia pages are organized in **Namespaces** (*files, templates...)*

Remove pages with Namespaces different from *articles* or *categories*

b) Administration categories are directly linked with content one

Identify and remove Administration categories: by *connectivity*...

(linked to *“Wikipedia Administration”*)

...and by *Natural Language Processing* (names)

(wikipedia, wikiprojects, lists, mediawiki)

c) Stubs are managed with less care than full articles

Remove all: they generate more noise than content

d) Eventually remove categories left **empty** by the previous steps

Repeated even during the rest of the process

4. Implementation III: Process

- Step 1: cleanup

- Step 2: Chose a sufficiently homogeneous sub-tree
  - Strongly different topics have necessarily different statistics

    Even close categories like *“Baseball”* and *“Fencing”*

4. Implementation III: Process

- Step 1: cleanup
- Step 2: Chose a sufficiently homogeneous sub-tree
- Step 3: Apply combined methods of the three kinds in a breadth-first fashion
  - Most significant statistics are between categories at the same level

4. Implementation III: Process

- Step 1: cleanup
- Step 2: Chose a sufficiently homogeneous sub-tree
- Step 3: Apply combined methods of the three kinds in a breadth-first fashion
- Step 4: Modification check after visiting each level
  - If the tree has been significantly modified: restart from the tree root

    Removal of empty categories doesn't affect statistical values too much

    Almost any other change does
  - Else: proceed to the next level, resume from step three

4. Implementation III: Process

- Step 1: cleanup
- Step 2: Chose a sufficiently homogeneous sub-tree
- Step 3: Apply combined methods of the three kinds in a breadth-first fashion
- Step 4: Modification check after visiting each level
- Repeat until the last level of the tree is reached

  The modification check on the last level has proven unnecessary

4. Implementation IV: the Prototype(s)

- Two different networks for two different purposes:

1) Light and fast network for “on-line” context identification

Unlabeled links, implied meaning: “has something to do with”

“*Films directed by Steven Spielberg*”

“Films produced by Steven Spielberg”

Films → [have something to do with] → *Steven Spielberg*

2) Complete semantic network for more complex tasks

Labeled links

Correctness is much more critical and hard to achieve

4. Implementation V: Unrelated Branches Pruning

- Measures used

  Total number of sub-nodes (both categories and pages)

  Eccentricity: distance from the farthest leaf

  Tangledness: number of sub-nodes shared with brother classes

- Word matching helps identifying where the unrelated branch stems

4. Implementation V: Unrelated Branches Pruning

- Measures used

  Total number of sub-nodes (both categories and pages)

  Eccentricity: distance from the farthest leaf

  Tangledness: number of sub-nodes shared with brother classes

- Word matching helps identifying where the unrelated branch stems
- Example: *“Military uniforms”*

  Total sub-nodes = 11803; Level average ≈ 550

  Eccentricity = 11 ; Level average = 4

  Tangledness = 1.3% ; Level average > 70%

4. Implementation V: Unrelated Branches Pruning

- Measures used

  Total number of sub-nodes (both categories and pages)

  Eccentricity: distance from the farthest leaf

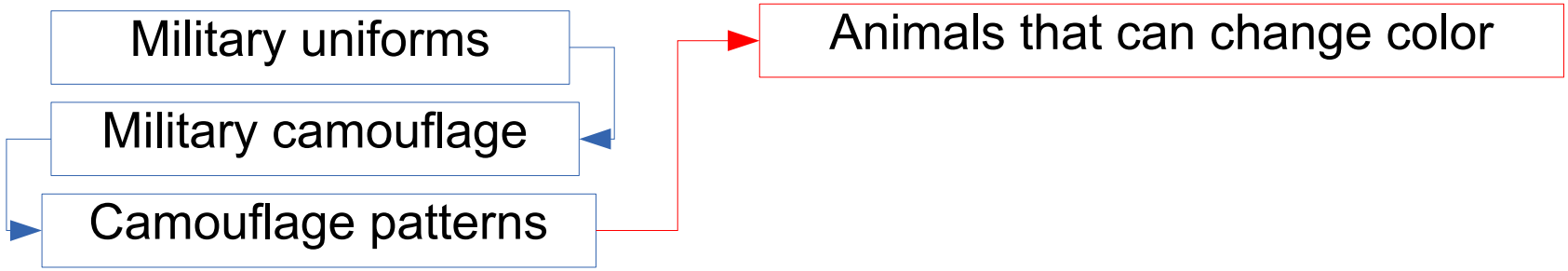
  Tangledness: number of sub-nodes shared with brother classes

- Word matching helps identifying where the unrelated branch stems
- Example: *“Military uniforms”*

  Total sub-nodes = 11803; Level average ≈ 550

  Eccentricity = 11 ; Level average = 4

  Tangledness = 1.3% ; Level average > 70%



4. Implementation V: Unrelated Branches Pruning

- Measures used

  Total number of sub-nodes (both categories and pages)

  Eccentricity: distance from the farthest leaf

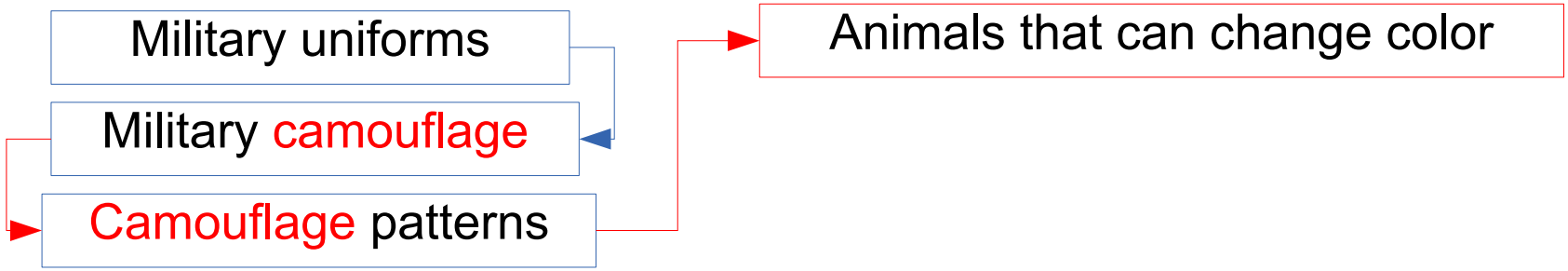
  Tangledness: number of sub-nodes shared with brother classes

- Word matching helps identifying where the unrelated branch stems
- Example: *“Military uniforms”*

  Total sub-nodes = 11803; Level average ≈ 550

  Eccentricity = 11 ; Level average = 4

  Tangledness = 1.3% ; Level average > 70%



4. Implementation VI: Cycle Detection

- Measures used

  Total sub-nodes, eccentricity, tangledness

- Search is leaded by statistics and ended by word matching

4. Implementation VI: Cycle Detection

- Measures used

  Total sub-nodes, eccentricity, tangledness

- Search is leaded by statistics and ended by word matching
- Example: *“Strasbourg”*

