NLP

Assignment - 05

-By Ayushi Gupta (20162312)

PROBLEM STATEMENT

• To get a more comprehensive understanding of chunking and chunk types see the posguidelines.pdf attached under this topic

• Using the algorithms learnt in the classes before, develop a chunker.

• Use all other linguistic resources we have dealt with previously : POS, Morph etc.

• Data–English : import this data from nltk: http://www.nltk.org/howto/corpus.html#chunked-corpora

here is the document related to the data: http://www.cnts.ua.ac.be/conll2000/pdf/12732tjo.pdf

• Develop a POS LM

• From the chunk-annotated corpora develop a chunk-level LM.

–First make an LM for chunk tags.

–Generate some chunk-sequences from that LM. are they grammatical ?.

–Make chunk-tag specific LMs i.e. P(wordi|wordi−1,chunk − tag) for each word.

–Now, given a chunk-tag, you can generate words in a chunk.

–Combine all of this and you get a chunk-based LM generation

• Compare POS LM and Chunk LM .

PART A :

Step1:

Brown corpus downloaded from below nltk link:

http://www.nltk.org/howto/corpus.html#chunked-corpora

Step2:

Divided Conell corpus in training and testing dataset.

75%- training (8936 sentences)

25% - testing (2012)

Step3:

PART1: Training

1. Read tagged corpus
2. Tokenize the tagged corpus
3. Find all bi-grams in training dataset.
4. Find list of all words, tags and chunks in training corpus.
5. Compute $\mathbf{P(t_i|t_{i-1})}$ (transition probability)
6. Compute $\mathbf{P(w_i|t_i)}$ (emission probability)
7. Compute $\mathbf{P(ch_i|ch_{i-1})}$ (probability of chunk given previous chunk)
8. Compute $\mathbf{P(ch_i|t_i)}$ (probability if chunk given tag)
9. Find probability of every tag being a starting tag (Applied Add-1 smoothing while computing this field)
10. Find probability of every chunk being starting chunk (Applied Add-1 smoothing while computing this field)

PART2 : Testing

1. Read test data
2. For each sentence

- Using emission $\mathbf{P(w_i|t_i)}$ and transition probabilities $\mathbf{P(t_i|t_{i-1})}$, list of all possible tags and words computed on training data using viterbi algorithm find max probability path of tags
- Using emission $\mathbf{P(ch_i|t_i)}$ and transition probabilities $\mathbf{P(ch_i|ch_{i-1})}$, list of all possible chunks and tags computed on training data using viterbi algorithm find max probability path of tags

3. Compute efficiency of code.

```
For each sentence :
        For every word:
                If predicted_chunk==actual_chunk:
                        correct+=1
                Total +=1
efficiency=(correct/total)*100
```

EXPERIMENTAL RESULTS:

A pack of CoNELL corpus was selected to compute the efficiency of viterbi algorithm.

Total Number of training sentences : 8936

Total Number of testing sentences : 2012

Efficiency :

Count of total number of chunks predicted : 47377

Total tags chunks correctly = 13903

efficiency= 29.6%

PART B: