Chapter 17, Speech and Language Processing, 3rd edition draft, Jurafsky & Martin (2020).

6.3. Relation Extraction Methods

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

Regular Expressions to Extract Patterns

- Example relation: [holdsOfficeIn]
- \PER, POSITION of ORG\

  George Marshall, Secretary of State of the United States
- \PER (named|appointed|chose|etc.) PER Prep? POSITION\

  Truman appointed Marshall Secretary of State
- \PER [be]? (named|appointed|etc.) Prep? ORG POSITION\

  George Marshall was named US Secretary of State
- Often, high recall requires a large number of patterns.

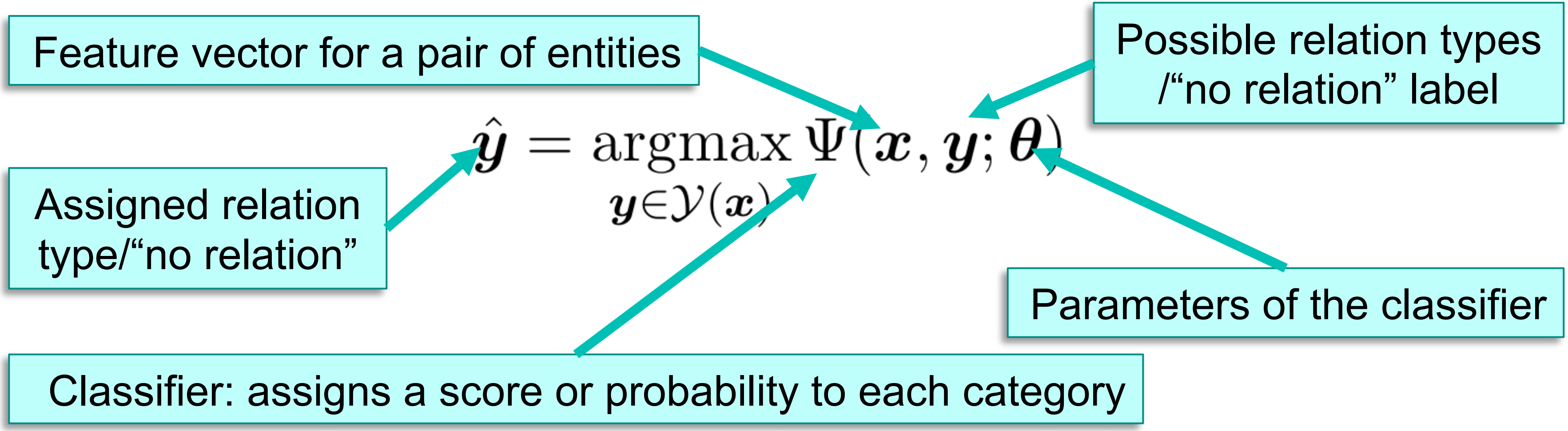
Text Classifiers

- Supervised learning from training data rather than hand-crafting patterns.
- The developers work shifts to designing suitable features.
- How can we apply a classifier to extract relations?

Section 17.2.2, Speech and Language Processing, 3rd edition draft, Jurafsky & Martin (2020).

Text Classifiers for RE

1. List all pairs of named entities mentioned within a sentence.
2. Apply a classifier to each pair, where the class label is either "no relation" or a relation type like [holdsOfficeIn].



$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \Psi(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

Feature vector for a pair of entities
Possible relation types /"no relation" label
Assigned relation type/"no relation"
Parameters of the classifier
Classifier: assigns a score or probability to each category

A Feature Vector for RE

Feature	Entity 1	Entity 2
Unigram	"American" , "Airlines"	"Tim", "Wagner"
UnigramNextToken	"Employs"	"."
UnigramPrevToken	None	"Employs"

- Word features for both entities:
  - Unigrams (bag-of-words)
  - Bigrams
- Word features for neighbouring words:
  - Tokens between the entities
  - Tokens immediately before or after the entities.

A Feature Vector for RE

Feature	Entity 1	Entity 2
Unigram	“American”, “Airlines”	“Tim”, “Wagner”
UnigramNextToken	“Employs”	“.”
UnigramPrevToken	None	“Employs”
EntityType	ORG	PER
	Relation Features	
ConcatenatedTypes	ORG-PER	

- Named entity types
  - Type of entity 1
  - Type of entity 2
  - Concatenation of types of entities 1 and 2

A Feature Vector for RE

- Dependency Path:
  - Airlines, ←nsubj, employs, →obj, Wagner

Feature	Entity 1	Entity 2
Unigram	“American”, “Airlines”	“Tim”, “Wagner”
UnigramNextToken	“Employs”	“.”
UnigramPrevToken	None	“Employs”
EntityType	ORG	PER
	Relation Features	
ConcatenatedTypes	ORG-PER	
DependencyPath	Airlines, ←nsubj, employs, →obj, Wagner	

Feature-based Classifiers for RE

- Put it all together: concatenate into a single vector.
- How do we handle named entity types and path features?
  - Treat them in a similar way to words;
  - Extract a vocabulary for each type of feature.
- Apply any classifier to these feature vectors:
  - Naïve Bayes classifier
  - Logistic regression
  - Neural network

Summary

- Regular expressions can be used to extract some relations.
- Text classifiers can be applied to each pair of entities in a sentence to determine if they are related and what type of relation they have.
- For naïve Bayes or logistic regression, design a feature vector containing word features of the entities and tokens in between, parse tree features, and entity types