Ontology Extension for Reading the Web

Mohamed Thahir

Outline

- Traditional and Open Relation Extraction
- Read the Web Relation Extraction
- Experimental Results
- Coupled learning of Predicates
- Challenges and ongoing work

Traditional Relation Extraction

- A relation is instantiated with a set of manually provided positive and negative examples
- city “capital of” Country

  Positive Seeds:

  {“washington d.c , USA”;”New Delhi , India”..}

  Negative Seeds:

  {“USA , Canada”;”London , India”….}

Open Relation Extraction

- Proposed by Banko et.al 2007
- A classifier is built which given the entities and their context, identifies if there a valid relation
- Performs "Unlexicalized" extraction
- E1 Context E2

  Some Features:
  - Part of Speech (POS) tags in 'Context'
  - Number of tokens and stop words in 'Context'
  - POS tag to left of E1 and to right of E2

Comparison

- Banko et.al 2008 – “TradeOff between Open and Traditional RE”
- Comparison between Traditional (R1-CRF) and Open RE (O-CRF)

Averaged results for 4 common relations

O-CRF (P)	O-CRF (R)	R1-CRF (P)	R1-CRF (R)	Train Ex
75.0	18.4	73.9	58.4	5930

Open RE vs. Traditional RE

Pros:

- ✓ Open RE can scale to the size of the web (hundreds of thousands of relation predicates)
- ✓ Does not require human input unlike traditional RE
- ✓ Pretty reasonable level of precision

Open RE vs. Traditional RE

Cons:

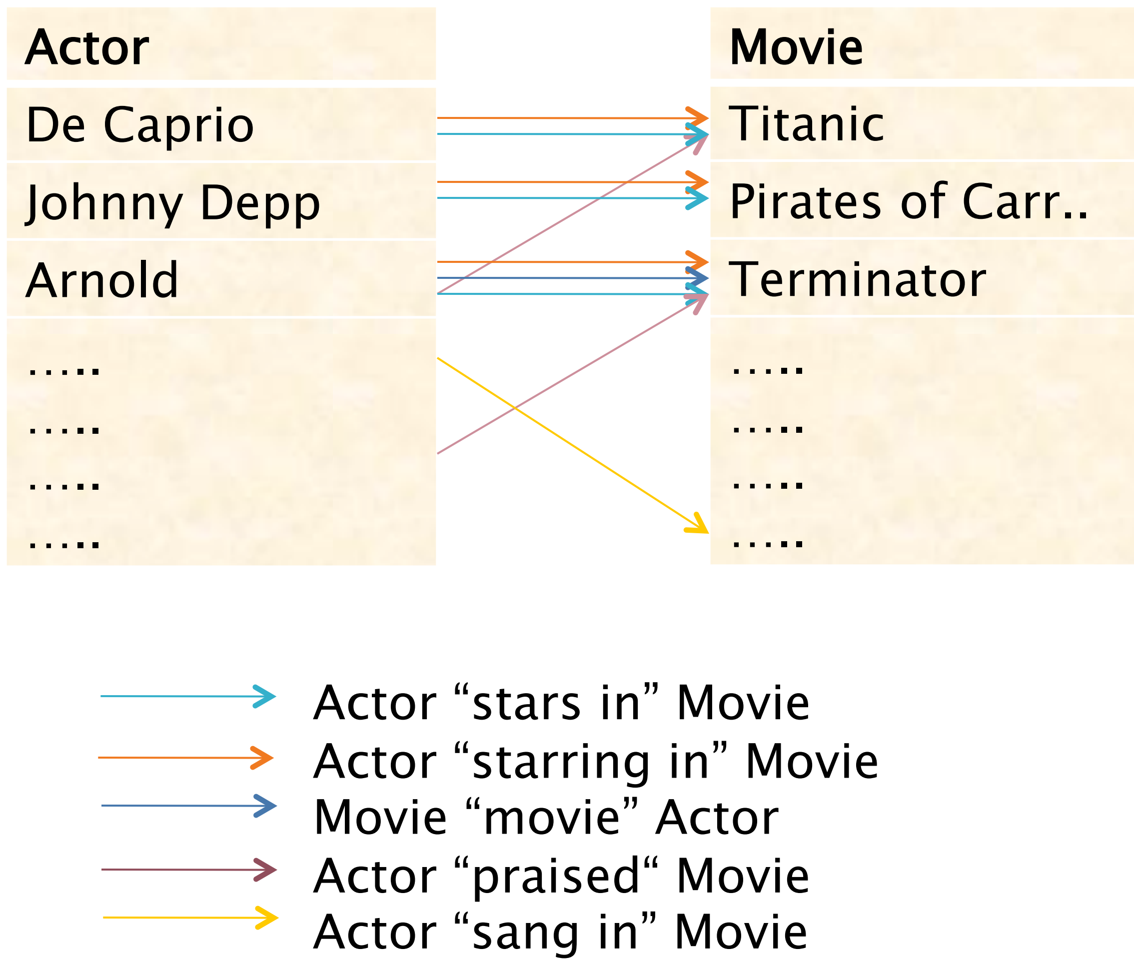
- Open RE has much lower recall
- 30% of extracted tuples are not *well-formed* (does not imply a relation)
  - (demands, securing of, border)
  - (29, dropped , instruments)
- 87% of *well-formed* tuples are abstract/ underspecified
  - (Einstein, derived, theory) – abstract tuple
  - (Washington dc, capital of, USA) – concrete tuple

