DISTANCE LEARNING FOR RELATION EXTRACTION WITHOUT LABELLED DATA

By

Toby Peoples 18.12.13

Introduction

- New approach for extracting relational data from unstructured text without the need of labelled data.
- Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky, Stanford University
- Relation Extraction
  "the task of recognizing the assertion of a particular relationship between two or more entities in text" (Banko & Etzioni, 2008)
  - "Kevin Shields was born in New York"
- Applications; information retrieval, text summarization, question answering

Previous Work

- Previous Approaches have typically relied on relatively small datasets
- Many used little or no information
- More recent approaches use deeper syntactic information
- Similar is the effective method of Wu and Weld(2007)

Previous Work

- Previous learning paradigms
  - Supervised approaches
  - Purely unsupervised information extraction
  - Bootstrap learning

Previous Work

- Supervised approaches
  - Sentences in a corpus are first hand labeled
  - ACE systems then extract features: lexical, syntactic, semantic
  - Supervised classifiers label the relation
- Disadvantages
  - Labeling: time consuming, expensive, few relations, small corpus, does not scale, domain-dependent
  - Labeled on a particular corpus, biased towards text domain.

Previous Work

- Unsupervised approaches
  - Extracts strings of words between entities
  - Can use very large amounts of data
- Disadvantages
  - Resulting relations not easy to map
  - Results questionable: Supervised subcomponents (NER, tagger, parser)

Previous Work

- Bootstrapping
  - Use a very small number of seed instances or patterns.
  - Seeds used with a large corpus in an iterative fashion.
  - Resulting patterns often suffer from low precision and semantic drift (loss of relevance).

Distant Supervision

- Combines some of the advantages of the previous approaches
- An extension of the paradigm used by Snow et al (2005), by using WordNet to extract hypernym relations between entities.
- The algorithm uses a large semantic database called Freebase

Terminology

- 'Relation' refers to an ordered, binary relation between entities
- 'Relation instances' refers to individual ordered pairs.
- Example, the person-nationality relation holds between the entities named 'Stephin Merritt' and 'United States', (Stephin Merritt, United States)

Freebase

- A large semantic database
  - Contains 116 million instances of 7,300 relations between 9 million entities.
  - Data in Freebase is collected from a variety of sources. Wikipedia, NNDB, MusicBrainz, SEC.
  - Freebase also contains the reverses of many of its relations, these are merged.
    e.g (book-author v. author-book)

Freebase

Relation name	Size	Example
/people/person/nationality	281,107	John Dugard, South Africa
/location/location/contains	253,223	Belgium, Niljen
/people/person/profession	208,888	Dusa McDuff, Mathematician
/people/person/place_of_birth	105,799	Edwin Hubble, Marshfield
/dining/restaurant/cuisine	86,213	Mac Ayo's Mexican Kitchen, Mexican

The Freebase relations that are used, with their size and an instance of each relation.

Architecture

- Training step
  - Entities are identified in sentences using a named entity tagger.
  - Sentence containing two freebase entities,features are extracted from that sentence and are added to the feature vector for the relation.
- Example
  - Text 'Footscray is a suburb 5 km west of Melbourne, Victoria, Australia."
  - Freebase /location/australian_suburb
    /location/citytown
  - Training Data (Footscray,Melbourne)
    Label: Suburb, Feature X is a Y

Architecture

- Testing Step
  - Entities again identified using the named entity tagger.
  - Every pair of entities in a sentence is considered a potential relation instance.
  - Example, a pair of entities in 10 sentences and each sentence has 3 features extracted from it, the entity pair will have 30 associated features.
  - Each entity pair is run through feature extraction.
  - Regression classifier predicts a relation name for each entity pair.

Architecture

- Testing Step
  - Location-contains relation,(Virginia,Richmond) & (France,Nantes). 'Richmond, the capital of Virginia.' and 'Henry's Edict of Nantes helped the Protestants of France'

Architecture

- One of the main advantages of the architecture is its ability to combine information from many different mentions of the same relation.
  - (Coen Brothers, The Big Lebowski)
  - ‘‘[The Coen Brothers]'s film [the big Lebowski] is inspired by the work of Raymond Chandler.
  - ‘‘Tim Bevan co-produced the cult film [the big Lebowski], directed by [The Coen Brothers]…
- The first sentence, while providing evidence for film-director, could instead be evidence for film-writer or film-producer.

