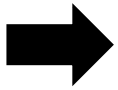
CS60092: Information Retrieval

Information Extraction

Debaditya Roy

Information Extraction (IE)

"Google acquired DeepMind in 2014 for $500 million."



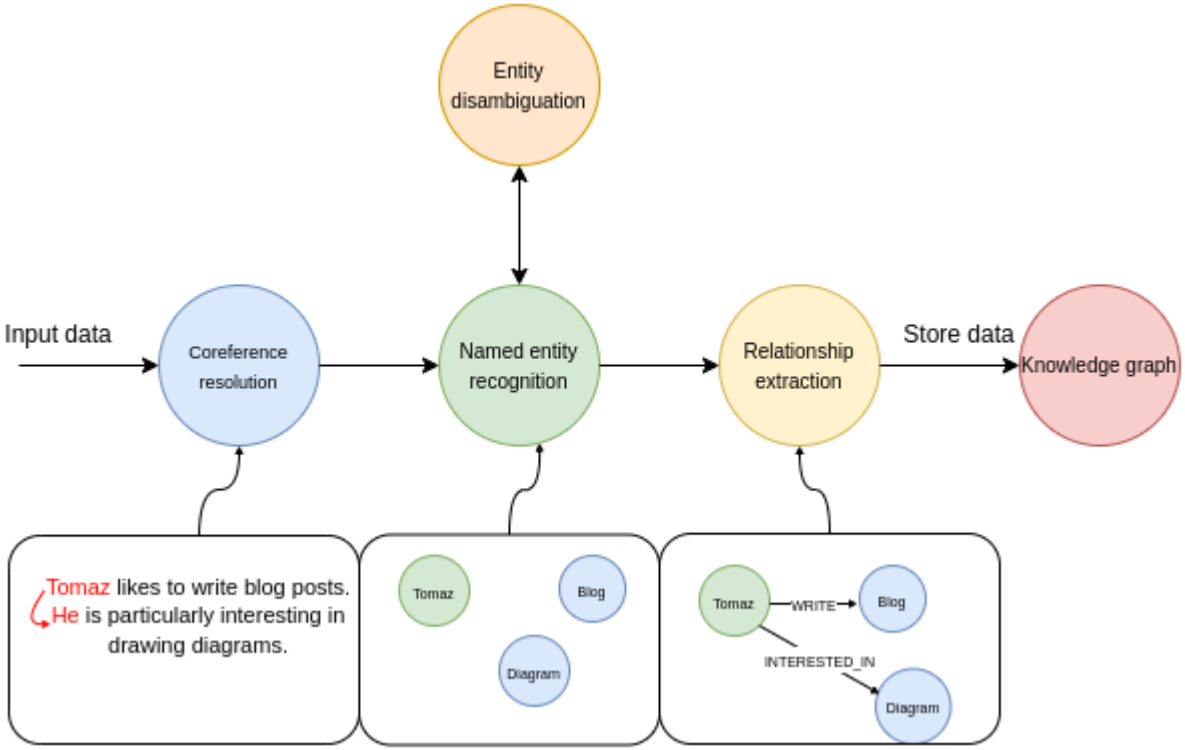
```
Entity:
Google (Organization)
DeepMind (Organization)

Relation:
acquired(Google, DeepMind)

Event:
Acquisition
    Acquirer: Google
    Target: DeepMind
    Time: 2014
    Price: $500M
```

IE answers

- What entities exist?
- How are they related?
- What events occurred?

Information Extraction (IE)

Converting **unstructured natural language** into **structured data** that machines can reason over



https://miro.medium.com/1%2A93ZK9-74dYv4eXY-Oe_bkA.png

BiLSTM-CRF for NER

- **BiLSTM** → captures contextual representation of each word
- **CRF** → enforces globally consistent label sequences

Example: Barack Obama lives in Washington

True labels (BIO scheme)

Word	Label
Barack	B-PER
Obama	I-PER
lives	O
in	O
Washington	B-LOC

Tag	Meaning
B	Beginning of an entity
I	Inside an entity
O	Outside any entity

BiLSTM-CRF

Word	Label
Barack	B-PER
Obama	I-PER
lives	O
in	O
Washington	B-LOC

- A simple neural classifier might predict
  - Barack → B-PER
  - Obama → B-LOC ❌
- B-LOC cannot follow B-PER inside a name
- BiLSTM-CRF solves this by modeling dependencies between labels

Input sentence
↓
Word embeddings
↓
BiLSTM
↓
Token representations
↓
Linear layer (emission scores)
↓
CRF decoding
↓
Best label sequence

Word Embeddings

- Each word becomes a vector

Token embeddings

$$x_1, x_2, x_3, x_4, x_5;\ x_i \in \mathbb{R}^d$$

Example:

John → [0.21, 0.44, ...]

lives → [0.01, 0.88, ...]

New → [0.76, 0.14, ...]

York → [0.73, 0.11, ...]

