Chapter 7: Information extraction

Information Extraction

- Information extraction (IE): turns the unstructured information embedded in texts into structured data
  - Named Entity Recognition (NER): find each mention of a named entity in the text and label its type.

Information Extraction

- Information extraction (IE): turns the unstructured information embedded in texts into structured data
  - Relation Extraction (RE): find and classify semantic relations among the text entities

Information Extraction

- Information extraction (IE): turns the unstructured information embedded in texts into structured data
  - Event Extraction: find events in which these entities participate

Named Entity Recognition (NER)

Named Entity Recognition (NER)

- Detect the entities in the text.
  - A named entity: a person, a location, an organization.
  - Extended term: dates, times, and temporal expressions, prices.

Named Entity Recognition (NER)

Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower- cost carriers. American Airlines, a unit of AMR Corp., immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Denver to San Francisco.

Named Entity Recognition (NER)

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PER Tim Wagner] said. [ORG United], a unit of [ORG UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as [LOC Chicago] to [LOC Dallas] and [LOC Denver] to [LOC San Francisco].

5 organizations, 4 locations, 2 times, 1 person, and 1 mention of money

Named Entity Recognition (NER)

Application of NER:

- extract relationship between participants
- extracting events
- sentiment analysis

NER as Sequence Labeling

- A word-by-word sequence labeling task, in which the assigned tags capture both the boundary and the type
- In IOB tagging we introduce a tag for the beginning (B) and inside (I) of each entity type, and one for tokens outside (O) any entity.

Words	IOB Label	IO Label
American	B-ORG	I-ORG
Airlines	I-ORG	I-ORG
,	O	O
a	O	O
unit	O	O
of	O	O
AMR	B-ORG	I-ORG
Corp.	I-ORG	I-ORG
,	O	O
immediately	O	O
matched	O	O
the	O	O
move	O	O
,	O	O
spokesman	O	O
Tim	B-PER	I-PER
Wagner	I-PER	I-PER
said	O	O
.	O	O

Named entity tagging as a sequence model, showing IOB and IO encodings.

NER as Sequence Labeling

- NER tagging:
  - feature based (HMM, CRF)
  - neural (bi-LSTM)
  - rule-based

A feature-based algorithm for NER

- extract features
- train a sequence classifier (HMM, CRF)
- use it to label the tokens

A feature-based algorithm for NER

identity of w_i, identity of neighboring words
embeddings for w_i, embeddings for neighboring words
part of speech of w_i, part of speech of neighboring words
base-phrase syntactic chunk label of w_i and neighboring words
presence of w_i in a **gazetteer**
w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
w_i is all upper case
word shape of w_i, word shape of neighboring words
short word shape of w_i, short word shape of neighboring words
presence of hyphen

Figure 17.5 Typical features for a feature-based NER system.

A feature-based algorithm for NER

A gazetteer is a list of place names

Example: L’Occitane

prefix(wi) = L

prefix(wi) = L’

prefix(wi) = L’O

prefix(wi) = L’Oc

word-shape(wi) = X’Xxxxxxxx

suffix(wi) = tane

suffix(wi) = ane

suffix(wi) = ne

suffix(wi) = e

short-word-shape(wi) = X’Xx

A feature-based algorithm for NER

- Word shape features: represent the abstract letter pattern of the word by mapping lower-case letters to ‘x’, upper-case to ‘X’, numbers to ’d’, and retaining punctuation.
  - I.M.F would map to X.X.X
  - DC10-30 would map to XXdd-dd
- Short word shape features: consecutive character types are removed.
  - I.M.F would still map to X.X.X
  - DC10-30 would be mapped to Xd-d

Word	POS	Chunk	Short shape	Label
American	NNP	B-NP	Xx	B-ORG
Airlines	NNPS	I-NP	Xx	I-ORG
,	,	O	,	O
a	DT	B-NP	x	O
unit	NN	I-NP	x	O
of	IN	B-PP	x	O
AMR	NNP	B-NP	X	B-ORG
Corp.	NNP	I-NP	Xx.	I-ORG
,	,	O	,	O
immediately	RB	B-ADVP	x	O
matched	VBD	B-VP	x	O
the	DT	B-NP	x	O
move	NN	I-NP	x	O
,	,	O	,	O
spokesman	NN	B-NP	x	O
Tim	NNP	I-NP	Xx	B-PER
Wagner	NNP	I-NP	Xx	I-PER
said	VBD	B-VP	x	O
.	,	O	.	O

.6 Word-by-word feature encoding for NER.

