Distant Supervision for Relation Extraction

Machine Based Question Answering

Presented by Fraser Bowen
Structure of this presentation

1. Introduction to Relation Extraction for Question Answering systems
2. Presentation of the best way to generate relations: distant supervision
3. Formal definition of the problem, implementation steps (Surdeanu, 2012)
4. How to evaluate Relation Extraction
5. Latest development in distant supervision: presentation of (Min et al., 2013)'s paper
6. How to expand implementation to include Min’s improvements
7. Comparison of results
8. Some difficult, thought-provoking questions for discussion
How does Machine Learning based Question Answering work?

An answer to a question is determined from a model trained from data.

Many possible ways to train a model, e.g. End-to-end.

Today we’ll talk about a special type of training data: relations.
What is a relation?

mother-of(X, Y)

Instance of “mother-of” relation: mother-of(“Queen Elizabeth”, “Prince Charles”)

Instance of “calorie-content” relation: calorie-content(“Banana”, “89”)

● The system needs to understand which relation to use in order to answer the question
● It means you’ll need to find a lot of relations somewhere...
What is Relation Extraction?

Extract relations from large corpus

Text based corpus with lots of information, ie. Wikipedia

Fully automatic

Train a classifier to say whether a relation is positive or negative (open information extraction)

“John, not Jane, works at IBM”

Accuracy rate: 70% (Riedel, 2010)
Types of training data for your Q&A machine

Small hand-labeled data  Too expensive!

Fully automatic labelling for lots of data on a text corpus  Not very accurate!

Using a small number of seed instances to do bootstrap learning  Low precision! And semantic drift!

Distant supervision using a knowledge base  Best option! Topic for today

History of relation extraction: (Mintz, 2009)
What does a Knowledge Base look like?

<table>
<thead>
<tr>
<th>Relation name</th>
<th>Size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/people/person/nationality</td>
<td>281,107</td>
<td>John Dugard, South Africa</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>253,223</td>
<td>Belgium, Nijlen</td>
</tr>
<tr>
<td>/people/person/profession</td>
<td>208,888</td>
<td>Dusa McDuff, Mathematician</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>105,799</td>
<td>Edwin Hubble, Marshfield</td>
</tr>
<tr>
<td>/dining/restaurant/cuisine</td>
<td>86,213</td>
<td>MacAyo’s Mexican Kitchen, Mexican</td>
</tr>
<tr>
<td>/business/business_chain/location</td>
<td>66,529</td>
<td>Apple Inc., Apple Inc., South Park, NC</td>
</tr>
<tr>
<td>/biology/organism_classification_rank</td>
<td>42,806</td>
<td>Scorpaeniformes, Order</td>
</tr>
<tr>
<td>/film/film/genre</td>
<td>40,658</td>
<td>Where the Sidewalk Ends, Film noir</td>
</tr>
<tr>
<td>/film/film/language</td>
<td>31,103</td>
<td>Enter the Phoenix, Cantonese</td>
</tr>
<tr>
<td>/biology/organism_higher_classification</td>
<td>30,052</td>
<td>Calooteryx, Calooterygidae</td>
</tr>
<tr>
<td>/film/film/country</td>
<td>27,217</td>
<td>Turtle Diary, United States</td>
</tr>
<tr>
<td>/film/writer/film</td>
<td>23,856</td>
<td>Irving Shulman, Rebel Without a Cause</td>
</tr>
<tr>
<td>/film/director/film</td>
<td>23,539</td>
<td>Michael Mann, Collateral</td>
</tr>
<tr>
<td>/film/producer/film</td>
<td>22,079</td>
<td>Diane Eskenazi, Aladdin</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>18,814</td>
<td>John W. Kern, Asheville</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>18,619</td>
<td>The Octopus Project, Austin</td>
</tr>
<tr>
<td>/people/person/religion</td>
<td>17,582</td>
<td>Joseph Chartrand, Catholicism</td>
</tr>
<tr>
<td>/book/author/works_written</td>
<td>17,278</td>
<td>Paul Auster, Travels in the Scriptorium</td>
</tr>
<tr>
<td>/soccer/football_position/players</td>
<td>17,244</td>
<td>Midfielder, Chen Tao</td>
</tr>
<tr>
<td>/people/deceased_person/cause_of_death</td>
<td>16,709</td>
<td>Richard Daintree, Tuberculosis</td>
</tr>
<tr>
<td>/film/film/music</td>
<td>14,070</td>
<td>Stavisky, Stephen Sondheim</td>
</tr>
<tr>
<td>/business/company/industry</td>
<td>13,805</td>
<td>ATS Medical, Health care</td>
</tr>
</tbody>
</table>
Glossary