Features

- Features are based on standard lexical and syntactic features from the literature.
  - Lexical
  - Syntactic
  - Named Entity Tag
  - Feature Conjunction

Lexical features

- The sequence of words between the two entities
- The part-of-speech tags of these words
- A flag indicating which entity came first in the sentence
- A window of k words to the left of Entity 1 and their part-of-speech tags
- A window of k words to the right of Entity 2 and their part-of-speech tags

Lexical features

Feature type	Left window	NE1	Middle	NE2	Right window
Lexical	[]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[]
Lexical	[Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[,]
Lexical	[#PAD#, Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[, Missouri]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]

Features for 'Astronomer Edwin Hubble was born in Marshfield, Missouri'.

Syntactic features

- Features based on syntax
- Each sentence is parsed with the broad-coverage dependency parser MINIPAR
- A dependency parse consists of a set of words ('Edwin Hubble','Missouri') and chunks, linked by directional dependencies('pred','lex-mod')
- For each sentence a dependency path between each pair of entities is extracted.
- Dependency path consists of series of dependencies, directions and words/chunks representing a traversal of the parse.

Syntactic features

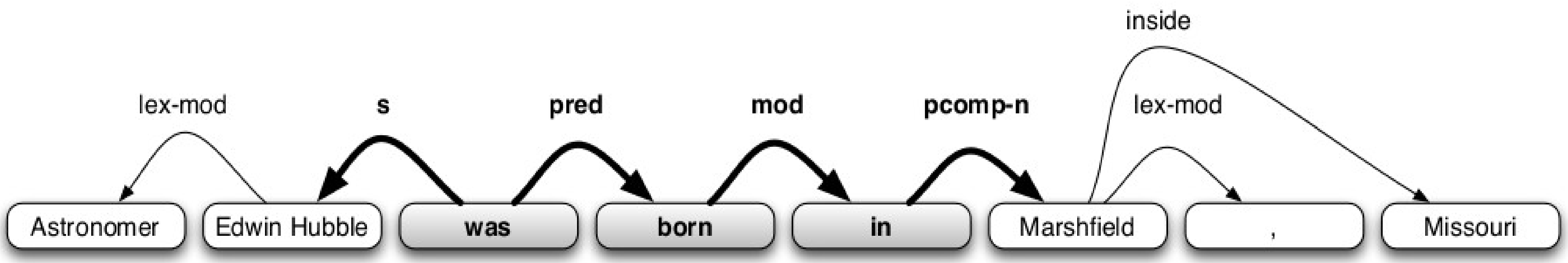


Figure 1: Dependency parse with dependency path from 'Edwin Hubble' to 'Marshfield' highlighted in boldface.

Syntactic features

- Consists of the conjunction of:
  - A dependency path between the two entities
  - For each entity, one 'window' node that is not part of the dependency path
- A window node is a node connected to one of the two entities and not part of the dependency path.

Named entity tag features

- Every feature contains additionally, named entity tags for the two entities.
- The tagger provides each word with a label from {person, location, organization, miscellaneous, none}.

Feature conjunction

- Each feature consists of the conjunction of several attributes of the sentence, plus the named entity tags.
- For two features to match, all of their conjuncts must match exactly. This yields low-recall but high-precision features.

Feature Type	Left Window	NE1	Middle	NE2	Right Window
Lexical	[#PAD#, Astronomer]	PER	[was/VERBborn/VERB in/CLOSED]	LOC	[, Missouri]
Syntatic	[EdwinHubble ⇓lex−mod]	PER	[⇑s was ⇓pred born ⇓mod in ⇓pcomp−n]	LOC	[⇓inside Missouri]

Experiments

- For unstructured text the Freebase Wikipedia Extraction is used.
- The dump consists of approximately 1.8 million articles, an average of 14.3 sentences per article, 601,600,703 words.
- For experiments half of the articles are used:
  - 800,000 for training
  - 400,000 for testing

Experiments

Training and testing

- For held-out evaluation experiments, half of the instances of each relation are not used in training.
- Later used to compare against newly discovered instances.
- For human evaluation experiments, all 1.8 million relation instances are used in training.
- Only relation instances not appear in training data are extracted, i.e. not already in Freebase.