A feature-based algorithm for NER

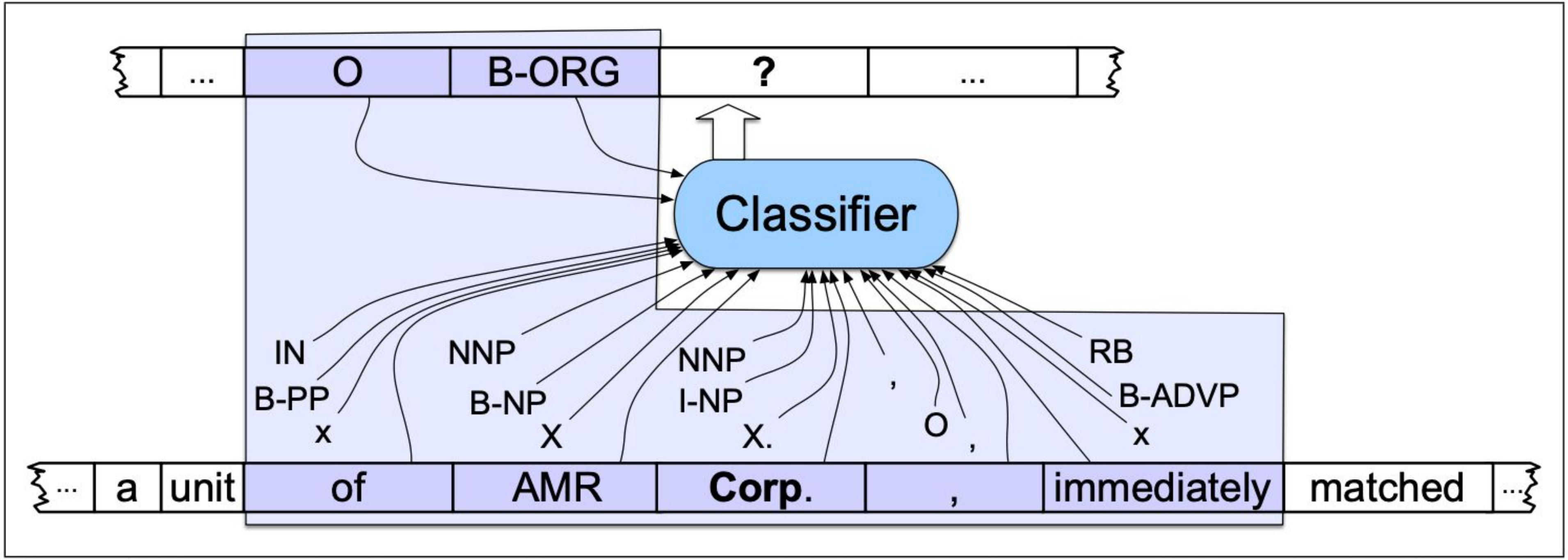


Figure 17.7 Named entity recognition as sequence labeling. The features available to the classifier during training and classification are those in the boxed area.

