POS Tagging Using HMM and Rule-based Chunking

Asif Ekbal  
Samiran Mandal  
Sivaji Bandyopadhyay  
Computer Science & Engineering Department  
Jadavpur University  
Kolkata

Email: ekbal_asif12@yahoo.co.in  
samiranjucs@gmail.com  
sivaji_cse_ju@yahoo.com
Part of Speech (POS) Tagging

- What is POS tagging?
  → POS Tagset

- Why POS tagging?

- Machine Learning approaches to POS tagging
POS tagging using HMMs

Let $W$ be a sequence of words

$$W = w_1, w_2, \ldots, w_n$$

Let $T$ be the corresponding tag sequence

$$T = t_1, t_2, \ldots, t_n$$

Task: Find $T$ which maximizes $P(T | W)$

$$T' = \arg\max_T P(T | W)$$
POS tagging using HMMs

By Bayes Rule,

\[ P(T | W) = \frac{P(W | T) * P(T)}{P(W)} \]

\[ T' = \text{argmax}_T \ P(W | T) * P(T) \]

Transition Probability,

\[ P(T) = P(t_1) * P(t_2 | t_1) * P(t_3 | t_1 t_2) \ldots * P(t_n | t_1 \ldots t_{n-1}) \]

Applying Tri-gram approximation,

\[ P(T) = P(t_1) * P(t_2 | t_1) * P(t_3 | t_1 t_2) \ldots * P(t_n | t_{n-2} t_{n-1}) \]

Introducing a dummy tag, $, to represent the beginning of a sentence,

\[ P(T) = P(t_1 | $) * P(t_2 | $ t_1) * P(t_3 | t_1 t_2) \ldots * P(t_n | t_{n-2} t_{n-1}) \]
POS tagging using HMMs

- **Smoothing Transition Probabilities**
  - Sparse data problem
  - Linear interpolation method
    
    \[
    P(t_i | t_{i-2}, t_{i-1}) = \lambda_1 P(t_i) + \lambda_2 P(t_i | t_{i-1}) + \lambda_3 P(t_i | t_{i-2}, t_{i-1})
    \]
    
    such that the \(\lambda\)s sum to 1
POS tagging using HMMs

*Calculation of $\lambda$s*

set $\lambda_1 = \lambda_2 = \lambda_3 = 0$

for each tri-gram $t_1, t_2, t_3$ with $freq(t_1, t_2, t_3) > 0$

depending on the maximum of the following three values:

- case $\frac{freq(t_1, t_2, t_3) - 1}{freq(t_1, t_2) - 1}$: increment $\lambda_3$ by $freq(t_1, t_2, t_3)$
- case $\frac{freq(t_2, t_3) - 1}{freq(t_2) - 1}$: increment $\lambda_2$ by $freq(t_1, t_2, t_3)$
- case $\frac{freq(t_3) - 1}{N - 1}$: increment $\lambda_1$ by $freq(t_1, t_2, t_3)$

end

dependence

normalize $\lambda_1, \lambda_2, \lambda_3$
POS tagging using HMMs

- Emission Probability,

\[ P(W | T) \approx P(w_1 | t_1) * P(w_2 | t_2) * \ldots * P(w_n | t_n) \]

- Context Dependency

To make more dependent on the context the emission probability is calculated as:

\[ P(W | T) \approx P(w_1 | $ t_1) * P(w_2 | t_1 t_2) \ldots * P(w_n | t_{n-1} t_n) \]
POS tagging using HMMs

- Smoothing technique is applied
  \[ P'(w_i | t_{i-1}, t_i) = \theta_1 P(w_i | t_i) + \theta_2 P(w_i | t_{i-1}, t_i) \]
  Sum of all \( \theta \)'s is equal to 1

- \( \theta \)'s are different for different words.

- Calculation of \( \theta \)'s for every word takes a considerable time.

- In general, \( \theta \)'s are calculated from the bi-gram tag sequence following the algorithm same as \( \lambda \)'s
POS tagging using HMMs

\[
P(w_{i-2} \mid t_{i-2}) \quad P(w_{i-1} \mid t_{i-1}) \quad P(w_i \mid t_i) \quad P(w_{i+1} \mid t_{i+1})
\]

\[
P(t_{i-2} \mid t_{i-4} t_{i-3}) \quad P(t_{i-1} \mid t_{i-3} t_{i-2}) \quad P(t_i \mid t_{i-2} t_{i-1}) \quad P(t_{i+1} \mid t_{i-1} t_i)
\]

2\textsuperscript{nd} order Hidden Markov Model
POS tagging using HMMs

2nd order Hidden Markov Model (Proposed)

January 8, 2007
POS tagging using HMMs

Complexity

- Considering all possible tag sequence of a given word sequence takes exponential time.