  Total sub-nodes = 5820478 ; Level average ≈ 5000

  Eccentricity = 29 ; Level average ≈ 7

  Tangledness = 100%

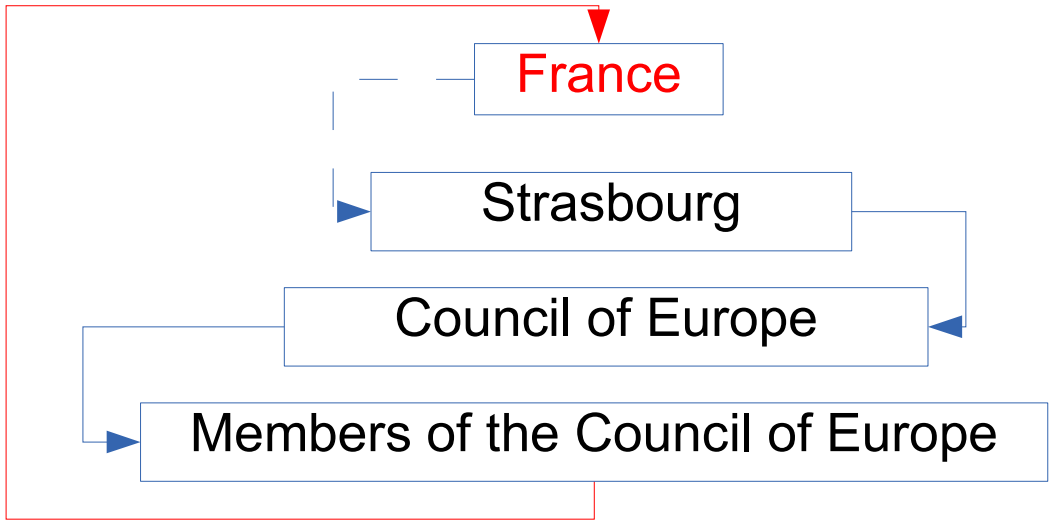
4. Implementation VI: Cycle Detection

- Measures used

  Total sub-nodes, eccentricity, tangledness

- Search is leaded by statistics and ended by word matching
- Example: *“Strasbourg”*

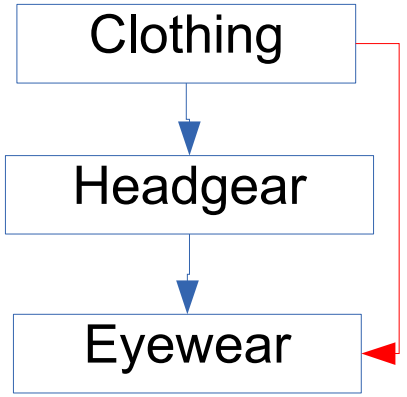
  Total sub-nodes = 5820478 ; Level average ≈ 5000
  Eccentricity = 29 ; Level average ≈ 7
  Tangledness = 100%



4. Implementation VII: Level Adjustment

- Push-down specialized categories
  - Low importance (total number of sub-nodes)
  - Multiple parents

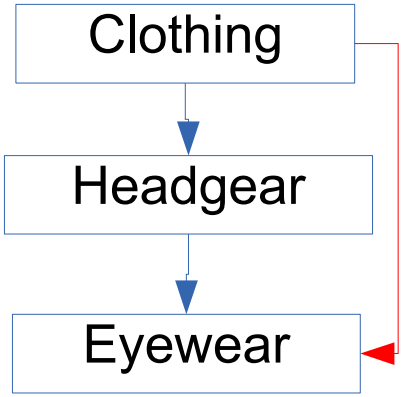
    Leave only the lowest (or lower) level parents



4. Implementation VII: Level Adjustment

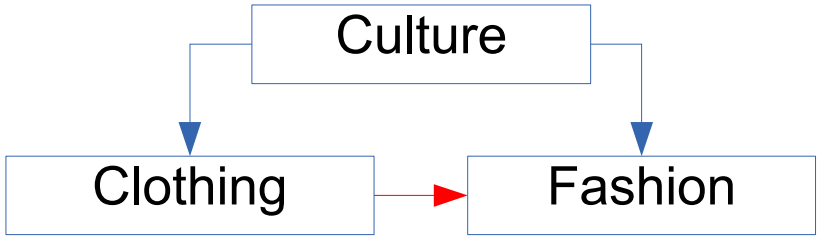
- Push-down specialized categories
  - Low importance (total number of sub-nodes)
  - Multiple parents

    Leave only the lowest (or lower) level parents



- Push-up general categories
  - High importance
  - Shorter path to common parent

    Remove connection with lower-level parent



4. Implementation VIII: Compound Categories Splitting

- Split compound categories through simple word matching
  - Perform a word matching between a parent category and each of its sub-categories

    Names can be lemmatized (cars → car)

    and/or simplified (*"Mini_(marque)"* → Mini; Mini_marque)

4. Implementation VIII: Compound Categories Splitting

- Split compound categories through simple word matching
  - Perform a word matching between a parent category and each of its sub-categories
  - If the sub-category name contains the parent one

    Mark the sub-category as *Compound Category* candidate

    Mark the parent category as C*ompound Root* for that lemma

4. Implementation VIII: Compound Categories Splitting

- Split compound categories through simple word matching
  - Perform a word matching between a parent category and each of its sub-categories
  - If the sub-category name contains the parent one → Mark both
  - Confirm the category as *Compound* when all its components are in one of these conditions:
    i. Have a *Compound Root*
    ii. Are recognized as preposition not part of a named entity
    iii. Match the pattern [VBN IN]

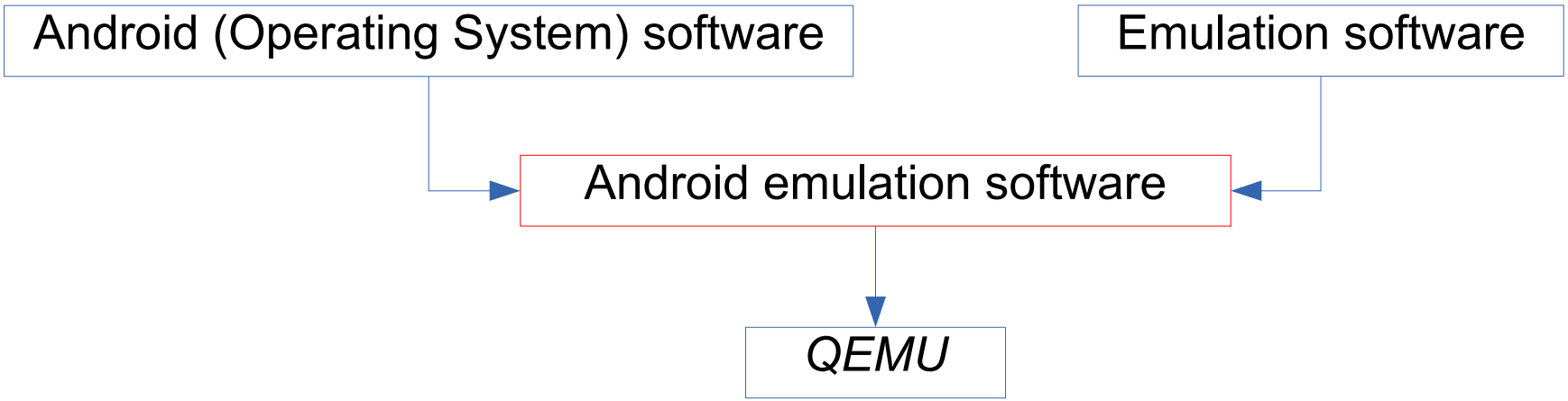
4. Implementation VIII: Compound Categories Splitting

- Split compound categories through simple word matching
  - Perform a word matching between a parent category and each of its sub-categories
  - If the sub-category name contains the parent one → Mark both
  - Confirm the category as *Compound* when all its components are in one of these conditions:
    i. Have a *Compound Root*
    ii. Are recognized as preposition not part of a named entity
    iii. Match the pattern [VBN IN]
  - Split the category by extending all its link to its *Compound Roots*