RTW Relation Extraction

Combine beneficial aspects of Traditional and Open Relation Extraction with RTW

- Find new Relation Predicates automatically
- Also extract positive seed examples and negative seed examples automatically
- Leverage the constrained & coupled learning offered by RTW
- Improve learning of the existing category and relation predicates as well

Learning new Relations



Learning new Relations

- Patterns which are rare are removed
- Patterns which have either a very small Domain or very small Range are removed
  - Removes many irrelevant patterns (caused due to ambiguity)

    NP "was engulfed in" flames

    Vehicle ***Sportsteam***
  - Removes very specific patterns

Learning new Relations

	starring	stars in	movie	sang in	praised
DeCaprio:Titanic	10	22	15	0	2
Depp:Pirates of..	22	10	19	0	0
Arnold:Terminat.	12	15	20	0	1
Arnold:Titanic	0	0	0	0	6
X:Y	0	0	0	7	3
XX:YY	3	5	2	0	0

Learning new Relations

	starring	stars in	movie	sang in	praised
DeCaprio:Titanic	10	22	15	0	2
Depp:Pirates of..	22	10	19	0	0
Arnold:Terminat..	12	15	20	0	1
Arnold:Titanic	0	0	0	0	6
X:Y	0	0	0	7	3
XX:YY	3	5	2	0	0

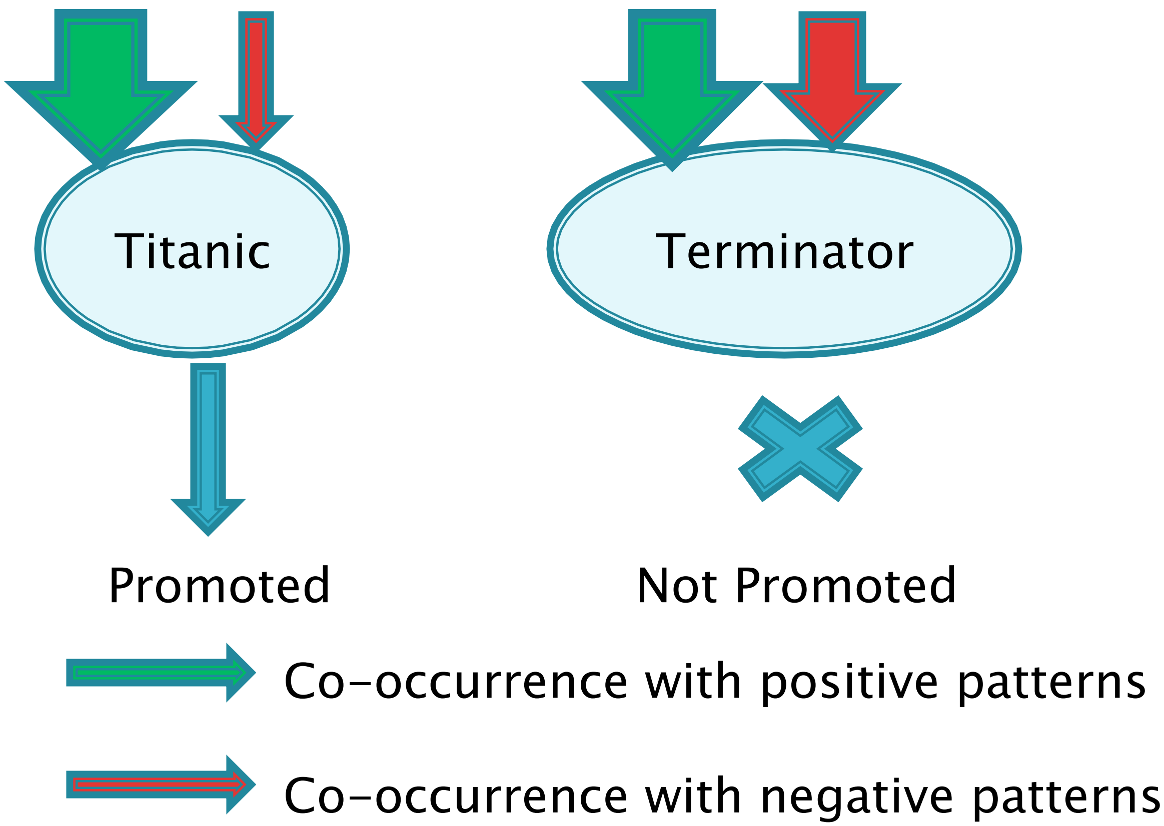
- TF/IDF Normalization
- K-means clustering