- For the tag set of size $S$ then run time of the approach is $O(S^{|W|})$
Viterbi Algorithm

for \( i = 1 \) to Number_of_Words_in_Sentence
    for each state \( c \in \text{Tag\_Set} \)
        for each state \( b \in \text{Tag\_Set} \)
            for each state \( a \in \text{Tag\_Set} \)
                for the best state sequence ending in state \( a \) at time \((i-2)\), \( b \) at time \((i-1)\), Compute the probability of that state sequence going to state \( c \) at time \( i \)
                end
            end
        end
    end
end
determine the most-probable state sequence ending in state \( c \) at time \( i \)
end

◆ Complexity

Viterbi algorithm runs in \( O(S^3 \times |W|) \) time
Handling of Unknown Words

- Viterbi algorithm attempts to assign a tag to the unknown word
- Approaches adopted to handle unknown words in Bengali
  - Unknown Word Features
  - Named Entity Recognition
  - Lexicon Features
Handling of Unknown Words

- Unknown Word Features

- Instead of $P(w_i | t_i)$ we take the probability $P(f_i | t_i)$ which is calculated based on the feature of unknown word

- Feature
  - Suffixes
    - 435 different suffixes.
    - Null suffixes.
    - Probability distribution of a particular suffix with respect to specific POS tags is generated from all words in the training set that share the same suffix.
  - Symbol
  - Number
Handling of Unknown Words

- Named Entity Recognition in Bengali
  - Pattern directed shallow parsing approach
  - Lexical Seed Patterns Generation
    - Training corpus searched for NNPC and NNP tags
    - *Lexical* pattern \( p \) using a context window of width 4 around the left and right tags considered, e.g.,
      \[
      P = [ l_{-2} l_{-1} <T> l_{+1} l_{+2} ],
      \]
      where \( l_{\pm i} \) are the context of \( p \) and \( T \) is NNPC* NNP.
    - Any of \( l_{\pm i} \) may be a *punctuation symbol* (e.g.; , ? etc.).
    - All these patterns form the set of potential patterns, denoted by \( P \)
Handling of Unknown Words

- Generation of new patterns through bootstrapping
  - Every pattern \( p \in P \) matched against the entire training corpus
  - \( p \) predicts one boundary of a name where the context of \( p \) matches
  - System considers all possible noun, verb and adjective inflections during matching
  - Maximum length of NE considered to be of six words
  - New patterns generated from the existing patterns of the set of potential patterns \( P \)
  - Bootstrapping applied on the training corpus until no new patterns can be generated
Handling of Unknown Words

- A particular threshold value of relative frequency chosen.
- Patterns (along with the type) having more or equal to this threshold value added to the new pattern table.
- Patterns (only for that type) having 'relative frequency' of less than threshold values are discarded.
- All these patterns form the set of accepted patterns (denoted by Accept Pattern).
- These patterns of the Accept Pattern set applied on the test set.
Handling of Unknown Words

Lexicon Features

- Lexicon has been developed in an unsupervised way from a tagged Bengali news corpus.
- Lexicon contains the root words and their basic part of speech information (e.g. noun, verb, pronoun, adjective, adverb and indeclinable).
- Lexicon contains 20000 entries.
- Unknown words (from the test set) found in the lexicon assigned the appropriate POS tags.
Evaluation of POS Tagger

- **POS tagger** is trained by Bengali, Hindi and Telegu training data.

- **POS tagger** is tested on the development sets of Bengali, Hindi and Telegu.

- After testing, the development test sets are included as part of the training sets to evaluate the system with the unannotated tests sets of Bengali, Hindi and Telegu.

- **POS tagger** assigns tags to all the words in test sets and hence the *precision* and *recall* figures of the POS tagger are same and have been considered as the *accuracy* of the tagger.
Evaluation of POS Tagger

Training, Development and Test sets statistics:
- No. of tokens in the training set (NTTR)
- No. of tokens in the development and unannotated test sets (NTTST)
- No. of unknown tokens in the test set (UTST)
- Percentage of unknown tokens in the test set (UTST in %).

