

# Lecture 09

Dihui Lai

dlai@wustl.edu

March 24, 2020

# Part of Speech (POS) Tagging

- General Part-of-speech (POS) categories:
  - nouns (data, chair),
  - verbs (debate, discuss),
  - adverbs (slowly),
  - adjectives (funny)
  - etc.
- Fine-grained POS categories:
  - nouns: proper noun (Lisa, Paris) v.s. common nouns (woman, city);
  - adverb: comparative (more happily) v.s. superlative (most happily)
  - etc.
- A word could have multiple POS tags e.g.
  - A number of factors account [verb] for the differences between the two scores.
  - How do I open an account [noun] with your bank?

# English Penn Treebank part-of-speech Tagset

An important tagset for English is the 45-tag Penn Treebank tagset.

[https://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

# POS-tag Model

- Label the words in a document using POS tags, e.g.  
The[DT] Itek[NNP] Air[NNP] Boeing[NNP] 737[CD] took[VBD]  
off[RP] bound[VBN] for[IN] Mashhad[NNP] in[IN] north-eastern[JJ]  
Iran[NNP].
- If a word  $w$  that could be tagged as  $t_1, t_2, \dots, t_k$ , the probabilities the word has tagged  $t_i$  is calculated as  $p(t_i|w) = \frac{c(w, t_i)}{\sum_{i=1}^k c(w, t_i)}$

**This approach does not take the order of the word into consideration!**

# POS tag Sequencing Model

Provided that we have a sequence of words  $W = w_1, w_2, \dots, w_i, \dots, w_n$  and we want to figure out the their POS tags  $T = t_1, t_2, \dots, t_i, \dots, t_n$

Using Bayes' theorem

$$P(T|W) = P(W|T)P(T)/P(W) = \text{const} \times P(W|T)P(T)$$

Assume that  $t_i$  is only dependent on  $t_{i-1}$  and  $w_i$ , we have

$$\begin{aligned} P(T) &= P(t_1)P(t_2|t_1)P(t_3|t_1, t_2)P(t_4|t_1, t_2, t_3) \dots P(t_n|t_1, t_2, \dots, t_{n-1}) \\ &= P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3) \dots P(t_n|t_{n-1}) \end{aligned}$$

$$P(W|T) = P(w_1|t_1)P(w_2|t_2)P(w_3|t_3) \dots P(w_n|t_n)$$

# POS tag Sequencing Model (Continued)

In summary, we have

$$P(T|W) \approx P(t_1)P(t_2|t_1)...P(t_n|t_{n-1})P(w_1|t_1)P(w_2|t_2)...P(w_n|t_n)$$

Each term on the right hand side of the equation can be calculated as

$$P(t_i|t_{i-1}) = \frac{c(t_{i-1}, t_i)}{c(t_{i-1})} \text{ (transition probability)}$$

$$P(w_i|t_i) = \frac{c(w_i, t_i)}{c(t_i)} \text{ (emission probability)}$$

where

$c(t_i)$  = count of  $t_i$  in the corpus,

$c(w_i, t_i)$  = count of  $(w_i, t_i)$  in the corpus,

$c(t_{i-1}, t_i)$  = count of  $(t_{i-1}, t_i)$  in the corpus

# POS tag Sequencing Model (Continued): Hidden Markov Model and Viterbi Algorithm

The best tag sequence is the sequence that maximize the conditional probability  $P(T|W)$  i.e.

$$\begin{aligned}(t_1, t_2, \dots, t_n) &= \arg \max_{t_1, t_2, \dots, t_n} P(t_1, t_2, \dots, t_n | w_1, w_2, \dots, w_n) \\ &= \arg \max_{t_1, t_2, \dots, t_n} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})\end{aligned}$$

To solve the problem, we can use the Viterbi Algorithms for Hidden Markov Model

# POS tag Model: the State of the Art

[https://aclweb.org/aclwiki/POS\\_Tagging\\_\(State\\_of\\_the\\_art\)](https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art))



# Name Entity Recognition (NER)

- POS tag usually label a single word. However, to understand text, we also need to consider the meaning of a set of words.
- NER labels words in a texts that are names of things e.g. person, organization, money amount, gene/protein names
- John (person) Lee (person) is the chief of CBSE (organization).
- the output of POS-tag could be used as input to accomplish a NER model.

# NER and IOB Format Tag

IOB: I-Inside, O-Outside, B-Begin

Example: Alex is going to Los Angeles

Alex I-PER

is O

going O

to O

Los B-LOC

Angeles I-LOC

# NER model and the State of the Art

- The state of art algorithms: <https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003>
- Useful packages, softwares, services  
<https://nlp.stanford.edu/software/CRF-NER.shtml>  
[http://nlp\\_architect.nervanasys.com/installation.html](http://nlp_architect.nervanasys.com/installation.html)  
<https://cloud.google.com/natural-language/>