    Compound Roots may as well be splitted later

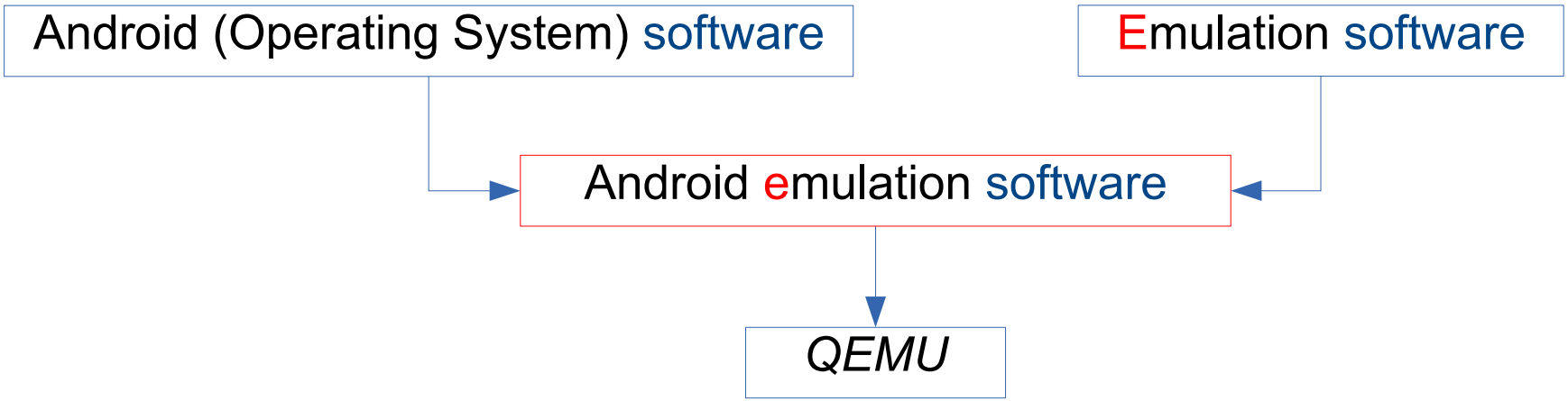
4. Implementation VIII: Compound Categories Splitting

- Split compound categories through simple word matching



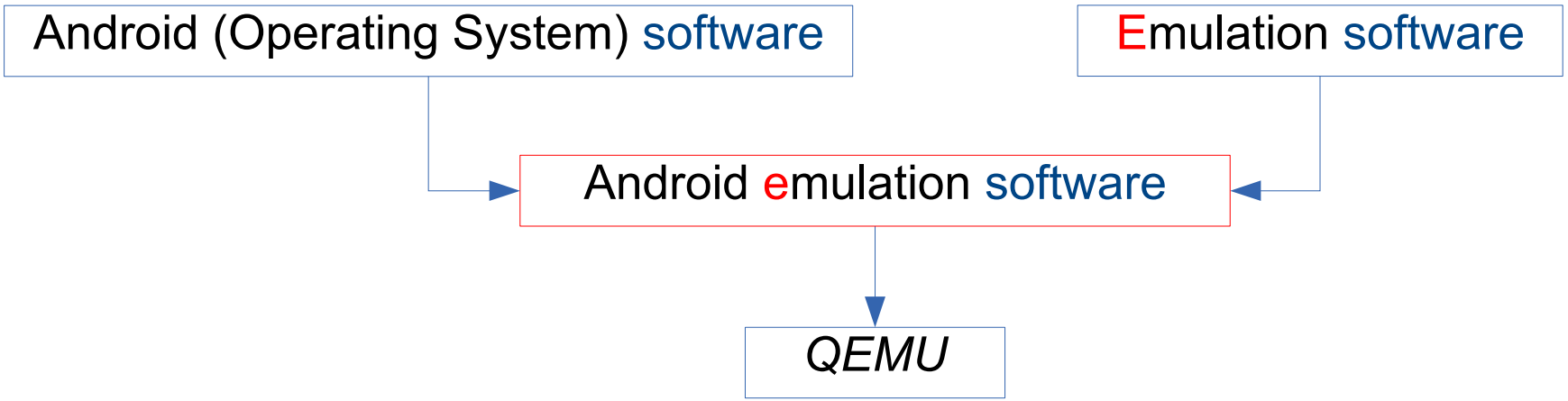
4. Implementation VIII: Compound Categories Splitting

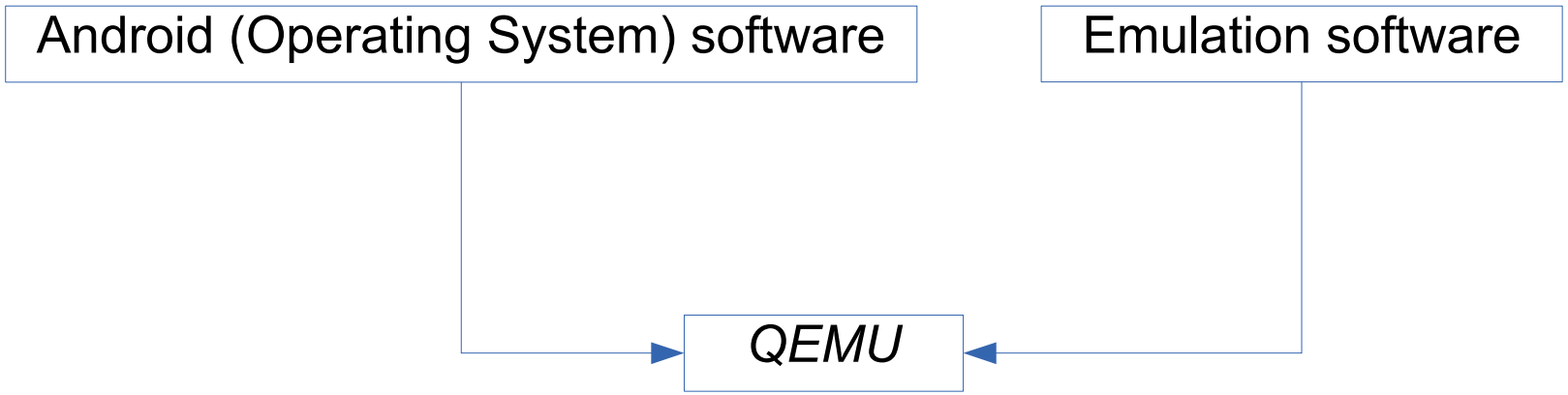
- Split compound categories through simple word matching