A neural algorithm for NER

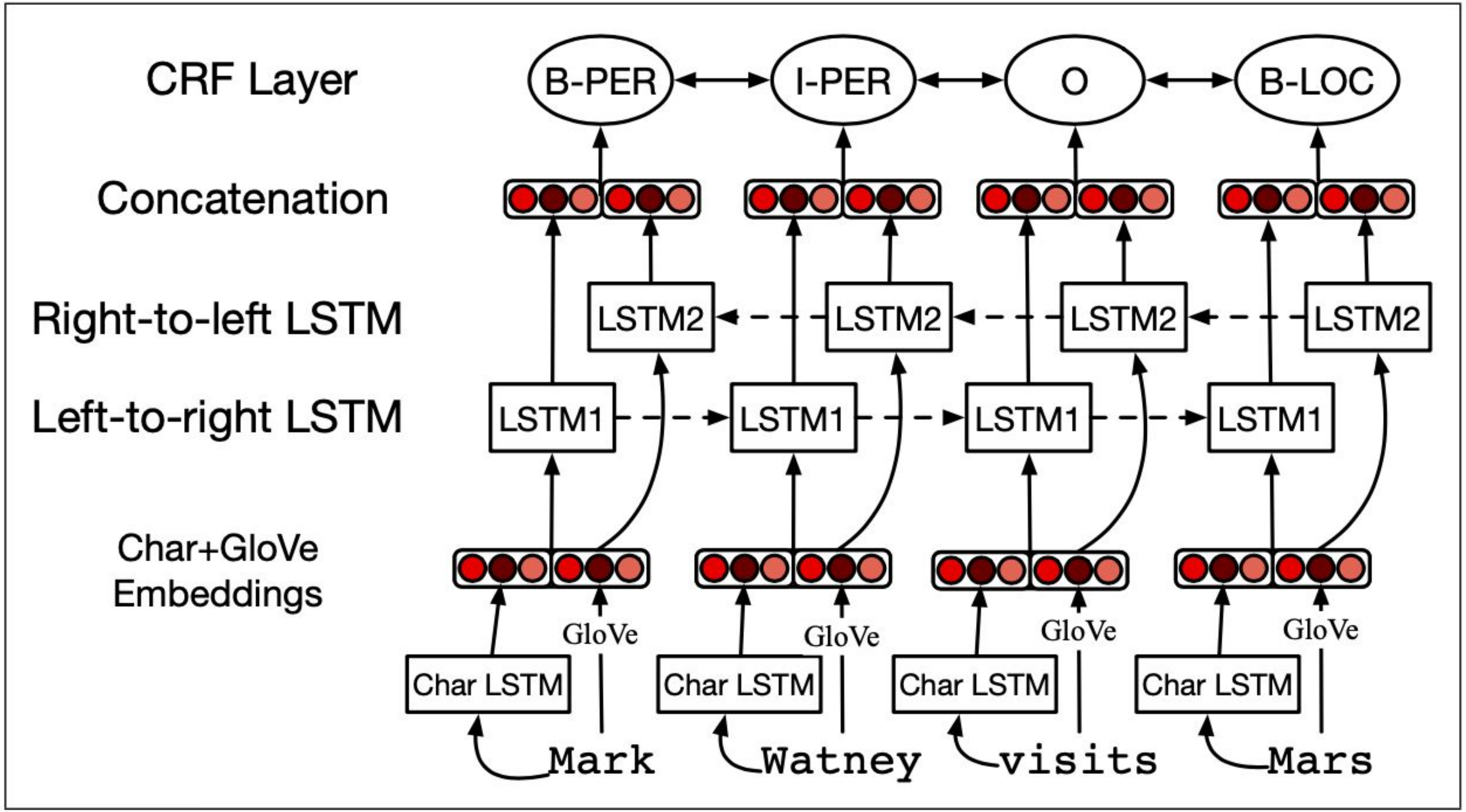


Figure 17.8 Putting it all together: character embeddings and words together a bi-LSTM sequence model. After Lample et al. (2016).

Evaluation of Named Entity Recognition

Recall, Precision, and F1 measure are used to evaluate NER systems

- Recall: the ratio of the number of **correctly labeled responses** to the **total that should have been labeled**
- Precision: the ratio of the number of **correctly labeled responses** to **the total labeled**