Learning new Relations

- Each cluster with sufficient instances is taken as a new relation predicate (*NR*)
- Instances near the centroid of the cluster are taken as seed instances
- Relations whose domain and range are mutually exclusive to the domain and range of *NR* are considered as mutually exclusive for *NR*
- *NR* is introduced to RTW system as a new predicate

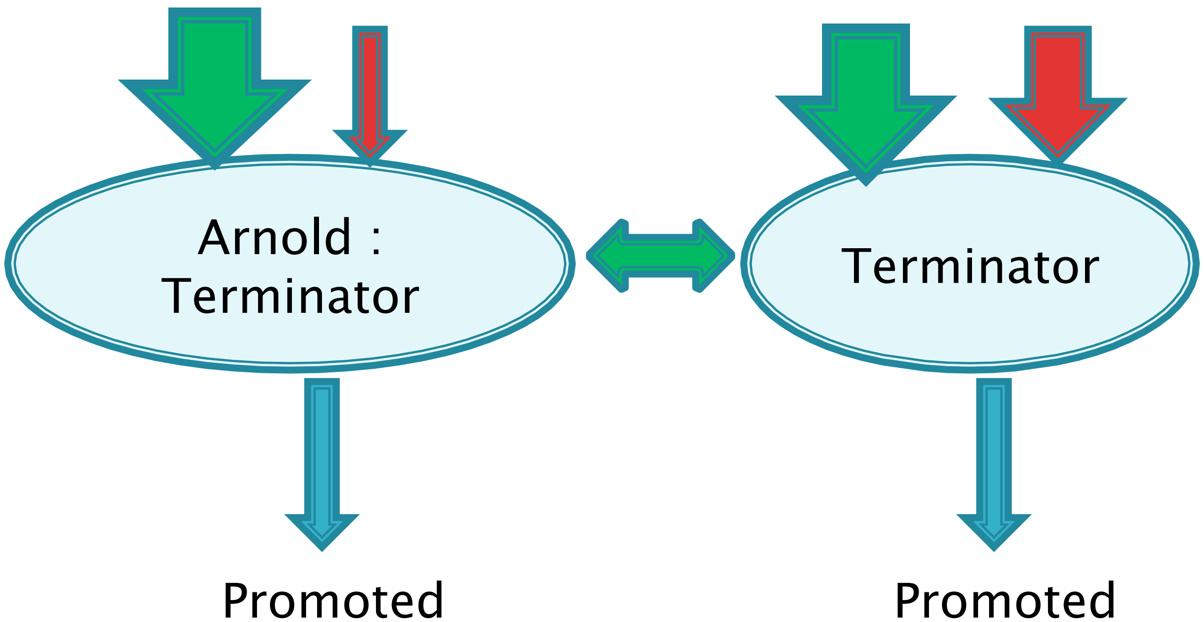
RTW Category Instance Promotion

- *Movie category* predicate classifier



RTW Relation Instance Promotion

- *Actor-Movie relation* predicate classifier



- New Relation helps learning new Category instances

Experimental Results

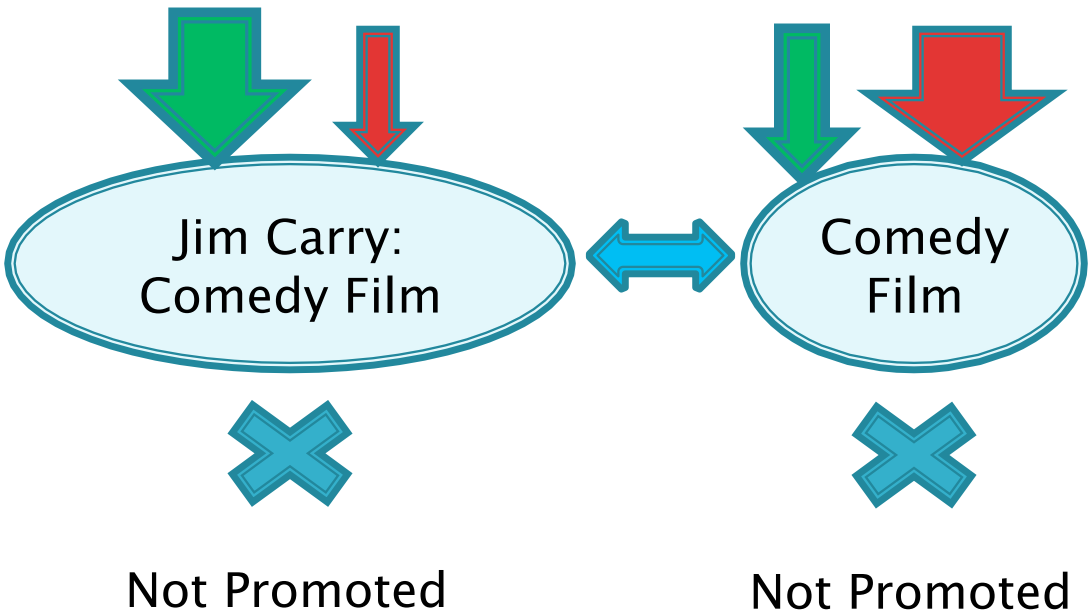
- Improved learning for existing category predicates
- Validation without running the RTW
- Actor : Movie predicate and its high confidence relation pattern set ***R***
- Obtained all instances of "NP1 Context NP2" Where,
  - Context is in ***R***
  - Either NP1 or NP2 is a promoted Actor instance
  - List the other NP that is not the Actor

Experimental Results

- 200+ new movie instances
- Constrained by the number of promoted Actor instances (~800 in CBL)
- Future iterations should cause further increase in Actor and Movie instances.
- $>$ 80% precision
  - Negatives: comedy film
- RTW system category predicate classifiers would ideally not promote these negatives

RTW Relation Instance Promotion

- *Actor-Movie relation* predicate classifier



- Promoted only when category classifier is reasonably confident about the instance

Experimental Results

Repeated same experiment for *Food-Food* relation predicates

Two relations were extracted

Relation	Patterns	Instances	Precision
Contains	"contain", "is rich in", "are rich in"	>700	~60%
typeOf	"Such as", "and other", "including"	>3000	~70%