- Embeddings could be
  - Word2Vec
  - GloVe
  - BERT embeddings

Bidirectional LSTM

- BiLSTM processes the sentence in both directions.
- Forward LSTM: $h_i^{\rightarrow} = LSTM(x_i, h_{i-1})$
- Backward LSTM: $h_i^{\leftarrow} = LSTM(x_i, h_{i+1}^{\leftarrow})$
- Final token representation

$$h_i = [h_i^{\rightarrow}; h_i^{\leftarrow}]$$

- Each word representation knows
  - past context
  - future context

Example: New York. BiLSTM understands that **York depends on New**

Emission Scores

Word	Label
John	B-PER
lives	O
in	O
New	B-LOC
York	I-LOC

- BiLSTM output is mapped to **scores for each possible tag**
- Linear layer $s_i = Wh_i + b$
  - s_i = score vector for token i
  - dimension = number of labels

Emission scores for **York**

Label	Score
B-PER	0.3
I-PER	0.2
B-LOC	1.8
I-LOC	2.4
O	0.1

I-LOC is highest.
Still ignores sequence constraints.

Conditional Random Field (CRF)

- CRF models **dependencies between neighboring labels**
  - B-LOC → I-LOC (valid)
  - B-PER → I-PER (valid)
  - B-LOC → I-PER (unlikely)
- CRF learns a transition matrix $T_{y_i, y_{i+1}}$

From	To	Score
B-LOC	I-LOC	1.5
B-LOC	O	0.8
B-LOC	I-PER	-2.0

- Illegal transitions get **large negative scores**

Global Sequence Score

CRF assigns a score to the **entire label sequence**

- Sentence

$$X = (x_1, \ldots, x_n)$$

- Label sequence

$$Y = (y_1, \ldots, y_n)$$

- Sequence score

$$Score(X, Y) = \sum_{i=1}^{n} s_i(y_i) + \sum_{i=1}^{n} T_{y_{i-1}, y_i}$$

BiLSTM confidence + label transition consistency

Conditional Probability

- CRF converts scores to probabilities.

$$P(Y \mid X) = \frac{\exp\big(Score(X, Y)\big)}{\sum_{Y'} \exp\big(Score(X, Y')\big)}$$

- Denominator sums over **all possible label sequences**
- This is called the **partition function**

During training

- Model maximizes the log-likelihood of the correct label sequence

$$L = -\log P(Y^{true}|X)$$

- maximize correct sequence score
- minimize incorrect sequences

Decoding (Inference)

n = sentence length
k = number of labels

At prediction time we want

$$Y^* = \arg\max_Y Score(X, Y)$$

Computed using the **Viterbi algorithm** $O(nk^2)$

- Instead of evaluating all sequences, compute the **best partial sequence ending at each state**.
- Meaning: Label y at position i comes from the best previous label.

Viterbi Algorithm: A Worked Example

Sentence	John	lives
Labels	PER	O

Input Scores

Emission Scores

How well does each word match each label?

Word	PER	O
John	3	1
lives	0	2

Transition Scores

How likely is one label to follow another?

From	To	Score
START	PER	1
START	O	0
PER	O	1
PER	PER	0
O	O	1
O	PER	-1

Step 1 — Initialisation (Position 1: "John")

$$score\,(label) \;=\; T_{START,label} \;+\; s(John, label)$$

PER

```
trans(START → PER)
  = 1
emit(John, PER)
  = 3
```

Score(PER) = **4**

O

```
trans(START → O)
  = 0
emit(John, O)
  = 1
```

Score(O) = **1**

After Position 1 : PER = 4 | O = 1 → Best so far: PER

Step 2 — Propagation (Position 2: "lives") → Label O

For each label: find the best predecessor, then add emission score

From PER (score = 4)

```
4  +  trans(PER→O)
= 4 + 1
= 5   ✓ BEST
```

From O (score = 1)

```
1  +  trans(O→O)
= 1 + 1
= 2
```

Add emission score:

best predecessor score 5 + emit(lives, O) 2 = **7**

Step 2 — Propagation (Position 2: "lives") → Label PER

From PER (score = 4)

```
4  +  trans(PER→PER)
= 4 + 0
= 4   ✓ BEST
```

From O (score = 1)

```
1  +  trans(O→PER)
= 1 + (−1)
= 0
```

Add emission score:

best predecessor score 4 + emit(lives, PER) 0 = **4**

Final Scores & Backtracking

Final scores at Position 2:

Label	Score	Winner?
PER	4	
O	7	★ Best

Best final label

O

Backtrack:

Position 2 **lives** → **O** ← *backptr: PER*

Position 1 **John** → **PER** ← *backptr: START*

Correct NER sequence: John/PER lives/O ✓

Viterbi DP Table

Each cell stores: best score + backpointer to predecessor label

Position / Word	PER	O
1 — John	4 ← START	1 ← START
2 — lives	4 ← PER	7 ★ ← PER

Reading the backpointers →

- Best final label at position 2 → O (score = 7)
- O at position 2: backpointer → PER (best predecessor)
- PER at position 1: backpointer → START
- Sequence → John / PER, lives / O ✓

Why BiLSTM-CRF?

Advantages

✓ captures context via BiLSTM

✓ enforces label consistency via CRF

✓ global sequence optimization

State-of-the-art NER from 2015–2019

Limitations:

- CRF adds decoding cost
- Long sequences expensive
- Now replaced by Transformer + CRF

References

Speech and Language Processing
Authors: Daniel Jurafsky & James H. Martin

Topic	Chapter
Information Extraction	Chapter 17

Foundations of Statistical Natural Language Processing
Authors: Christopher Manning and Hinrich Schütze