2. Relation Extraction

Relation Extraction

- Detect and extract the relationships that exist among the detected entities

Relation Extraction

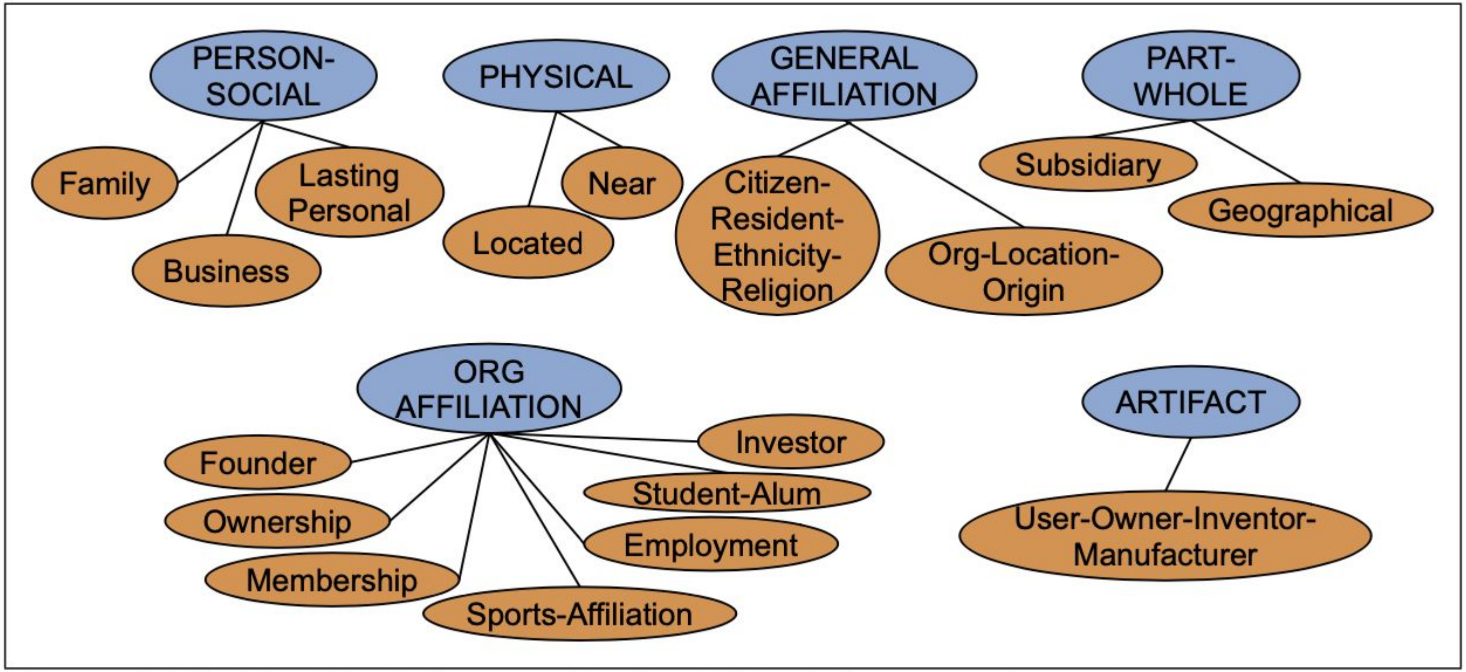


Figure 17.9 The 17 relations used in the ACE relation extraction task.

- Automatic Content Extraction (ACE) Program

Relation Extraction

Relations	Types	Examples
Physical-Located	PER-GPE	**He** was in **Tennessee**
Part-Whole-Subsidiary	ORG-ORG	**XYZ**, the parent company of **ABC**
Person-Social-Family	PER-PER	**Yoko**'s husband **John**
Org-AFF-Founder	PER-ORG	**Steve Jobs**, co-founder of **Apple**...

Figure 17.10 Semantic relations with examples and the named entity types they involve.

2.1 Using Patterns to Extract Relations

Using Patterns to Extract Relations

Lexico-syntactic patterns

NP$_H$ as the parent/hyponym

NP {, NP}* {,} (and|or) other NP$_H$	temples, treasuries, and other important civic buildings
NP$_H$ such as {NP,}* {(or|and)} NP	red algae such as Gelidium
such NP$_H$ as {NP,}* {(or|and)} NP	such authors as Herrick, Goldsmith, and Shakespeare
NP$_H$ {,} including {NP,}* {(or|and)} NP	common-law countries, including Canada and England
NP$_H$ {,} especially {NP}* {(or|and)} NP	European countries, especially France, England, and Spain

Figure 17.12 Hand-built lexico-syntactic patterns for finding hypernyms, using {} to mark optionality (Hearst 1992a, Hearst 1998).

Using Patterns to Extract Relations

- Modern versions of the pattern-based approach extend it by adding named entity constraints.

"Who holds what office in which organization?", we can use patterns like the following:

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

Using Patterns to Extract Relations

Performance

- high-precision, tailored to specific domains
- low-recall

2.2 Relation Extraction via Supervised Learning

2.2 Relation Extraction via Supervised Learning

- A fixed set of relations and entities is chosen. A corpus is hand-annotated with the relations and entities
- The annotated texts are then used to train classifiers to annotate an unseen test set.

2.2 Relation Extraction via Supervised Learning

Feature extraction:

- Word features (as embeddings, or 1-hot, stemmed or not)
- Named entity features
- Syntactic structure

2.2 Relation Extraction via Supervised Learning

Example: *American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said*

<u>Word features:</u>

- The headwords of M1 and M2 and their concatenation
  - Airlines, Wagner, Airlines-Wagner
- Bag-of-words and bigrams in M1 and M2
  - American, Airlines, Tim, Wagner, American Airlines, Tim Wagner

2.2 Relation Extraction via Supervised Learning

Example: American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

Word features:

- Words or bigrams in particular positions
  - M2: -1 spokesman
  - M2: +1 said
- Bag of words or bigrams between M1 and M2:
  - a, AMR, of, immediately, matched, move, spokesman, the, unit