4. Implementation VIII: Compound Categories Splitting

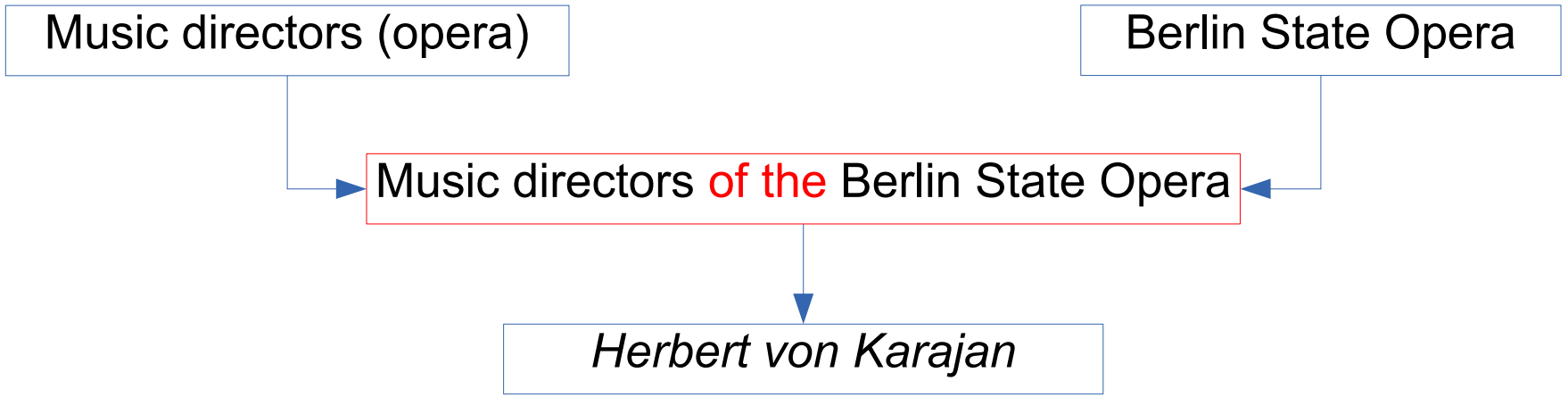
- Split compound categories through simple word matching

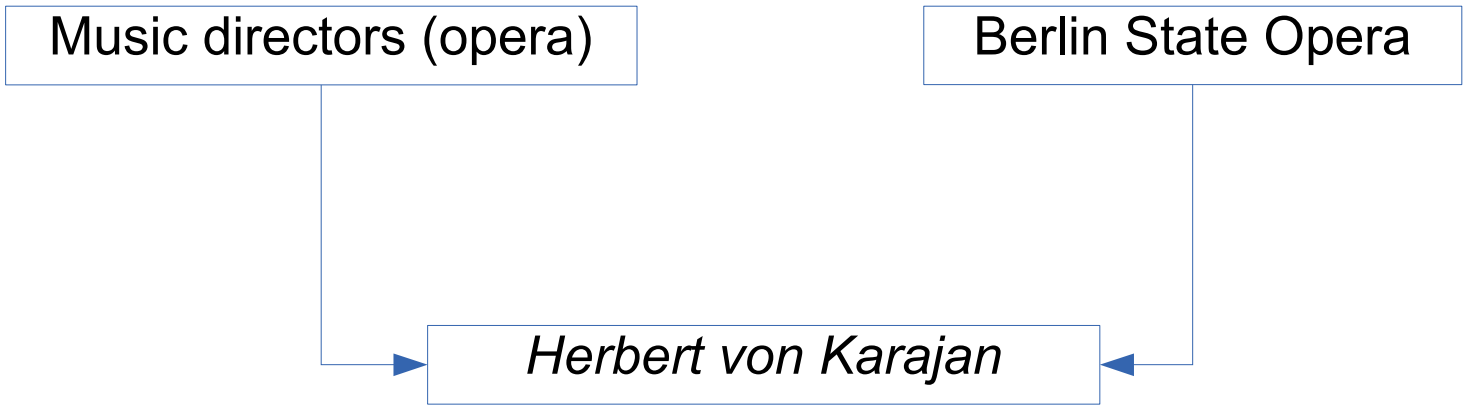




4. Implementation VIII: Compound Categories Splitting

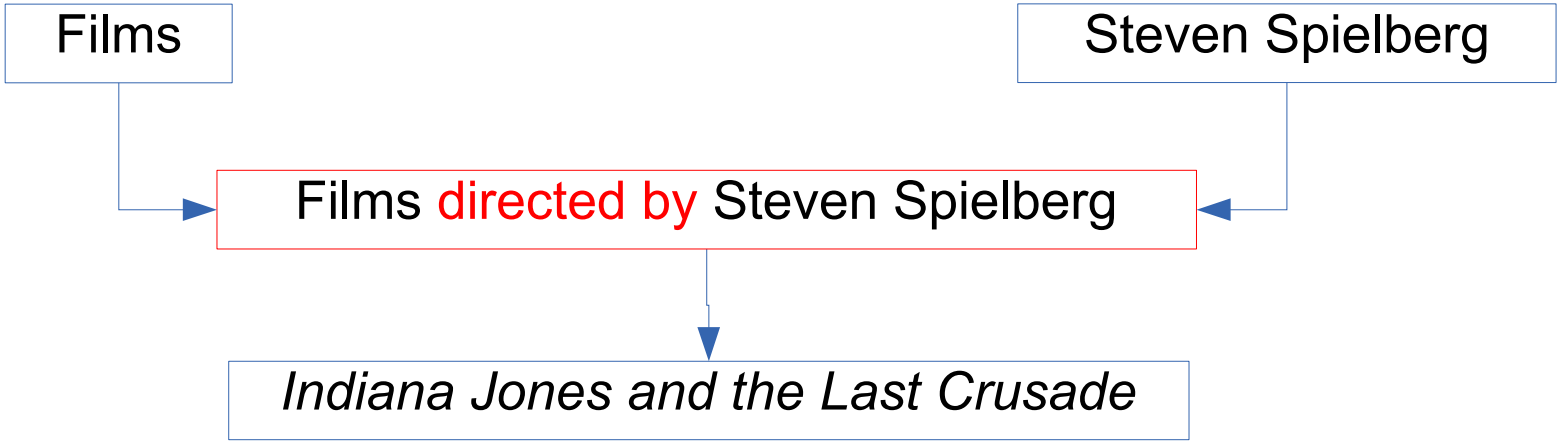
- Split compound categories through simple word matching

