codalab user name : powerstar   accuracy : 92.33%

Part-of-Speech Tagging with HMMs

Dept of Computer science, University of Houston

1 Introduction

Part-of-speech (POS) tagging also known as lexical analysis is an important task in natural language processing. POS tagging is the task of identifying nouns, verbs, adjectives, adverbs, and more in a given sentence. Several methods are employed to do POS tagging. These methods are broadly classified as Statistical approaches and Rule based approaches. This paper describes statistical approach of POS tagging implemented through Hidden Markov Model (HMM). HMM is a statistical sequence labelling model aiming towards generalization of training data.

In this paper, the improvements to the basic HMM model are made by applying smoothing techniques and handling the infrequent words. The designed model has the structure of a typical Bigram hidden markov model that estimates the emission probabilities and transition probabilities. The highest scoring function can be found by Viterbi algorithm.

2 Estimating the HMM parameters

The foremost step in evaluating HMM parameters is to go through the tagged corpus and estimate the probabilities by counting the occurrences of tags, tagtag pairs, and tagword pairs. These are very useful in calculating word given tag probability, transition probabilities of tags. Smoothing is done on the transition probabilities. The two probability dictionaries are computed. They are emission probability, named as word corpus and transition probability, named as tag corpus.

3 Viterbi algorithm

Viterbi algorithm is to find the optimal path in a hidden Markov model. This algorithm looks for the most probable path in a graph where node consist of states and edges consists of probabilities.

To tag a sentence, first the Viterbi algorithm is applied. The input to this algorithm is the sentence, and the two probability tables, word corpus and tag corpus. Two matrices are created with every possible tag as rows and individual word from input sentence as columns. The matrix Viterbi dictionary stores the probabilities and the matrix backpointer stores the corresponding tags.

During initialization, the Viterbi function initializes all the probability values to infinite and all the backpointer tags to None. During Recursion, it computes Viterbi probabilities of every possible combinations to reach the next level and picks one among them with the highest probability value. The three factors that are multiplied to compute the Viterbi probability at time t are [2]:

1.previous Viterbi path probability

2.emission probability

3.transition probability

The tag corresponding to this highest probability is stored in the backpointer matrix. During Termination, it retraces the steps back to the initial dummy link and returns the most likely path through the HMM state space.

4 Experiments:

The experiments are done with the tagging sets taken from conversational texts. The train set and development (dev) set has three columns, the first containing the words, the second with the language identification and the third with the POS tags. Test set is provided for which the POS prediction is to be obtained.

4.1 Combining language ID with words:

Since in the train/dev set there are two languages used, there is a need to identify the correct tag given a language ID. One way to solve this is by using a Trigram model i.e., by calculating the probability of the tag given the word and the language id. This gives the model wider tag context. Another approach is to concatenate the word and the language id to form a new word. This concatenated word is used throughout in this experiment. Bigram model is used to find the probability of the tag given the concatenated word.

Bigrams P(ti —ti1) = C(ti1,ti) / C(ti1) [3]

By using this way, the accuracy of the system has been increased by 1.5%

4.2 Laplace smoothing

Laplace smoothing has been implemented to smooth the transition probabilities of the bigram types. Just because an event doesnt occur in the training set mean it wont occur in the test set. There is chance of getting zero count in the numerator with the unseen bigram types. There is a need to give some probabilities to these unseen types in the train set, otherwise the system may perform badly. By smoothing, we add 1 to the numerator and divide it by the length of the total bigram types.
P(tn—tn1) =(count(tn1,tn)+1)/(count(tn1)+T)
where T is the number of distinct tags.
The purpose of smoothing is to shift some probability from seen types to unseen types.

4.3 Tagging unknown words

Typically, the tagging accuracy is much lower for words that arent observed in the training set. To improve the accuracy, it is very essential to deal with the unknown words. The words with the frequency less than or equal to cut-off are termed as unknown words, represented with UNK. In the given train/dev set, there are 5806 words with the cut-off = 1. The most frequent tag for these unknown words has been assigned when these words are seen in the test data. Thus the accuracy increased by around 4%. Experimenting with cut-off = 2 doesn't seems to increase the accuracy.