Negatives: *apple* "contains" *few calories*

Learning more Relation Instances

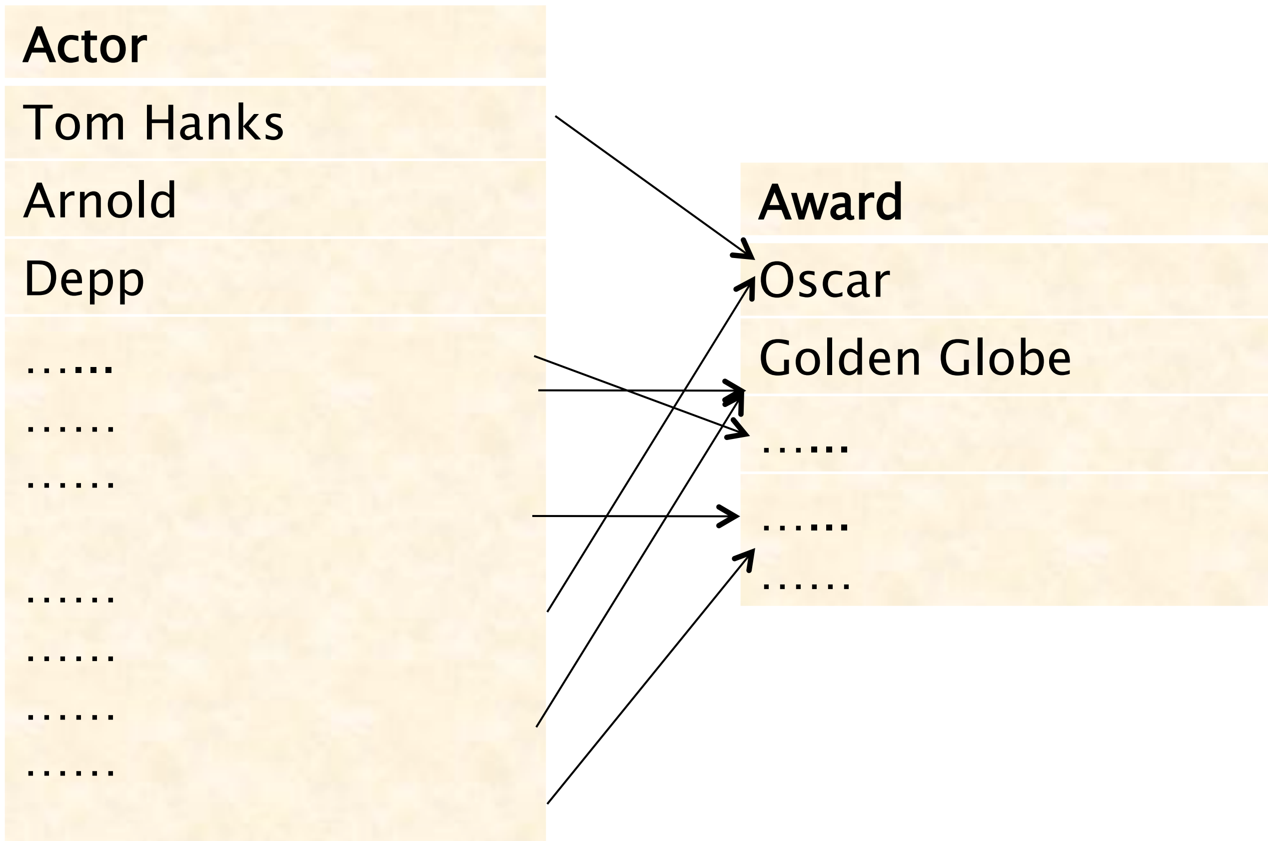
- Learning of Horn Clause rules
- *foodTreatsDisease(food,disease)* – existing predicate
- *isTypeOf(food1,food2)* – learnt predicate
- *isTypeOf(food1,food2)* & *foodTreatsDisease(food2,disease)* ⇨ *foodTreatsDisease(food1,disease)*
- Relation instances could be learnt even without direct contextual patterns connecting them (not possible in Open RE)

Coupled Learning of Predicates

- We saw that new relation predicates leads to learning more category & relation instances
- Learning more category & relation instances would also lead to learning new predicates

Actor	Award
Tom Hanks	Oscar
Arnold	Golden Globe
Depp	……
……	……

Coupled Learning of Predicates



Challenges & Ongoing work

- Many invalid relations are retrieved
- Un-lexicalized approaches to tackle them
- Banko & Etzioni 2008, suggest that 95% of relation patterns are classified into 8 categories

Rel. Frequency	Category	Pattern
37.8	E1 Verb E2	X established Y
22.8	E1 Noun+Prep E2	X settlement with Y
16.0	E1 Verb+Prep E2	X moved to Y
9.4	E1 Infinitive E2	X plans to acquire Y
5.2	E1 Modifier E2	X is Y winner

Challenges & Ongoing work

- Build a model which would estimate the validity of an extracted relation predicate
- Possible Features
  - Un-lexicalized features
  - One-One relations are mostly valid
