Copyright Notice

deeplearning.ai

Introduction to Named Entity Recognition

What is Named Entity Recognition?

- Locates and extracts predefined entities from text
- Places, organizations, names, time and dates

Types of Entities

Thailand:
Geographical

Google:
Organization

Indian:
Geopolitical

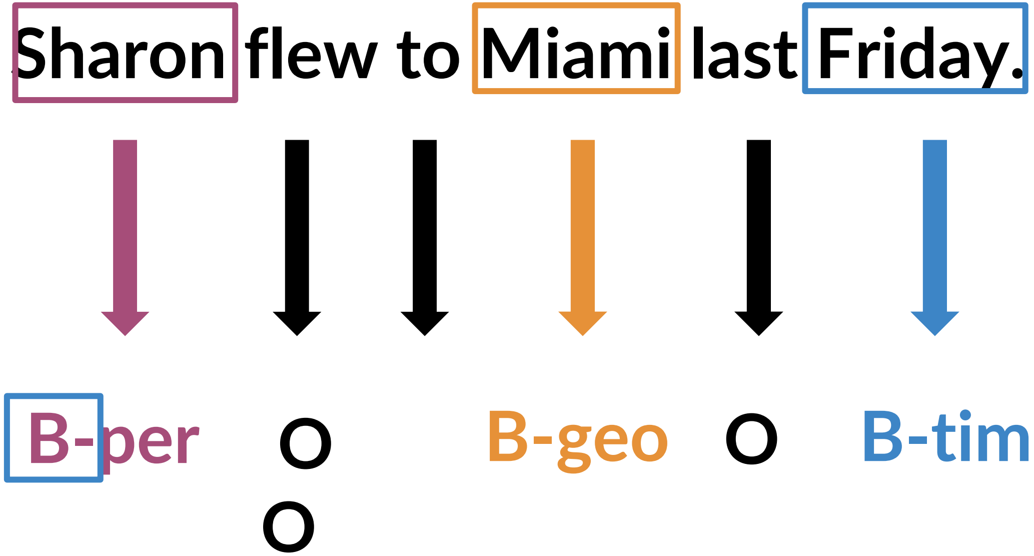
More Types of Entities

December:
Time Indicator

Egyptian statue:
Artifact

Barack Obama:
Person

Example of a labeled sentence



Sharon flew to Miami last Friday.

B-per O O B-geo O B-tim

Applications of NER systems

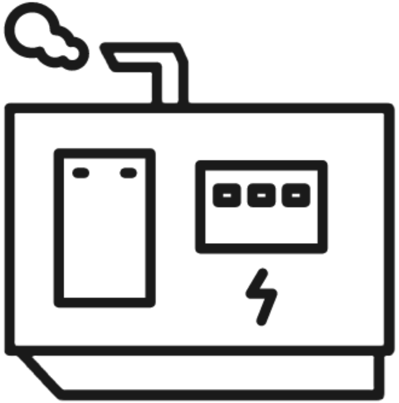
- Search engine efficiency
- Recommendation engines
- Customer service
- Automatic trading

deeplearning.ai

Training NERs: Data Processing

Outline

- Convert words and entity classes into arrays
- Token padding
- Create a data generator



Processing data for NERs

- Assign each class a number
- Assign each word a number

Sharon flew to Miami last Friday.

[4282, 853, 187, 5388, 2894, 7]

1 2 3

B-per O O B-geo O B-tim

Token padding

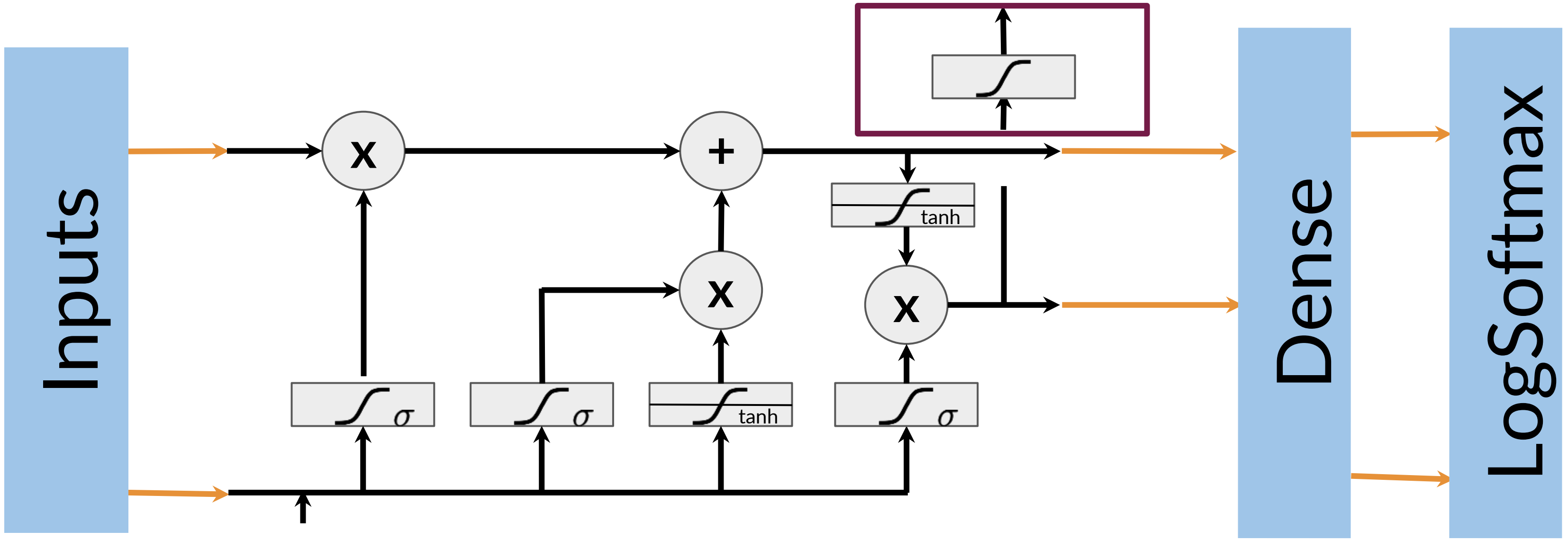
For LSTMs, all sequences need to be the same size.

- Set sequence length to a certain number
- Use the <PAD> token to fill empty spaces

Training the NER

1. Create a tensor for each input and its corresponding number
2. Put them in a batch → 64, 128, 256, 512 ...
3. Feed it into an LSTM unit
4. Run the output through a dense layer
5. Predict using a log softmax over K classes

Training the NER



Layers in Trax

```
model = tl.Serial(
    tl.Embedding(),
    tl.LSTM(),
    tl.Dense()
    tl.LogSoftmax()
  )
```

Summary

- Convert words and entities into same-length numerical arrays
- Train in batches for faster processing
- Run the output through a final layer and activation

deeplearning.ai

Computing Accuracy

Evaluating the model

1. Pass test set through the model
2. Get arg max across the prediction array
3. Mask padded tokens
4. Compare outputs against test labels

Evaluating the model in Python

```
def evaluate_model(test_sentences, test_labels, model):
    pred = model(test_sentences)
    outputs = np.argmax(pred, axis=2)
    mask = ...
    accuracy = 
np.sum(outputs==test_labels)/float(np.sum(mask))

    return accuracy
```

Summary

- If padding tokens, remember to mask them when computing accuracy
- Coding assignment!