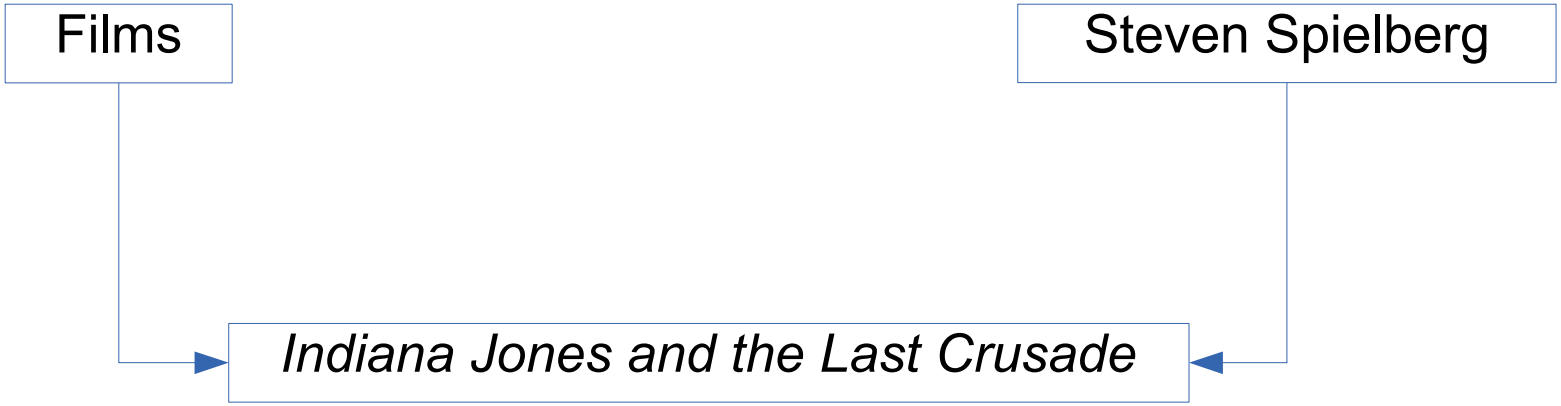




4. Implementation VIII: Compound Categories Splitting

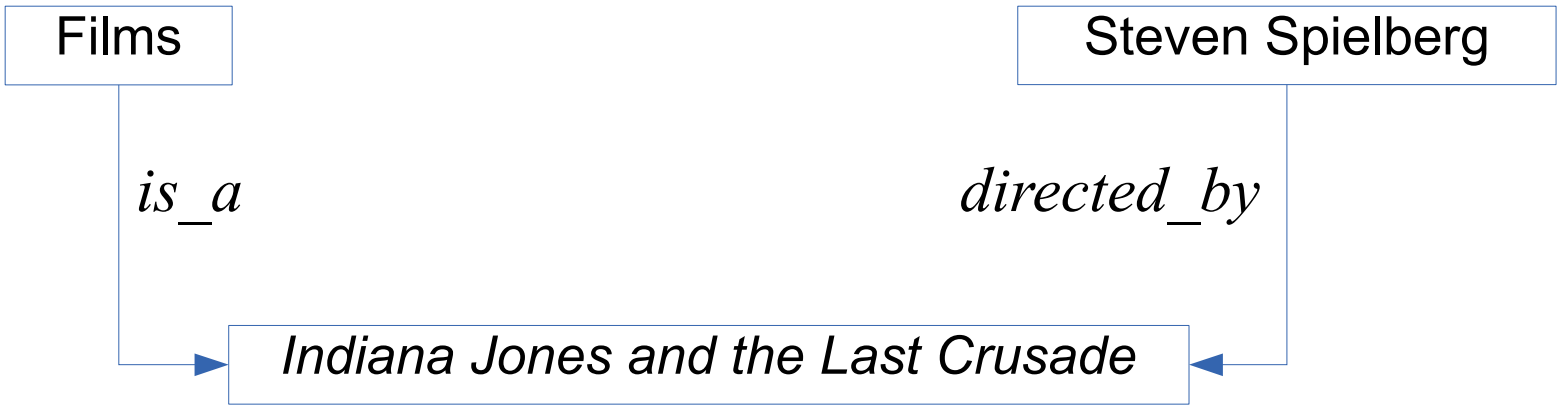
- Split compound categories through simple word matching





4. Implementation VIII: Compound Categories Splitting

- Under study: explicit meaning relations
  - If main entity of the Compound Category title (Noun Phrase head) is plural
    - → Set category
    - → *is_a* relation



4. Implementation VIII: Compound Categories Splitting

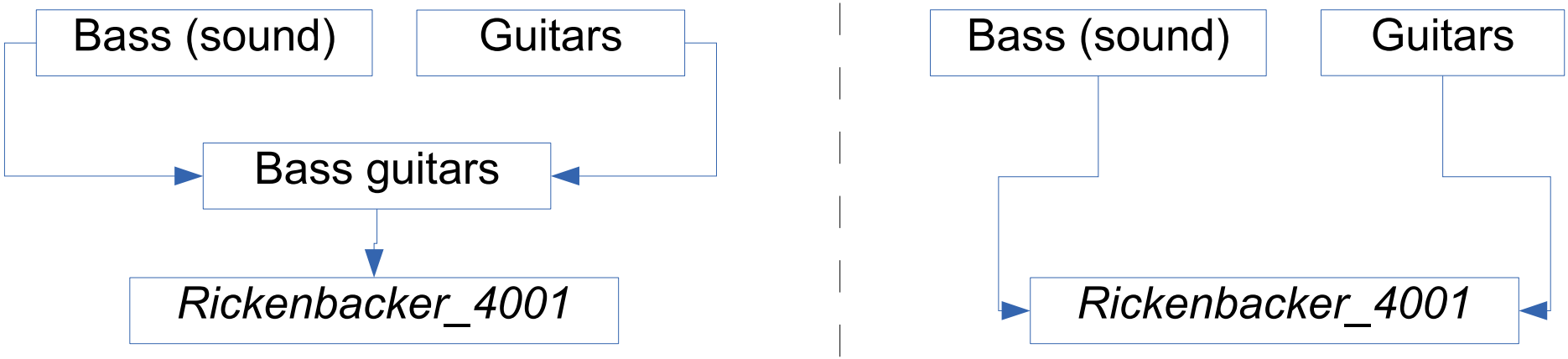
- Split compound categories through simple word matching
  - Increase precision at the cost of recall

    Doesn't lose in generality, but in performances
  - Still not error-proof

4. Implementation VIII: Compound Categories Splitting

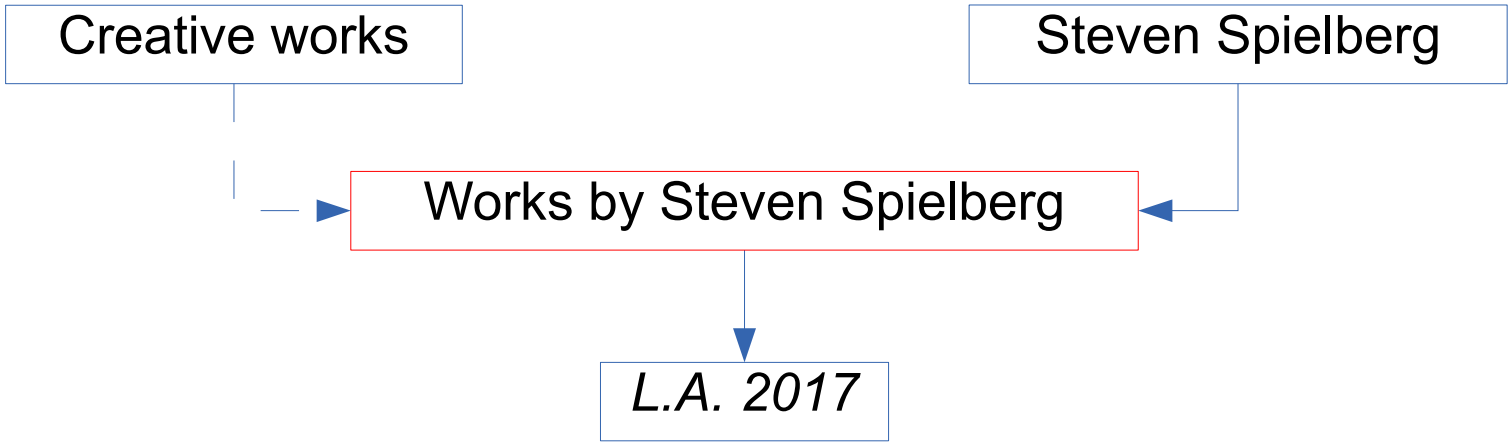
- Split compound categories through simple word matching
  - Increase precision at the cost of recall

