University of BRISTOL

5.2. Named Entity Recognition

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

bristol.ac.uk

What Do We Mean By "Spans"?

- A text span is a sub-sequence within a larger piece of text.
- I.e., One or more words in succession.
- For example, "United Airlines" is a span of two tokens within this text:

"Citing high fuel prices, United Airlines said …"

Named Entity Recognition (NER)

- Goal: label each word in a text sequence with entity type/no entity.
- Which sequence labelling task did we learn about before that is also useful for obtaining information for downstream tasks?
- Unlike part-of-speech tags, an entity can have more than one token.
- NER is more complex as we need to extract spans, that is, identify boundaries of spans as well as labelling tokens.
- Can spans overlap or be nested inside each other?

"New York Police Department"

NER as Sequence Labelling:

- Assume non-overlapping entities.
- We can label each token with either an entity type or ‘O’ for ‘outside’ of any span.

O	O	O	ORG	ORG	ORG	ORG	O	ORG
Several	airlines,	including	American	Airlines,	United	Airlines	and	BA…

- What problems are there with this scheme?
- Can you come up with a way to improve it?

NER as Sequence Labelling:

- Tags capture the entity type and the boundaries of the span.
- BIO tagging: beginning, inside or outside. (Ramshaw & Marcus, 1995)

O	O	O	B-ORG	I-ORG	B-ORG	I-ORG	O	B-ORG
Several	airlines,	including	American	Airlines,	United	Airlines	and	BA…

- O: not an entity
- B-: beginning of an entity span.
- I-: inside an entity span.
- 2n+1 different tags, where n is the number of entity types.

Sequence Labelling Models: Recap

- Can you recall two sequence labelling methods from our previous lectures?

Sequence Labelling Models

- Hidden Markov model (HMM):
  - Generative probabilistic graphical model.
  - Transition probability depends on previous tag.
- Conditional random field (CRF):
  - Discriminative, undirected probabilistic graphical model.
  - Transition probability depends on a neighbourhood of tags and features.

Sequence Labelling for NER

- We could apply an HMM or CRF.
- For all of these, we need a suitable feature vector for each token in the sequence.
- E.g., for the HMM, we need a vector $\boldsymbol{x}_i$:

$$P(y_i|\boldsymbol{x}_i) \propto P(\boldsymbol{x}_i|y_i)P(y_i|y_{i-1})$$

- Let's design a feature vector for NER…

A Feature Vector for NER

- On the right is an empty feature vector for two tokens, which we are going to fill up.
- The tokens are: "*Old*" and "*Sodbury*" from a sentence *"The bus service does not run to Old Sodbury on Sundays."*
- What kind of features should we include?

Token 1	Token 2

A Feature Vector for NER

- Unigrams: the tokens themselves or the index of the token in the vocabulary.
- Is the association between a word and a sequence label enough to find named entities?
  - Cannot recognise proper nouns that were not seen in the training data.
  - Can't recognise names like "Old Sodbury" that contain common words.

Token 1	Token 2
"Old"	"Sodbury"

A Feature Vector for NER

- Bigrams: pairs of consecutive words.
- We can also concatenate the feature vector of neighbouring words.
- Does that provide enough information for NER?
  - That might help with “New York”…
  - But we still can’t recognise proper nouns that were not seen in the training data!
  - Yet it’s easy for a human reader to spot a name – why?

Feature	Token 1	Token 2
Unigram	”Old”	“Sodbury”
Bigram	[“to”, “Old”]	[“Old”, “Sodbury”]
Bigram	[“Old”, “Sodbury”]	[“Sodbury”, “on”]

A Feature Vector for NER

- Prefixes and suffixes:
  - Identify any common suffixes and prefixes.
  - In some languages, names may have particular prefixes or suffixes.
- Presence in a list of known names (gazetteer):
  - E.g., a list of given names in the USA
  - Probably contains more names than any training dataset.

Feature	Example for token 1	Example for token 2
Unigram	”Old”	“Sodbury”
Bigram	[“to”, “Old”]	[“Old”, “Sodbury”]
Bigram	[“Old”, “Sodbury”]	[“Sodbury”, “on”]
Prefix	None	None
Suffix	None	“bury”
InPlaceList	No	Yes

A Feature Vector for NER

- Part-of-speech tags:
  - Provide syntactic clues.
  - Is the word a proper noun? If yes, it's probably a named entity.
- Other syntactic information, e.g., chunk tags:
  - Similar to parts of speech but describe the type of phrase a word is part of, rather than the type of the individual word
  - Named entity spans often correspond to *noun phrases*

Feature	Token 1	Token 2
Unigram	”Old”	“Sodbury”
Bigram	[“to”, “Old”]	[“Old”, “Sodbury”]
Bigram	[“Old”, “Sodbury”]	[“Sodbury”, “on”]
Prefix	None	None
Suffix	None	“bury”
InPlaceList	No	Yes
POS	PROPN	PROPN
Chunk	NP	NP

Feature Engineering and Deep Learning

- To choose the best set of features, evaluate the model's performance with different sets of features on a development set.
- Deep learning tries to reduce the need for feature engineering:
  - The features are learned automatically from a training set.
  - Deep learning is performed using neural networks with many hidden layers.
- Feature engineering incorporates expert *knowledge* about the task and about linguistics, while deep learning is *data-driven.*
- Downsides: huge amounts of training data + high computational costs of training and prediction + high memory costs

Summary

- We can model NER as a sequence labelling task using BIO tagging.
- A sequence taggers such as an HMM, MEMM or CRF can be trained to do NER.
- Features include word features, POS tags, chunks.