  - Relations with Hearst's patterns (isA /part of relation – "such as") have high chance of being valid. (Hearst 1992)

Challenges & Ongoing work

<u>Invalid Relations and causes</u>

- Error in the promoted instances
  - CBL promotes Months of the year as countries
  - **Organization** ***'meeting in'*** **Country**

    US Senate ***'meeting in'*** **November**
  - Cluster all **country** instances using the category patterns. Months might form a unique sub cluster.
  - If the **Organization** instances link only to a particular sub-cluster then it indicates a weak relation
  - Above metric could be used as another feature

Challenges & Ongoing work

<u>Invalid Relations and causes</u>

- Ambiguity
  - Animal names match with sports team names
  - **Animal *'won'* trophy**
  - Compare with other predicates which are mutex to it (**Sportsteam *won* Trophy**) and check if there have exactly matching patterns.
  - If the 'animal' instances associated with the *animal 'won' trophy* relation also have evidence that it is a 'Sportsteam' then this is a feature indicating the weakness of **Animal *'won'* trophy** relation

Challenges & Ongoing work

<u>Invalid Relations and causes</u>

- Underspecified Relations
  - These relations require more entities to be useful
  - **SportsTeam *'defeated* ' SportsTeam**
  - X defeated Y, Y defeated X etc.
  - There should be temporal and location information for this relation to make sense