Relation extraction

Bill MacCartney
CS224u
Stanford University

Problem formulation

Overview

- ~~The task of relation extraction~~
- ~~Data resources~~
- Problem formulation
- Evaluation
- Simple baselines
- Directions to explore

Problem formulation

- Inputs and outputs
- Joining the corpus and the KB
- Negative instances
- Multi-label classification

Inputs and outputs

What is the input to the prediction?

- A pair of entity mentions in the context of a sentence?
- A pair of entities, independent of any specific context?

What is the output to the prediction?

- A single relation (multi-class classification)?
- Or multiple relations (multi-label classification)?

Problem formulation

Joining the corpus and the KB

Classifying a pair of entity mentions in corpus? Get labels from KB.

Elon Musk, co-founder of PayPal, went on to establish SpaceX, …

✓ ←

relation	subject	object
founder	SpaceX	Elon_Musk

Classifying a pair of entities for the KB? Get features from corpus.

You may also be thinking of Elon Musk (founder of SpaceX), who …

Elon Musk announced the latest addition to the SpaceX arsenal …

If Space Exploration (SpaceX), founded by Paypal pioneer Elon Musk …

→

1	addition
1	announced
1	by
1	founded
1	founder
1	latest
1	of
1	PayPal
1	pioneer
2	the
1	to

(Elon_Musk, SpaceX)

Problem formulation

Joining the corpus and the KB

```
dataset = rel_ext.Dataset(corpus, kb)
dataset.count_examples()
```

relation	examples	triples	examples /triple
adjoins	58854	1702	34.58
author	11768	2671	4.41
capital	7443	522	14.26
contains	75952	18681	4.07
film_performance	8994	3947	2.28
founders	5846	1960	2.98
genre	1576	824	1.91
has_sibling	8525	2563	3.33
has_spouse	12013	2994	4.01
is_a	5112	2542	2.01
nationality	3403	1598	2.13
parents	3802	1586	2.40
place_of_birth	1657	1097	1.51
place_of_death	1523	831	1.83
profession	1851	1216	1.52
worked_at	3226	1150	2.81

Problem formulation

Negative instances

To train a classifier, we also need negative instances!

So, find corpus examples containing pairs of entities not related in KB

```
unrelated_pairs = dataset.find_unrelated_pairs()
print('Found {0:,} unrelated pairs, including:'.format(len(unrelated_pairs)))
for pair in list(unrelated_pairs)[:10]:
    print('    ', pair)
```

```
Found 247,405 unrelated pairs, including:
    ('Inglourious_Basterds', 'Christoph_Waltz')
    ('NBCUniversal', 'E!')
    ('The_Beatles', 'Keith_Moon')
    ('Patrick_Lussier', 'Nicolas_Cage')
    ('Townes_Van_Zandt', 'Johnny_Cash')
    ('UAE', 'Italy')
    ('Arshile_Gorky', 'Hans_Hofmann')
    ('Sandra_Bullock', 'Jae_Head')
```

Problem formulation

Multi-label classification

Many entity pairs belong to more than one relation:

```
dataset.count_relation_combinations()
```

```
The most common relation combinations are:
      1216 ('is_a', 'profession')
       403 ('capital', 'contains')
       143 ('place_of_birth', 'place_of_death')
        61 ('nationality', 'place_of_birth')
        11 ('adjoins', 'contains')
         9 ('nationality', 'place_of_death')
         7 ('has_sibling', 'has_spouse')
         3 ('nationality', 'place_of_birth', 'place_of_death')
         2 ('parents', 'worked_at')
```

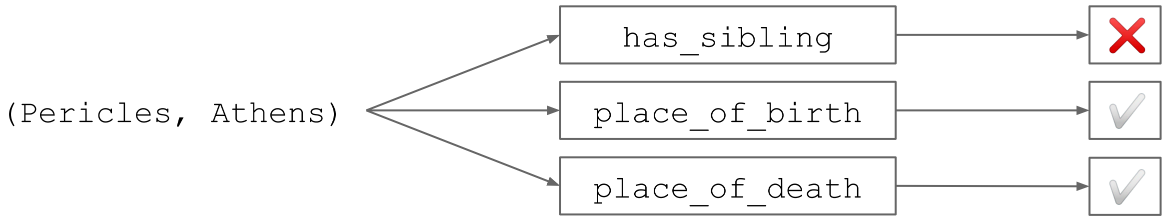
This suggests formulating our problem as *multi-label classification*.

Problem formulation

Multi-label classification: binary relevance

Many possible approaches to multi-label classification.

The most obvious is the *binary relevance method:*
just train a separate binary classifier for each label.



Disadvantage: fails to exploit correlations between labels.

Advantage: simple.

Problem formulation

Binary classification of KB triples

So here's the problem formulation we've arrived at:

Input:	an entity pair and a candidate relation
Output:	does the entity pair belong to the relation?

In other words: binary classification of KB triples!

That is, given a candidate KB triple, do we predict that it is valid?

```
(worked_at, Elon_Musk, SpaceX) ?
```