Experiments

Parsing and chunking

- Dependency parsed by MINIPAR to produce a dependency graph.
- Consecutive words with the same named entity tag are 'chunked', so that Bradford/PERSON Cox/PERSON becomes [Bradford Cox]/PERSON.
- Chunking is restricted by the dependency parse of the sentence(i.e., no chunks across subtrees).

Experiments

- System needs negative training data for the purposes of constructing the classifier.
- A feature vector in the training phase is built for an 'unrelated' relations.
- A multi-class logistic classifier returns a relation name and a confidence score
- Afterwards can be ranked with by confidence score and used to generate a list of the n most likely new relation instances.

Experiments

Relation	Feature type	Left window	NE1	Middle	NE2	Right window
/architecture/structure/architect	LEX $\curvearrowleft$		ORG	, the designer of the	PER	
	SYN	designed $\Uparrow_s$	ORG	$\Uparrow_s$ designed $\Downarrow_{by-subj}$ by $\Downarrow_{pcn}$	PER	$\Uparrow_s$ designed
/book/author/works_written	LEX		PER	s novel	ORG	
	SYN		PER	$\Uparrow_{pcn}$ by $\Uparrow_{mod}$ story $\Uparrow_{pred}$ is $\Downarrow_s$	ORG	
/book/book_edition/author_editor	LEX $\curvearrowleft$		ORG	s novel	PER	
	SYN		PER	$\Uparrow_{nn}$ series $\Downarrow_{gen}$	PER	
/business/company/founders	LEX		ORG	co - founder	PER	
	SYN		ORG	$\Uparrow_{nn}$ owner $\Downarrow_{person}$	PER	
/business/company/place_founded	LEX $\curvearrowleft$		ORG	- based	LOC	
	SYN		ORG	$\Uparrow_s$ founded $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/film/film/country	LEX		PER	, released in	LOC	
	SYN	opened $\Uparrow_s$	ORG	$\Uparrow_s$ opened $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ opened

Examples of high-weight features for several relations. Key: SYN = syntactic feature; LEX = lexical feature; = reversed; NE# = named entity tag of entity.

