Lecture Note - 10: POS Tagging, HMM, NER

Dihui Lai

November 15, 2020

Contents

1 Part-of-Speech (POS) Tagging

As you perhaps learned in grammer class that a word could be one the following types: noun, verb, article, adjective, preposition, pronoun, adverb etc. These categories are called part of speech tags. We can certainly have a refined version of the tags. An important tagset for English is the 45-tag Use Penn Treebank tagset. Here is one example of POS tag for a sentence using Penn Treebank:

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].

How can we get a program to understand the POS tag of a word in a sentence? A most straight forward way is to calculate the probability using the following algorithm

If a word w that could be tagged as t_1, t_2, ... t_k, the probability that the word has tag t_i is

$$p(t_i|w) = \frac{c(w, t_i)}{\sum_{i=1}^{k} c(w, t_i)}$$

However, this approach does not take the context of the word into consideration, which is very important to POS tagging. For example the word *account* could be a noun or verb depending on the context:

A number of factors *account* [VERB] for the differences between the two scores.

How do I open an *account* [NOUN] with your bank?

2 Hidden Markove Model

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

Using Bayes' theorem, the conditional probability of having a POS tag sequence T given a word sequence W is

$$P(T|W) = P(W|T)P(T)/P(W) = const \times P(W|T)P(T) \tag{1}$$

Generally, we have

$$P(T) = P(t_1)P(t_2|t_1)P(t_3|t_1, t_2)P(t_4|t_1, t_2, t_3)...P(t_n|t_1, t_2, ...t_{n-1})$$

Assume that t_i is only dependent on t_{i-1} (Markov assumption) we have

$$P(T) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)...P(t_n|t_{n-1}) \tag{2}$$

The conditional probability $P(t_i|t_{i-1})$ is called transition probability. It describe the probability that a tag t_i appears after a tag t_{i-1}. To calculate the emission probability we can use the following equation

$$P(t_i|t_{i-1}) = \frac{c(t_{i-1}, t_i)}{c(t_{i-1})} \text{ (transition probability)} \tag{3}$$

Where $c(t_{i-1}, t_i)$ is the frequency that tag t_i occurs next to t_{i-1} and $c(t_i)$ is the frequency that tag t_i occurs in a document.

To calculate the term $P(W|T)$ in equation, we assume that w_i is only dependent on t_i and

$$P(W|T) = P(w_1|t_1)P(w_2|t_2)P(w_3|t_3)...P(w_n|t_n) \tag{4}$$

The condtional probabily $P(w_i|t_i)$ is called emission probability, and can be calculated as

$$P(w_i|t_i) = \frac{c(w_i, t_i)}{c(t_i)} \text{ (emission probability)} \tag{5}$$

Where $c(w_i, t_i)$ is the frequency that a word w_i is tagged as t_i or that a tag t_i is used for word w_i. In the denominator, $c(t_i)$ is the frequency that the tag t_i occurs in a document.

In summary, we have

$$P(T|W) \sim P(t_1)P(t_2|t_1)...P(t_n|t_{n-1})P(w_1|t_1)P(w_2|t_2)...P(w_n|t_n) \tag{6}$$

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

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

To solve the problem, we use an algorithm called Viterbi Algorithms.

3 Named Entity Recognition

POS tag helps understanding document at a single word level. Yet we often need to go beyond single word level to understand the semantics of a document. When read independently, the word *white* and *house* means color and the place where people live. When put together, *white house* means a special place where the American president live. In order to understand thw words' meaning as a group. We use an algorithm called named entity recogition(NER). NER labels word chucks in a corpus that are names of things e.g. person, organization, money amount, gene and proteins etc.

For exmpale, Alex[$PERSON$] is going to Los [$LOCATION$] Angeles [$LOCATION$]

One chanllenge of NER tagging is that not all words belong to an entity. In the example abve, words *is going to* do not belong to any entity. On the other hand, Los Angeles belong to the same entity but we are not reflecting this fact by tagging them separately.

A better way of tagging the entiteis properly is using a tagging method called IOB [*I-Inside*], [*O-Outside*], [*B-Begin*].

For the example above, we tag the words as below

Alex[*I-PER*]
is[*O*]
going[*O*]
to[*O*]
Los[*B-LOC*]
Angeles[*I-LOC*]