Bagging & System Combination for POS Tagging

Dan Jinguji
Joshua T. Minor
Ping Yu
Bagging

• Bagging can gain substantially in accuracy
• The vital element is the instability of the learning algorithm
• Bagging slightly degrades the performance of stable algorithm
Bagging Results

• In all three learning algorithms, the performance of one bag is slightly worse than the baseline

• Bagging has different effects on the three baseline learning algorithms when 10 bags were used
Bagging Error Reduction

Bagging Effectiveness Over Baseline Systems

% Error Reduction

Systems

- TBL
  - MaxEnt
  - Trigram

10K
5K
1K
Bagging and Stability

- Bagging had the greatest effect on MaxEnt
- Bagging actually had a negative effect on trigrams
- Therefore we could say MaxEnt is the least stable of the ML algorithms tested
System Combination

• Two basic methods (work on any number of inputs):
  – Random Tag
    • Choose one of the input tags at random
  – Simple Voting
    • Count each input tag, output highest count tag
    • Ties go to the last tag seen

• Weighted Voting (three base systems only):
  • Train the ML systems on 80% of the training data
  • Use these as confidence scores for voting
  • Basically like regular voting with default to best system (TBL)
Combination Effectiveness

Combination System Effectiveness Over Single System Average

Combination Technique

- Weighted
- Voting
- Random

-5.000 0.000 5.000 10.000 15.000 20.000 25.000 30.000 35.000
% Error Reduction

10K 5K 1K
Hybrid Bagging

• Bags from all three systems combined into one pool.
• Using regular voting
  – Ideally would have used weighted voting
• Best performance of all:

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>5K</th>
<th>10K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>92.6568</td>
<td>95.2401</td>
<td>95.8752</td>
</tr>
<tr>
<td>Error Reduction</td>
<td>12.63</td>
<td>7.45</td>
<td>6.67</td>
</tr>
<tr>
<td>Over Voting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error Reduction</td>
<td>36.17!</td>
<td>27.86</td>
<td>26.03</td>
</tr>
<tr>
<td>Over Average</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overall Improvement Comparison

Systems

- Combined Bagged over Average
- Combined Bagged over Voting
- Bagged TBL
- Bagged MaxEnt
- Bagged Trigram
- Weighted
- Voting
- Random

% Error Reduction

<table>
<thead>
<tr>
<th>Systems</th>
<th>Random 10K</th>
<th>Voting 10K</th>
<th>Weighted 10K</th>
<th>Bagged Trigram 10K</th>
<th>Bagged MaxEnt 10K</th>
<th>Bagged TBL 10K</th>
<th>Combined Bagged over Voting 10K</th>
<th>Combined Bagged over Average 10K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>-1.34</td>
<td>20.74</td>
<td>21.67</td>
<td>-2.78</td>
<td>6.13</td>
<td>6.90</td>
<td>6.67</td>
<td>26.03</td>
</tr>
<tr>
<td>5K</td>
<td>-0.55</td>
<td>22.06</td>
<td>22.97</td>
<td>-3.97</td>
<td>8.68</td>
<td>5.55</td>
<td>7.45</td>
<td>27.86</td>
</tr>
<tr>
<td>1K</td>
<td>-0.30</td>
<td>26.95</td>
<td>29.55</td>
<td>-2.22</td>
<td>14.47</td>
<td>6.03</td>
<td>12.63</td>
<td>36.17</td>
</tr>
</tbody>
</table>
Overall Results

System Type

Accuracy

Training Data Set

Baselines
Trigram
MaxEnt
TBL
Average
Bagged Systems
Trigram
MaxEnt
TBL
Combinations
Random
Voting
Weighted
Combined Bagged

Accuracy:
82
84
86
88
90
92
94
96
98

Training Data Set:
1K
5K
10K
40K
# Baseline Data

<table>
<thead>
<tr>
<th>Method</th>
<th>1K</th>
<th>5K</th>
<th>10K</th>
<th>40K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram</td>
<td>85.680</td>
<td>92.116</td>
<td>93.438</td>
<td>95.359</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>88.304</td>
<td>93.548</td>
<td>94.629</td>
<td>96.342</td>
</tr>
<tr>
<td>TBL</td>
<td>91.639</td>
<td>94.566</td>
<td>95.225</td>
<td>96.349</td>
</tr>
</tbody>
</table>
Data Trends

• Increase in accuracy as the size of the training data increases
• Not a linear function
• Perhaps the change is proportional to the relative change in the size of the training data.
Accuracy vs. Log (Training Size)
Error vs. Log (Training Size)
Log (Error) vs. Log (Training Size)
Does This Make Sense?

• We consider the proportional increase in the size of the training data:
  \[ \log \text{(training data size)} \]

• As that increases, for example, as it doubles we see a proportional decrease in the percentage error:
  \[ \log (100 - \text{accuracy}) \]
Does This Trend Continue?

- Some additional data points
- Additional training data:
  100, 250, 500, 2K, 4K, 7K, 20K.
Log (Error) vs. Log (Training Size)
Some Analysis

• Linear Least Squares Regression
• Y-Intercepts:
  ▪ Trigram: 2.1348
  ▪ MaxEnt: 2.1647
  ▪ TBL: 1.7467
• Slope Values:
  ▪ Trigram: – 0.32732
  ▪ MaxEnt: – 0.35931
  ▪ TBL: – 0.26656
• Trigram: 0.995885
• MaxEnt: 0.995944
• TBL: 0.992777
• !!!
Potential Interpretations

• Y-Intercepts:
  ▪ Trigram: 2.1348
  ▪ MaxEnt: 2.1647
  ▪ TBL: 1.7467

• Slope Values:
  ▪ Trigram: −0.32732
  ▪ MaxEnt: −0.35931
  ▪ TBL: −0.26656

• Y-Intercept Meaning
  ▪ Potentially a measure of “robustness”

• Slope Meaning
  ▪ Potentially a measure of “trainability”
  ▪ Responsiveness of the ML algorithm to data size
  ▪ Coefficient of Training Efficiency?
Caveats

• It may be we are in a “sweet spot” for these algorithms.

• However, this relationship does seem to hold for a broad range of practical values for training data sizes:
  
  100 sentences – 40K sentences
Conclusions

- Bagging is effective for some algorithms and not others.
- System combination is an moderately effective way to maximize accuracy, especially if the ML algorithms involved model the data in different ways.
- Bagging and then combining systems is a good way to maximize accuracy but has major runtime drawbacks, such as computation time and system complexity.
- Doing this experiment allowed us to get some interesting quantitative comparison measures of three common ML algorithms. It is hard to say if these measures are generalizable.