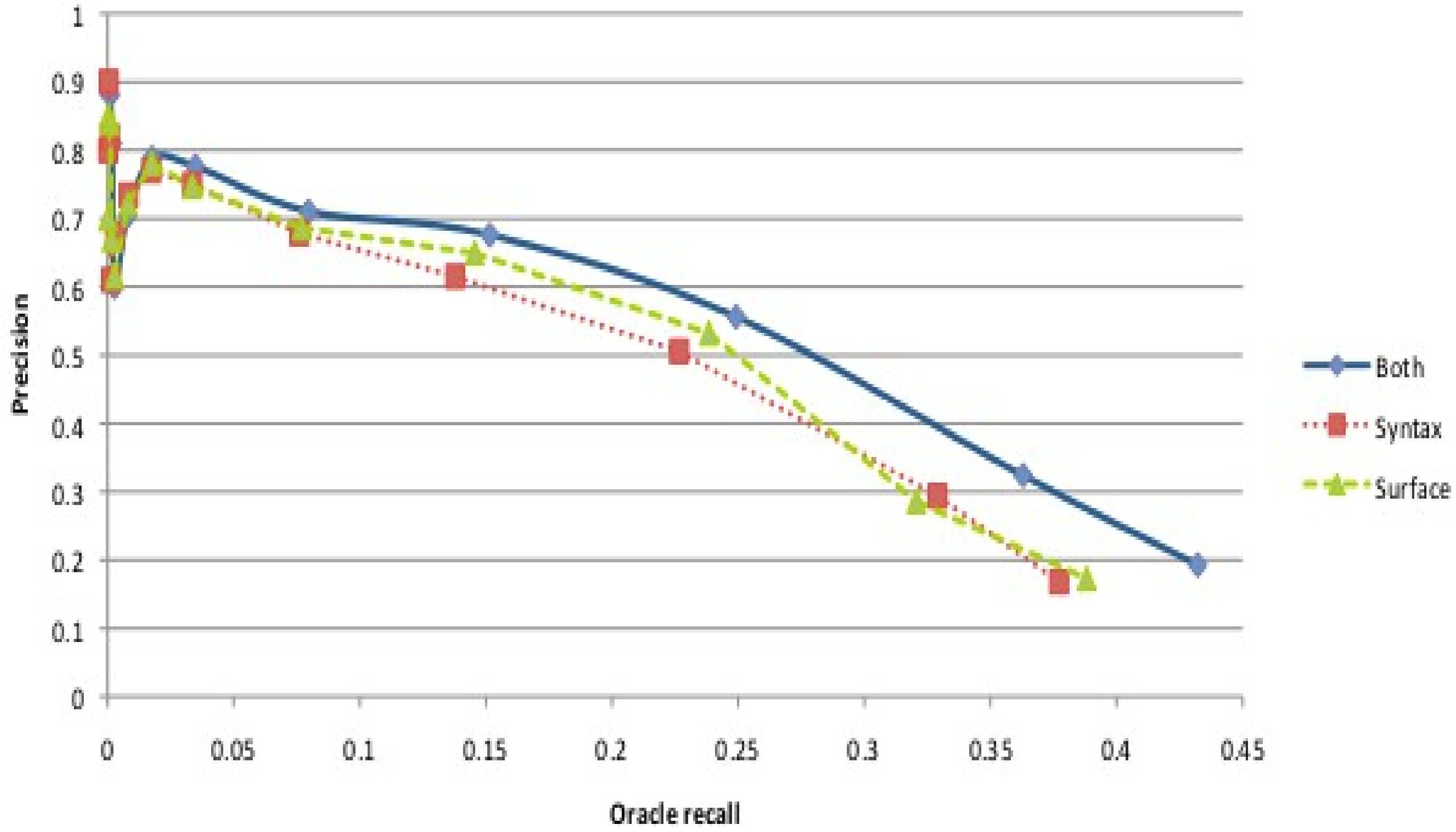
Evaluation

- Labels are evaluated in two ways:
  - Automatically, by holding out part of the data during training, and comparing newly discovered relation instances.
  - Manually, having humans who look at each positively labelled entity pair.
  - Both evaluations allow a precise calculation for the best N instances.

Evaluation

- Held out Evaluation
  - Suffers from false negatives.
  - Gives a rough measure of precision without requiring expensive human evaluation.
  - Useful for parameter setting.
  - Substantial improvement in precision over either of these feature sets on its own.

Evaluation



The performance of the classifier on held-out Freebase relation data

Evaluation

- Human evaluation
  - Performed by evaluators on Amazon's Mechanical Turk service.
- Three experiments were run:
  - one using only syntactic features;
  - one using only lexical features;
  - one using both syntactic and lexical features.

Evaluation

Relation name	100 instances			1000 instances		
	Syn	Lex	Both	Syn	Lex	Both
/film/director/film	**0.49**	0.43	0.44	**0.49**	0.41	0.46
/film/writer/film	**0.70**	0.60	0.65	**0.71**	0.61	0.69
/geography/river/basin_countries	0.65	0.64	**0.67**	**0.73**	0.71	0.64
/location/country/administrative_divisions	0.68	0.59	**0.70**	**0.72**	0.68	**0.72**
/location/location/contains	0.81	**0.89**	0.84	**0.85**	0.83	0.84
/location/us_county/county_seat	0.51	0.51	**0.53**	0.47	**0.57**	0.42
/music/artist/origin	0.64	0.66	**0.71**	0.61	**0.63**	0.60
/people/deceased_person/place_of_death	0.80	0.79	**0.81**	0.80	**0.81**	0.78
/people/person/nationality	0.61	0.70	**0.72**	0.56	0.61	**0.63**
/people/person/place_of_birth	**0.78**	0.77	**0.78**	0.88	0.85	**0.91**
Average	0.67	0.66	**0.69**	**0.68**	0.67	0.67

Estimated precision on human-evaluation experiments of the highest-ranked 100 and 1000 results per relation, using stratified samples.

Summary

- Distant supervision extracts high-precision patterns for a reasonably large number of relations.
- The combination of syntactic and lexical features provides better performance than either feature set on its own.
- Syntactic features consistently outperform lexical features.

Questions?

Thank you for your attention,
Any questions?