Relation: $\text{mother-of}(x, y)$

Instance: $\text{mother-of}(\text{“Queen Elizabeth”}, \text{“Prince Charles”})$

Entity: “Queen Elizabeth”

Entity pair: (“Queen Elizabeth”, “Prince Charles”)

Mention: “Queen Elizabeth is the mother of Prince Charles”

Bag: {“Queen Elizabeth is the mother of Prince Charles”, “Charles’ mum Queen Elizabeth…”, “Of all of Queen Elizabeth’s children, Charles was…”, “Prince Charles is the heir to Queen Elizabeth’s throne”}
Distant Supervision with a Knowledge Base (1)

Knowledge base = hand-labeled data

Sentences containing the entities in the relations described in the knowledge base are found in the corpus

Look at each sentence in your corpus, and extract the entities

For each possible entity pair, collect all mentions of them into bags

“The distant supervision assumption is that if two entities participate in a relation, any sentence that contain those two entities might express that relation” (Mintz, 2009, p.4)
Distant Supervision with a Knowledge Base (2)

For each bag, check to see if they correspond with a relation in the knowledge base.

If enough of the mentions express the relation, label the bag with the relation.

If the mentions seem to express something else, give it an “OTHER” label.

This is done with a trained binary classifier, just like in automatic relation extraction.
Distant Supervision with a Knowledge Base (3)

Result: bag of natural language sentences, and their corresponding correct relation

“John, not Jane, works at IBM”, “John’s job at IBM was…”

employee-of(“John”, “IBM”)

Encodes information into a feature vector: a set of sentences for each relation

employee-of(“John”, “IBM”) = {“John, not Jane, works at IBM”, “John’s job at IBM was well-paid”}
Create a matrix containing all the relations

--- Your answers! Training for your question-answering system
Illustration of Distant Supervision

Knowledge Base → Relation extraction → Bags of positive mentions → Internal representation → Feature vectors

Text corpus → Entity extraction → Bags of mentions → Question Answering neural network

Feature vectors → Training data
Different types of relation extraction

Single-instance binary classification asks whether a bag of mentions represent a relation or not

Multi-instance binary classification allows for some mentions to not represent the relation (at-least-one model)

Multi-instance multi-label classification further assumes that each bag, and each mention can have multiple labels

(Min et al. 2013) and (Surdeanu et al. 2012) use multi-instance multi-label classification (MIML)
MIML Relation Extractor - inputs and outputs

Inputs:

- Set of bags of mentions taken from a text corpus
- List of known relations from a knowledge base
- Extraction model, trained on relations that appear at least once in the corpus

Outputs:

- List of relations
- List of mentions for each relation

(Surdeanu et al., 2012)
Formal definition of the MIML extractor (1)

Every mention “x” is assigned a multi-label classification from the “z” classifier.

“Barack Obama, born and raised in the USA”

This mention is given two labels:

born-in(“Barack Obama”, “USA”)

raised-in(“Barack Obama”, “USA”)

Alternatively, it returns a “NIL” label
Formal definition of the MIML extractor (2)

We calculate the labels for every sentence bag-by-bag

“n” is the entity pair - or the bag of mentions

Now we need to assign the entire bag a label.

There is a binary classifier for each relation - are there enough mentions with this label?

If so, “y” for the respective relation -> positive label
Advantage of this model

It can learn that two relation labels are mutually exclusive:

- mother-of(x, y)
- daughter-of(x, y)

And that other labels often come together:

- location-contains(x, y)
- capital-of(x, y)
Training the model

The model is trained with your set of relations from the Knowledge Base

**Expectation Maximisation** is used to maximise the log-likelihood of the data

**Expectation step:** assign some labels at the mention level “z” and the entity level “y”

**Maximisation step:** update the weight values using logistic regression to maximise the log likelihood, given the current “z” and “y” labels.