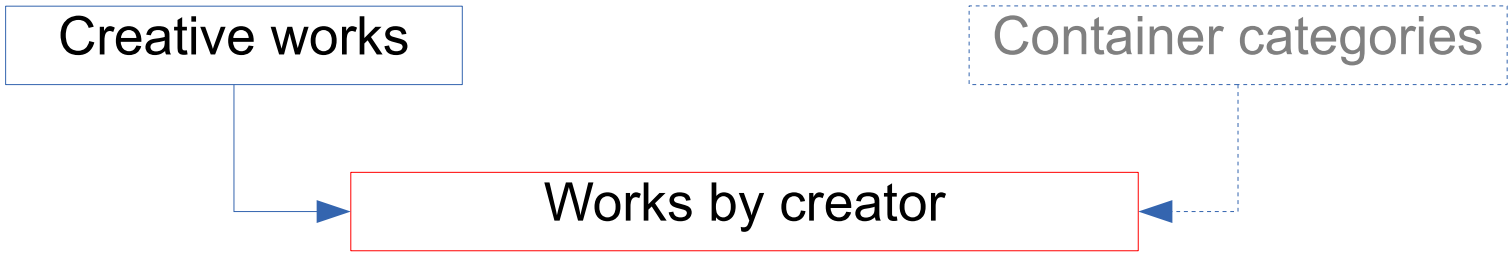
    Doesn't lose in generality, but in performances
  - Still not error-proof (*Under construction:* check on eponym articles)



4. Implementation IX: *X_by_Y* Categories

- *"Made by"* or *Diffusing Category* disambiguation
  - Absence of Compound Root for attribute *Y*
  - *Container Categories* connection





4. Implementation X: Observations and tests

- Implicit relation meanings
  - Still not a “real” Semantic Network
- Limited domain
  - Handmade selection of the sub-trees
  - Automatic identification of “good” sub-trees is under study

4. Implementation X: Observations and tests

- Implicit relation meanings
  - Still not a “real” Semantic Network
- Limited domain
  - Handmade selection of the sub-trees
  - Automatic identification of “good” sub-trees is under study
- Simple disambiguation algorithms have showed good results
  - On *ad hoc* phrases with different complexity

    “With my new Nvidia graphic card, my Dell computer has become legendary!”

    “without michael, the bulls are not the same anymore...”
  - Entities correctly identified
  - Good coarse-grained context identification (based on common parents)

Contents

1. Knowledge Representations & NLP
2. Wikipedia
3. Knowledge Extraction Example
4. Implementation
5. Conclusions

Part 2

5. Conclusions and Future Work

- Statistical methods are effective for network structure “repairs”
  - Increased network usability and efficiency
  - low semantic level

5. Conclusions and Future Work

- Statistical methods are effective for network structure “repairs”
- A mixture of the three method types is necessary
  - NLP and Connectivity-based to obtain semantic relations

    Hard to correctly apply to the Wikipedia Categorization System
  - Several cases cannot be distinguished by single-type methods

5. Conclusions and Future Work

- Statistical methods are effective for network structure “repairs”
- A mixture of the three method types is necessary

Next steps:

- Wikipedia features integration
  - Redirect and disambiguation links to form a Thesaurus
  - Lists to directly infer relations
  - Links between pages to infer relatedness

5. Conclusions and Future Work

- Statistical methods are effective for network structure “repairs”
- A mixture of the three method types is necessary

Next steps:

- Wikipedia features integration
  - Redirect and disambiguation links to form a Thesaurus
  - Lists to directly infer relations
  - Links between pages to infer relatedness
- Suitable disambiguation algorithms
  - Shortest path, preferred entities, co-occurrence probability

5. Conclusions and Future Work

- Statistical methods are effective for network structure “repairs”
- A mixture of the three method types is necessary

Next steps:

- Wikipedia features integration
  - Redirect and disambiguation links to form a Thesaurus
  - Lists to directly infer relations
  - Links between pages to infer relatedness
- Suitable disambiguation algorithms
  - Shortest path, preferred entities, co-occurrence probability
- External sources integration
  - WordNet, hand-crafted information, word occurrence probabilities

Improving the Machine Interpretation of Internet Posts

Q&A

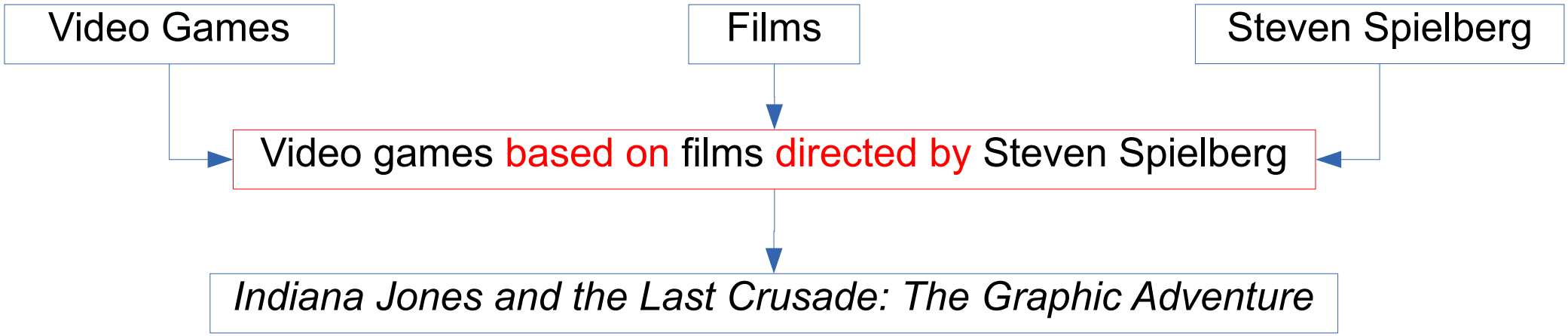
Giacomo Parigi

14.02.14

CVMLab Computer Vision and Multimedia Lab

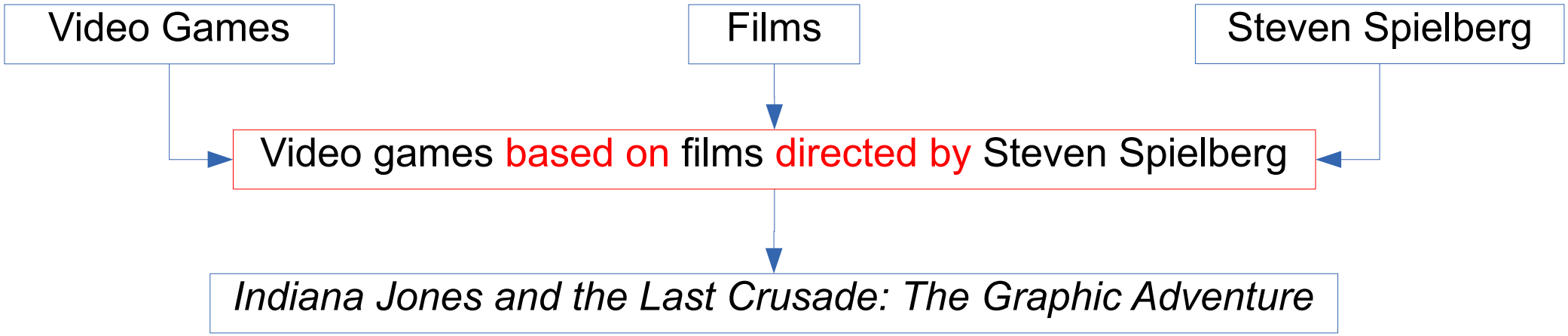
Extra: Lightweight & Semantic Network Difference