<table>
<thead>
<tr>
<th>Language Type</th>
<th>NTTR</th>
<th>NTTST</th>
<th>UTST</th>
<th>UTST (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>20396</td>
<td>5022</td>
<td>1002</td>
<td>19.96</td>
</tr>
<tr>
<td>Hindi</td>
<td>21470</td>
<td>5681</td>
<td>1132</td>
<td>19.93</td>
</tr>
<tr>
<td>Telegu</td>
<td>21415</td>
<td>6098</td>
<td>3411</td>
<td>55.94</td>
</tr>
</tbody>
</table>

Table 1: Training and development test sets statistics

<table>
<thead>
<tr>
<th>Language Type</th>
<th>NTTR</th>
<th>NTTST</th>
<th>UTST</th>
<th>UTST (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>25418</td>
<td>5225</td>
<td>1726</td>
<td>33.04</td>
</tr>
<tr>
<td>Hindi</td>
<td>27151</td>
<td>4924</td>
<td>1416</td>
<td>28.76</td>
</tr>
<tr>
<td>Telegu</td>
<td>27513</td>
<td>5193</td>
<td>2375</td>
<td>45.74</td>
</tr>
</tbody>
</table>

Table 2: Training and unannotated test sets statistics
Evaluation of POS Tagger for the Development Sets

- POS tagger tested with Bengali, Hindi and Telegu development sets
- POS tagger produces 86.38% accuracy for the Bengali development set with the unknown word features only
- Accuracy increases upto 88.02% with the inclusion of NER
- Accuracy further boosted to 90.9% with the inclusion of lexicon features

<table>
<thead>
<tr>
<th>Test Set</th>
<th>CTT</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>4564</td>
<td>90.9</td>
</tr>
<tr>
<td>Hindi</td>
<td>4661</td>
<td>82.05</td>
</tr>
<tr>
<td>Telegu</td>
<td>3898</td>
<td>63.93</td>
</tr>
</tbody>
</table>

Table 3: Result of the POS tagger for the development test sets (CTT: No. of tokens correctly tagged by the POS tagger)
Evaluation of POS Tagger for the Unannotated Sets

- POS tagger is then tested with three different unannotated test sets
- CTT: No of tokens correctly tagged by the POS tagger

<table>
<thead>
<tr>
<th>Test Set</th>
<th>CTT</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>4061</td>
<td>77.73</td>
</tr>
<tr>
<td>Hindi</td>
<td>4924</td>
<td>76.87</td>
</tr>
<tr>
<td>Telugu</td>
<td>3505</td>
<td>67.49</td>
</tr>
</tbody>
</table>

Table 4: Result of the POS tagger for the unannotated test sets
Evaluation of POS Tagger for the Unannotated Sets

- Initially POS tagger produces 73.17% accuracy for the Bengali unannotated test set with the unknown word features only.
- With the release of annotated test set, accuracy increases upto 77.73% with the inclusion of named entity recognition system and lexicon features.
- Accuracy of the POS tagger falls from 90.9% to 77.73% for the unannotated test set.
  - Unannotated test set contains more unknown words than the development test set.
Evaluation of POS Tagger

- POS tagger performs best for Bengali test sets.
  - Key to this higher accuracy, compared to Hindi and Telegu, is the mechanism of handling of unknown words.
  - Unknown word features, named entity recognizer and lexicon features are used to cope with the unknown words in the Bengali data.
- System cannot handle unknown words problem in Hindi and Telegu at present.
- POS tagger performs better with the Hindi compared to Telegu.
- Presence of large number of unknown words in the Telegu test sets (development and unannotated test sets) and the agglutinative nature of the language are the main reasons behind the fall in accuracy.
Evaluation of POS Tagger

- Error analysis of the POS tagger for the Bengali test set done with the help of a confusion matrix
- Unknown words have a tendency of being assigned noun tags (NN, in most cases) through probability calculation.
- **Tagging errors:**
  - NNC vs NN, JJ vs NN, JJ vs NN, JJ vs JVB, VFM vs VAUX, VRB vs NN
- Multiword extraction unit for Bengali would have taken care of the NNC vs NN problem
- Problem of JJ vs NN is hard to resolve and probably requires the use of linguistic rules.
- Other problems can be solved by linguistic rules
Chunking