2.2 Relation Extraction via Supervised Learning

Example: American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

<u>Named entity features:</u>

- Named-entity types and their concatenation (M1: ORG, M2: PER, M1M2: ORG-PER)

2.2 Relation Extraction via Supervised Learning

Example: American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

<u>Named entity features:</u>

• Entity Level of M1 and M2 (from the set NAME, NOMINAL, PRONOUN) M1: NAME [it or he would be PRONOUN]

M2: NAME [the company would be NOMINAL]

• Number of entities between the arguments (in this case 1, for AMR)

2.2 Relation Extraction via Supervised Learning

Example: American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

<u>Syntactic structure:</u>

- Constituent paths between M1 and M2

  NP↑NP↑S↑S↓NP

- Dependency-tree paths

  Airlines $\leftarrow_{sub\ j}$ matched $\leftarrow_{com\ p}$ said $\rightarrow_{sub\ j}$ Wagner

2.2 Relation Extraction via Supervised Learning

function FINDRELATIONS(*words*) **returns** *relations*

$relations \leftarrow nil$
$entities \leftarrow \text{FINDENTITIES}(words)$
forall entity pairs $\langle e1, e2 \rangle$ **in** *entities* **do**
  if $\text{RELATED?}(e1, e2)$
    $relations \leftarrow relations + \text{CLASSIFYRELATION}(e1, e2)$

Figure 17.13 Finding and classifying the relations among entities in a text.

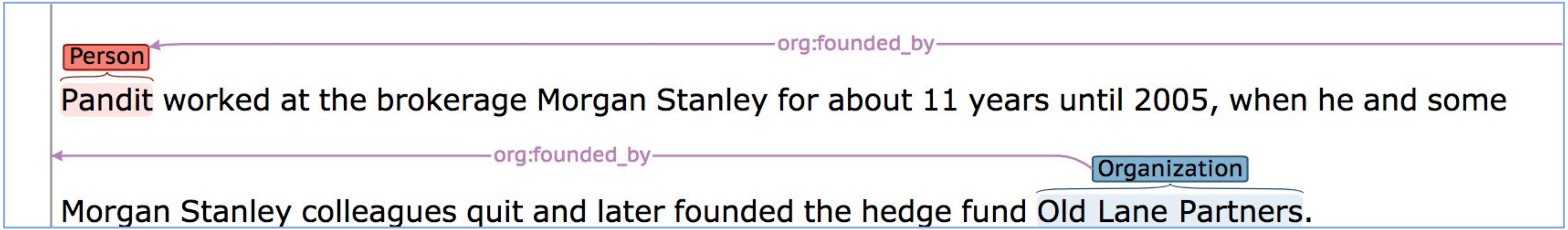
2.2 Neural supervised relation classifiers

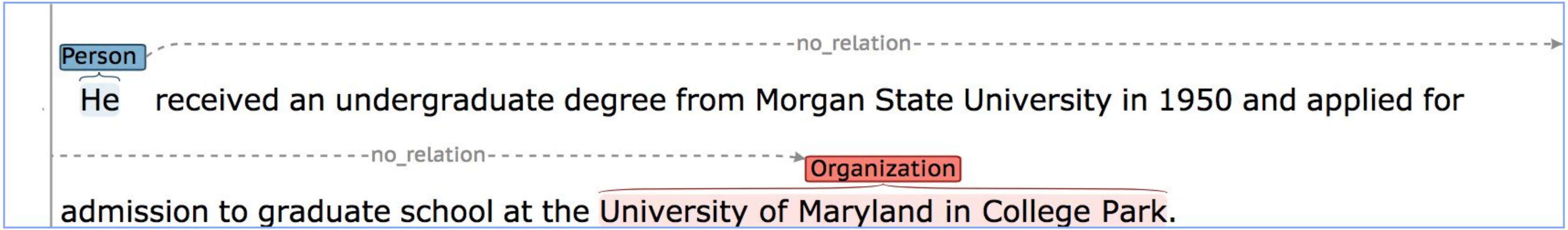
TACRED relation extraction dataset

In each TACRED example, the following annotations are provided:

- the spans of the subject and object mentions;
- the types of the mentions (among 23 fine-grained types used in the Stanford NER system);
- the relation held between the entities (among 41 TAC KBP canonical relation types), or no_relation label if no relation was found.