Initialised with a classifier that assumes each mention represents the relation perfectly (better than random classification!)
Evaluating the model

Corpus and knowledge base provided by the KBP shared task at the Text Analysis Conference

KBP provided a large gold standard of relation instances which is used to measure precision and recall.
Precision and Recall

**Precision:** Successfully classified bags / total number of classified bags

**Recall:** Successfully classified bags / total number of possible relation instances

KBP scorer was used to calculate the number of true positives, true negatives, false positives and false negatives.

Used several times on many sections on the data to get a spread of results
MIML in comparison to older models

Mintz (single-instance single-label)
Riedel (multi-instance, single-label)
Hoffmann (multi-instance, multi-label)
(Min et al. 2013)

It offers improvements on this otherwise great model.

It focuses on finding the false negatives, which are not taken into account.

Ignoring fewer false negatives will result in a higher recall.
WikiData: Sylvia Plath killed herself

Manner of death(“Sylvia Plath”, “suicide”)

[Image of Sylvia Plath]
But WikiData doesn’t know how Charlie Chaplin died

manner-of-death("Charlie Chaplin", ???)

Missing from the Knowledge Base!

...and nothing on 93.8% of people’s nationalities!
What to do with new information

manner-of-death(“Sylvia Plath”, “suicide”)
manner-of-death(“Charlie Chaplin”, ???)

Corpus contains:

- “Sylvia Plath died of suicide”
- “Charlie Chaplin died of a stroke”

...can we trust it?
Unlabeled mentions are mainly wrong

“The multiple sclerosis patient was always talking about Charlie Chaplin”

Therefore: manner-of-death(“Charlie Chaplin”, “multiple sclerosis”)

This is wrong!

But, around 10% of them are correct!

“Charlie Chaplin died of a stroke”
Example: Riedel corpus

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of positive to unlabeled</td>
<td>1:134</td>
</tr>
<tr>
<td>Percentage of false negatives</td>
<td>8.5%</td>
</tr>
<tr>
<td>Ratio of positive to false negative</td>
<td>1:11.4</td>
</tr>
</tbody>
</table>

(Min, 2013)
Example: KBP dataset

<table>
<thead>
<tr>
<th>Ratio of positive to unlabeled</th>
<th>1:37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of false negatives</td>
<td>11.5%</td>
</tr>
<tr>
<td>Ratio of positive to false negative</td>
<td>1:4</td>
</tr>
</tbody>
</table>

(Min, 2013)
Formal definition for the MIML Relation Extractor (1)

For each bag, evaluate each mention-level label, for every single relation

“z” still gives the “x” mention labels, or nothing

“y” is still the binary classifier which gives the bag a label for each relation

The difference is “l”, which is a third classifier
Formal definition for the MIML Relation Extractor (2)

The “l” figures out whether the bag should get a positive or negative label for each relation

It will mark false negative bags as positive

The “y” will then use this as input and compare the bags with the knowledge base
Formal definition for the MIML Relation Extractor (3)

**Encouraging false-negatives:**
If “l” is positive (the entity expresses the relation), and it isn’t in knowledge base, then “y” will be positive, instead of “unlabeled”

**Removing false-positives:**
If “l” is negative, and it IS in the knowledge base, this model will prevent “y” from receiving a positive label
Using Expectation Maximisation again to train

**Expectation step:**

Run the classifiers and assign labels to “l” and “z”

**Maximisation step:**

Update the weight matrices for the classifiers
Example of using false negatives

Knowledge base: location-contains(Virginia, Richmond)

Test set: “Richmond, the capital of Virginia”

Sentence is saved into a feature vector, including the information from the word ‘capital’

Also in test set, but not in knowledge base:

“Vienna, the capital of Austria”

Can infer from learned knowledge:

location-contains(Austria, Vienna)
Improvement with false negative pairs?

Comparison with Surdeanu et al. (2012)


Discussion points

Should we be limiting ourselves to two-place tuples? We have multi instance and multi labeled models, but nothing like:

relation(entity1, entity2, entity3)

birth(“Sadiq Khan”, “8.10.1970”, “St George’s Hospital”)

WikiData will carry on growing, will it ever be completed? Do we need to worry about unknown knowledge?

Alternatives to Expectation Maximisation?

Are relations the best thing to use?