- Non-overlapping regions of text
- Rule-based approach

**Chunk Boundary Identification**
- some handcrafted rules that determine which POS(s) can belong to the same chunk.

**Chunk Labeling**
- some handcrafted rules that determine the label with consisting POS(s)

**Chunker is designed for Bengali and then applied for Hindi and Telegu**
Chunk Boundary Identification

✓ Necessary to find the positions where a chunk can end and a new chunk can begin
✓ Positions also marked
✓ POS tag assigned to every token by the POS tagger used to discover these positions
✓ Chunk boundaries identified by some handcrafted linguistic rules
✓ Rules check whether two neighboring POS tags belong to the same chunk or not
✓ A chunk boundary in between the words is assigned if they do not belong to the same Chunk
For $i = 2$ to No_of_Word

$(i-1)$ position will be a chunk boundary if all of the following cases don’t satisfy:

a) $t_{i-1} \in \{\text{INTF}\}$ and $t_i \in \{\text{INTF, JJ, PRP, QF, QFNUM, NN, NNC, NNP, NNPC}\}$

b) $t_{i-1} \in \{\text{JJ, QF, VJJ}\}$ and $t_i \in \{\text{JJ, PRP, QF, QFNUM, NN, NNC, NNP, NNPC}\}$

c) $t_{i-1} \in \text{QFNUM}$ and $t_i \in \{\text{PRP, QFNUM, NN, NNC, NNP, NNPC, VNN}\}$

d) $t_{i-1} \in \text{VFM, VAUX, VJJ, VNN}$ and $t_i \in \{\text{NEG, RP}\}$

e) $t_{i-1} \in \{\text{NEG}\}$ and $t_i \in \{\text{VFM, VAUX, VNN, VJJ, VRB, JVB, NVB}\}$

f) $t_{i-1} \in \{\text{NNC, NNPC}\}$ and $t_i \in \{\text{NN, NNC, NNP, NNPC}\}$

g) $t_{i-1} \in \{\text{NN, NNP, PRP, VNN}\}$ and $t_i \in \{\text{PREP, RP, NLOC}\}$

h) $t_{i-1} \in \{\text{RB}\}$ and $t_i \in \{\text{RB}\}$

i) $t_{i-1} \in \{\text{JVB, NVB}\}$ and $t_i \in \{\text{VFM, VAUX}\}$

j) $t_{i-1} \in \{\text{PREP, NEG, NLOC, RP}\}$ and $t_i \in \{\text{RP}\}$

End

Last word of the sentence is also marked as a chunk boundary.
Chunk Labeling

✅ Rules for Chunk labeling

- A chunk will be labeled as **NP** if it contains at least one noun.
- If a chunk contains an adjective but not any noun then it will be labeled as **JJP**.
- A chunk will be labeled as **VG** if it contains at least one verb.
- A chunk will be labeled as **RBP** if it contains at least one adverb.
- A chunk will be labeled as **CCP** if it contains only a word tagged as **CC** or **QT**.
- Any other chunk will be labeled as **BLK**.
# Evaluation of Chunker

- **No. of chunks in the test set (NOCT)**
- **No. of correctly identified chunks (CIC)**
- **Chunk accuracy (CA)**

<table>
<thead>
<tr>
<th>Test Set</th>
<th>NOCT</th>
<th>CIC</th>
<th>CA (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>3915</td>
<td>3362</td>
<td>85.85</td>
</tr>
<tr>
<td>Hindi</td>
<td>2757</td>
<td>2037</td>
<td>73.88</td>
</tr>
<tr>
<td>Telegu</td>
<td>3556</td>
<td>1990</td>
<td>55.96</td>
</tr>
</tbody>
</table>

Table 5: Results for Chunking (Development test sets)

<table>
<thead>
<tr>
<th>Test Set</th>
<th>NOCT</th>
<th>CIC</th>
<th>CA (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>5225</td>
<td>4213</td>
<td>80.63</td>
</tr>
<tr>
<td>Hindi</td>
<td>4924</td>
<td>3528</td>
<td>71.65</td>
</tr>
<tr>
<td>Telegu</td>
<td>5193</td>
<td>2760</td>
<td>53.15</td>
</tr>
</tbody>
</table>

Table 6: Results for Chunking (Unannotated test sets)
Conclusion and Future Works

- HMM based POS Tagger and Rule-based Chunker initially designed for Bengali
- The same tagger and chunker are then used for Hindi and Telegu
- Accuracy figures of Chunker can be increased by deriving the rules of chunking for Hindi and Telegu languages
- Future work includes incorporation of other statistical learning techniques such as MEMM and CRF for POS tagging