5 Error Analysis

The purpose of development set is used to analyse the system and improve the model. The error analysis is carried out with the development set. The model is trained on the training set and POS tagging is done on the development set.The state of art accuracy has reached 91.64%. Further on analyzing the results, the model failed to predict the words such as ['lookingeng'], ['soengADV'], ['sA\xadspaINTJ']

5.1 Analysis 1

The model failed to predict the word 'lookingeng' correctly. On analyzing from the word corpus for the word: 'lookingeng'

['lookingeng'] 'Tags': 'VERB': 'Count': 53, 'Prob': -6.778870754861816, 'Total count': 53.

This comes clear that, lookingeng has occurred at 53 instances in train set, all of which were tagged as verb. So the model predicted it as VERB. However in the dev set, it was actually tagged as NOUN.

Lack of context: This is a common problem in machine learning algorithms. Hidden markov model works by generalization of word occurrence pattens across the corpus. By tokenizing the sentences, such approaches loses the essential context. It is hard for the model to understand the context and the sense in which it was used.

5.2 Analysis 2

- Actual: ['mediumengADJ', 'sizeengNOUN', 'soengADV', '. . . ?eng&spaPUNCT']

- Predicted:['mediumengADJ','sizeengNOUN', 'soengSCONJ', '. . . ?eng&spaPUNCT']

- Prediction failed for: ['soengADV']

  The following shows the word corpus for the word: 'soeng'

  word corpus - ['soeng'] 'Tags': 'SCONJ': 'Count': 756, 'Prob': -2.4897450141860418, 'INTJ': 'Count': 473, 'Prob': -3.3922782314330253, 'ADV': 'Count': 310, 'Prob': -4.141187106484205, 'Total count': 1539

  From above, it is clear that ['soeng'] has highest emission probability to be tagged as SCONJ than ADV. This in turn made the Viterbi probability highest to be tagged as SCONJ.

5.3 Analysis 3

- Actual: [['sA\xadspaINTJ'], ...]

- Predicted: [['sA\xadspaX'], ...]

- Prediction failed: ['sA\xadspaINTJ']
  I found for more than 40% of instances, the prediction failed when it actually starts the sentence.

codalab user name : **powerstar** **accuracy** : **9233%**

6 Improvements

The model can be improved by handling the start and the end tags properly. This is done by including start and end symbols before and after each sentence. Instead of Statistical approach to tag a sentence, rule based approach can be implemented that should increase the accuracy of tagging. Rule based approaches requires a lot of human effort to prepare rules and resources such as part-of-speech taggers, bilingual dictionaries etc. [1] A more sophisticated model could be a developed by hybrid approach that utilizes both the approaches to perform the task.

7 Challenges in POS tagging

- One of the key challenges in Part-of-speech tagging is the tokenization of sentences. Tokenization is the practise of breaking a sentence into words. This should be an easy task for languages where words are separated by whitespaces. But it is not the case for languages like Chinese, Japanese where there is no separation between words.

- Another key challenge for POS tagging is the ambiguity in natural language. Many words in English, for instance, can be associated with multiple parts-of-speech, and it is important to understand the context for proper POS tagging.

8 Conclusion

This paper shows that a feature rich hidden markov model using Viterbi algorithm can reach reasonably good accuracies. The accuracy of the POS taggers reaches human level for a well-structured corpora, such as The Economic Times article or a news item from the Financial Times. However, with conversational text such as Snapchat or SMS texts achieving high accuracy for POS tagging is much harder. This is because there is no strict rules to grammar or syntax in conversational texts. There are hybrid approaches that make the best use of both Statistical and Rule based approaches to achieve better results with conversational text.

References

[1] Vishaal Jatav, Ravi Teja, Srini Bharadwaj, and Venkat Srinivasan. Improving part-of-speech tagging for nlp pipelines. *arXiv preprint arXiv:1708.00241*, 2017.

[2] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London:, 2014.

[3] Wiebke Wagner. Steven bird, ewan klein and edward loper: Natural language processing with python, analyzing text with the natural language toolkit. *Language Resources and Evaluation*, 44(4):421–424, 2010.