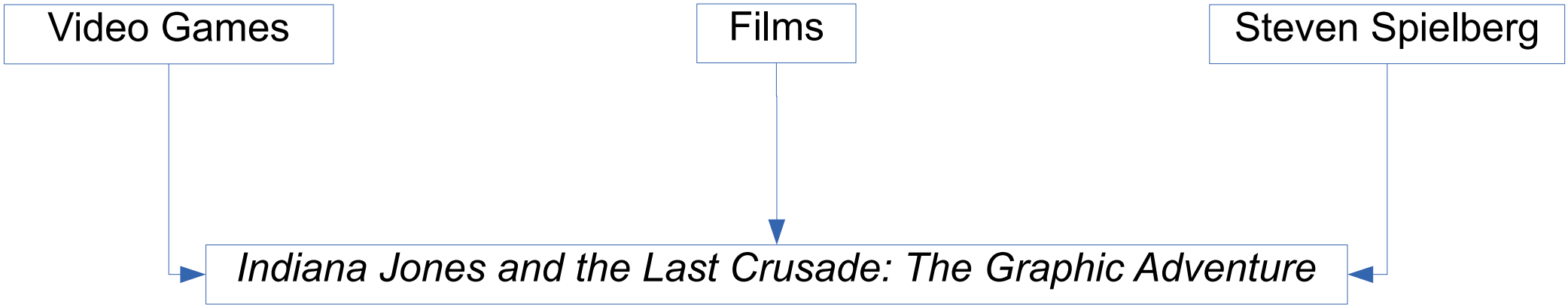
- What about more complex titles?



Extra: Lightweight & Semantic Network Difference

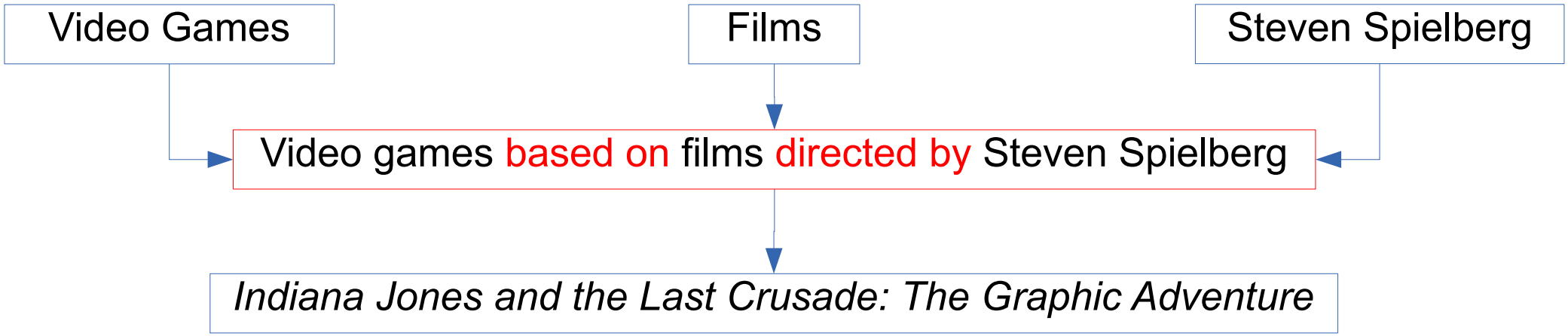
- Lightweight network

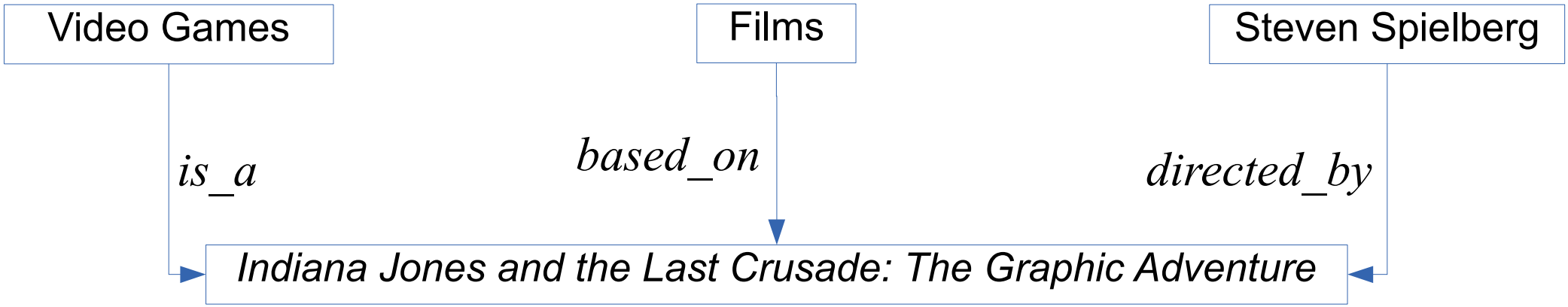




Extra: Lightweight & Semantic Network Difference

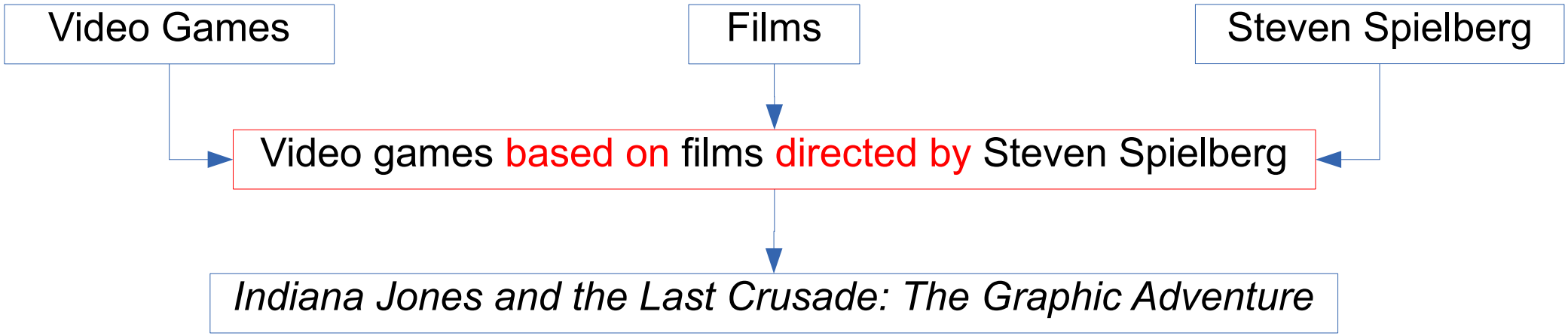
- Complete semantic network – Current methods