2.2 Neural supervised relation classifiers





2.2 Neural supervised relation classifiers

https://nlp.stanford.edu/pubs/zhang2017tacred.pdf

https://github.com/yuhaozhang/tacred-relation/tree/master/data

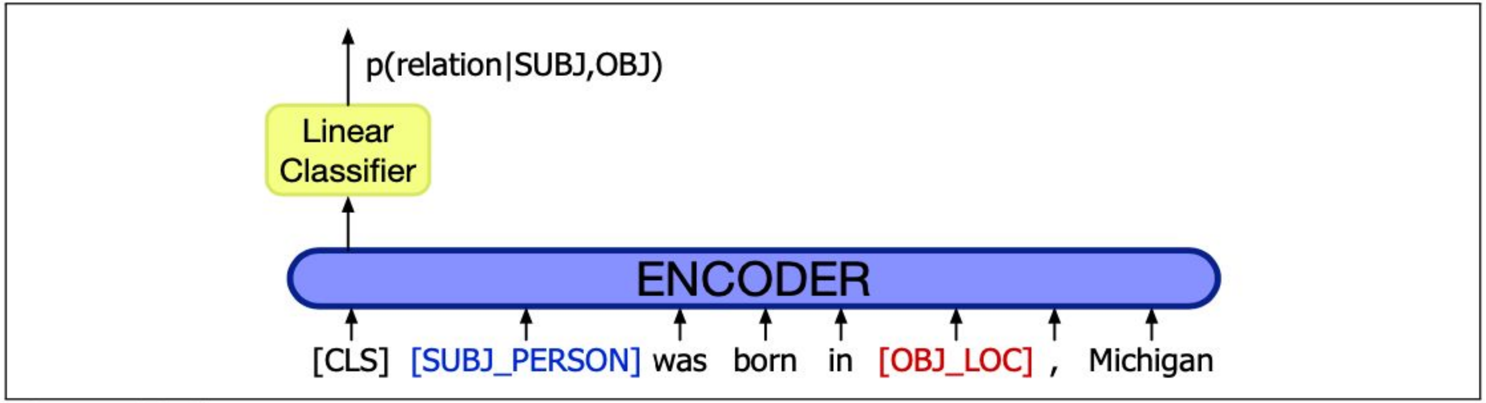


Figure 20.6 Relation extraction as a linear layer on top of an encoder (in this case BERT), with the subject and object entities replaced in the input by their NER tags (Zhang et al. 2017, Joshi et al. 2020).

2.3 Semi-supervised Relation Extraction via Bootstrapping

2.3 Semisupervised Relation Extraction via Bootstrapping

- Supervised machine learning assumes that we have lots of labeled data: expensive.
- With a few high-precision seed patterns, bootstrapping
    - takes the entities in the seed pair
    - then finds sentences that contain both entities.
- From all such sentences, we extract and generalize the context around the entities to learn new patterns.

2.3 Semisupervised Relation Extraction via Bootstrapping

function BOOTSTRAP(*Relation R*) **returns** *new relation tuples*

tuples ← Gather a set of seed tuples that have relation *R*
iterate
sentences ← find sentences that contain entities in *tuples*
patterns ← generalize the context between and around entities in *sentences*
newpairs ← use *patterns* to identify more tuples
newpairs ← *newpairs* with high confidence
tuples ← *tuples* + *newpairs*
return tuples

Figure 20.7 Bootstrapping from seed entity pairs to learn relations.

Conditional Random Field (CRF)

Conditional Random Field (CRF)

- Discriminative sequence model
- We compute the posterior p(Y|X) directly, training the CRF to discriminate among the possible tag sequences.

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y|X)$$

Conditional Random Field (CRF)

Let's assume we have K features, with a weight w_k for each global feature F_k:

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}$$

$$p(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)$$

$$Z(X) = \sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)$$

Conditional Random Field (CRF)

We compute F_k by a sum of local features for each position i in Y :

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

Linear chain CRF: f_k depends on the current and previous output tokens y_i and y_{i-1}

Inference and Training for CRFs

How should we decode to find this optimal tag sequence?

Viterbi algorithm

$$
\begin{aligned}
\hat{Y} &= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y|X) \\
&= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right) \\
&= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \exp\left(\sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)\right) \\
&= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i) \\
&= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)
\end{aligned}
$$

Inference and Training for CRFs

- filling an N × T array with the appropriate values, maintaining backpointers as we proceed.
- follow pointers back from the maximum value in the final column to retrieve the desired set of labels.
- the Viterbi value of time t for state j is computed as:

$$v_t(j) = \max_{i=1}^{N} \left[v_{t-1}(i) + \sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X, t) \quad 1 \leq j \leq N, 1 < t \leq T \right]$$