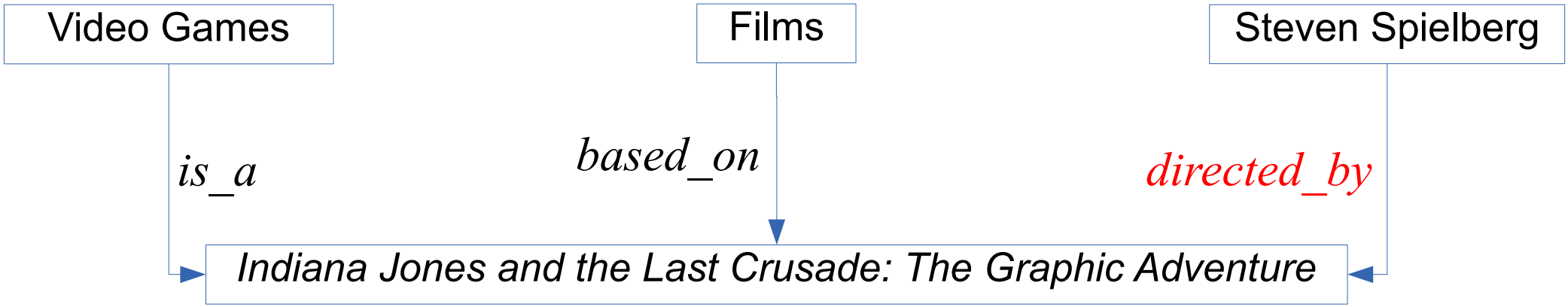




Extra: Lightweight & Semantic Network Difference

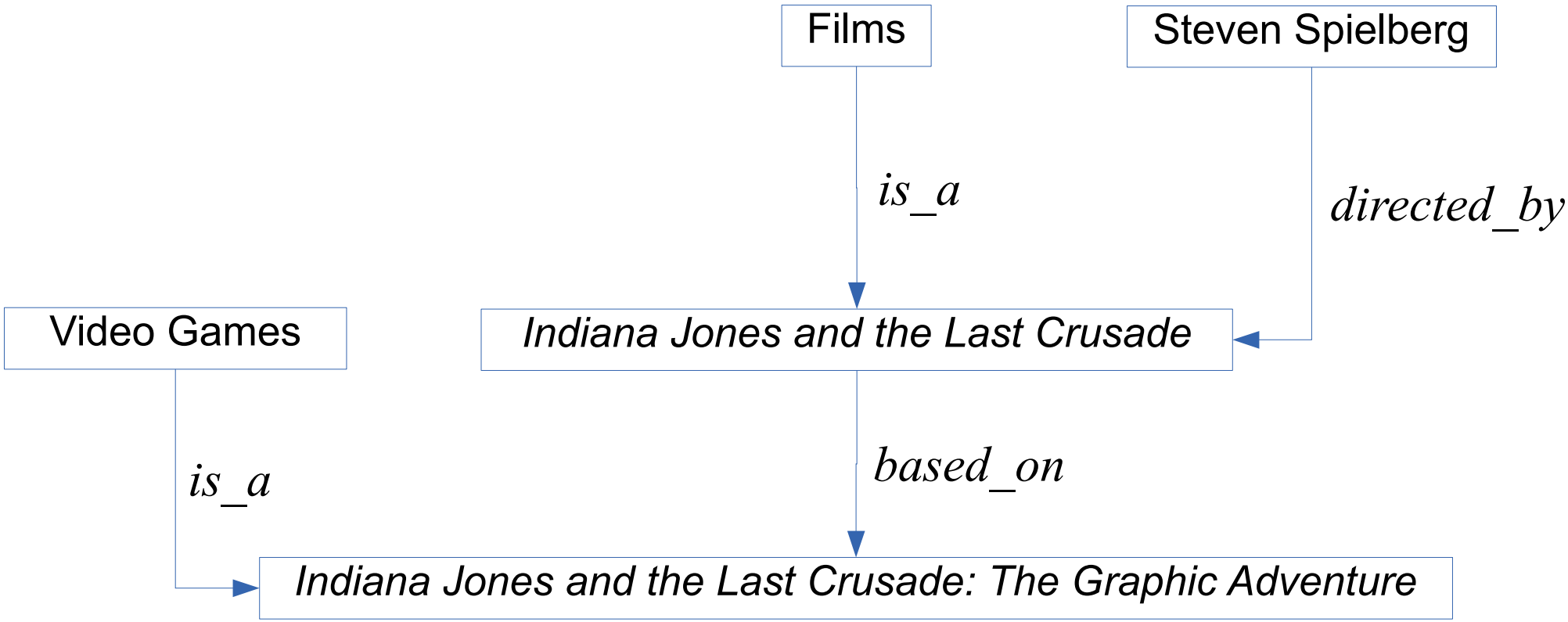
- Complete semantic network – Current methods





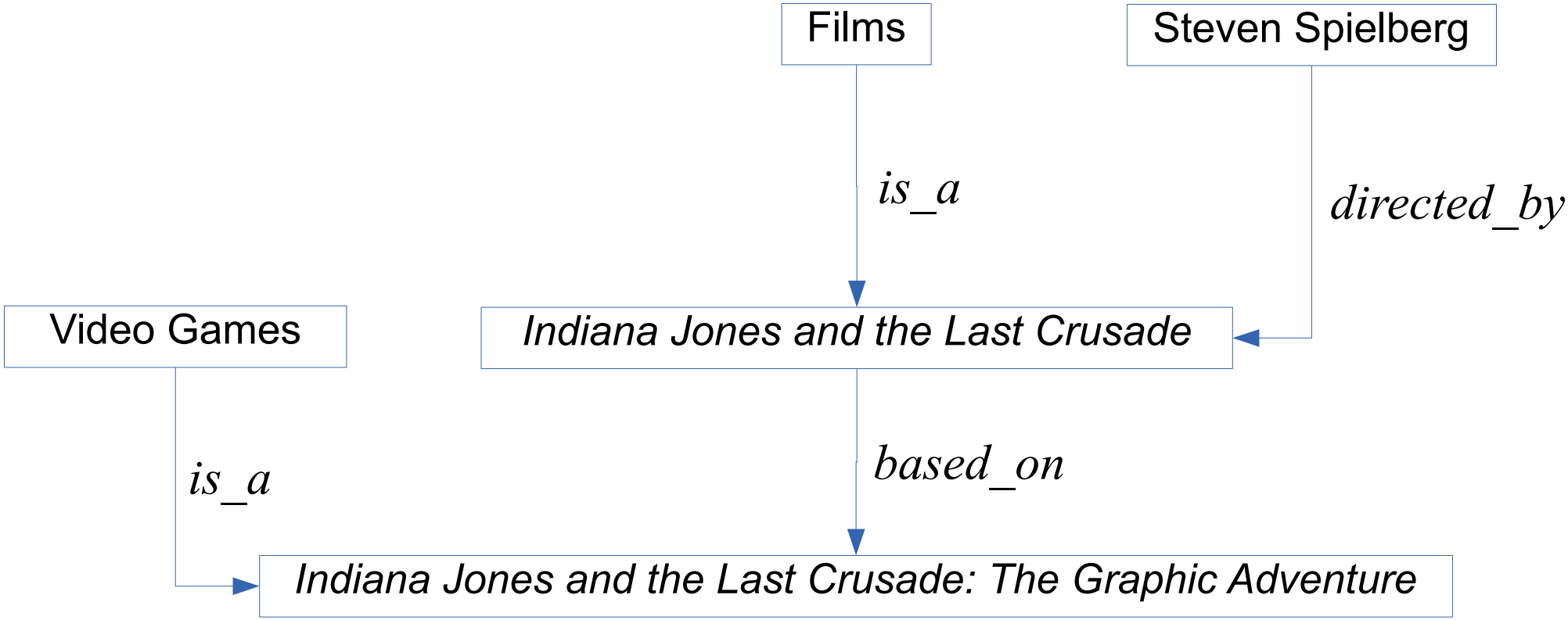
Extra: Lightweight & Semantic Network Difference

- Complete semantic network – Possible “correct” representation



Extra: Lightweight & Semantic Network Difference

- Complete semantic network – Possible “correct” representation
  - How to realize it?
    - Title splitting based on grammatical hierarchy
    - Medium-long range connectivity based methods



Improving the Machine Interpretation of Internet Posts

Thank you

Giacomo Parigi

14.02.14

Computer Vision and Multimedia Lab