Bagging & System Combination for POS Tagging

Dan Jinguji
Joshua T. Minor
Ping Yu

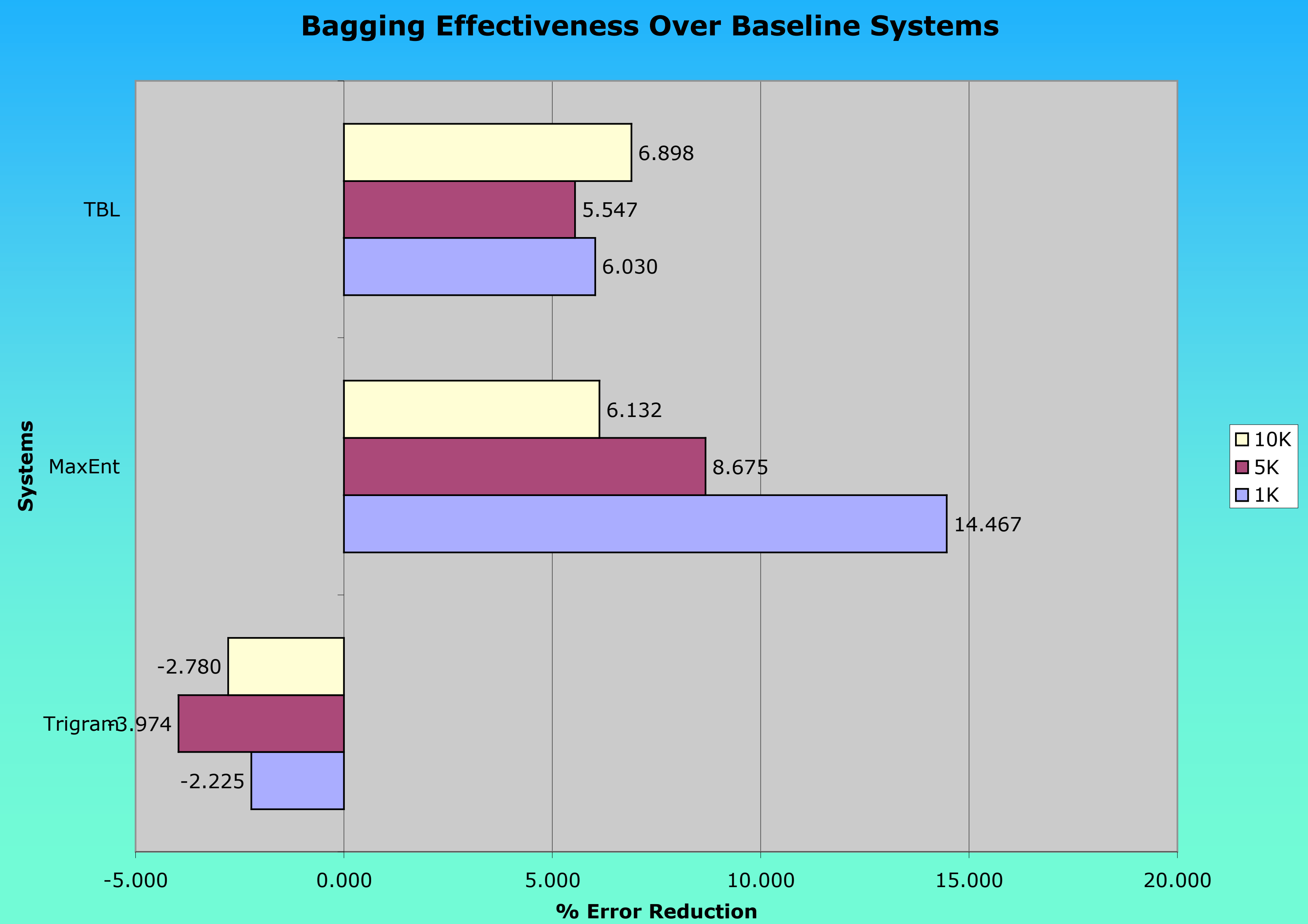
Bagging

- 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



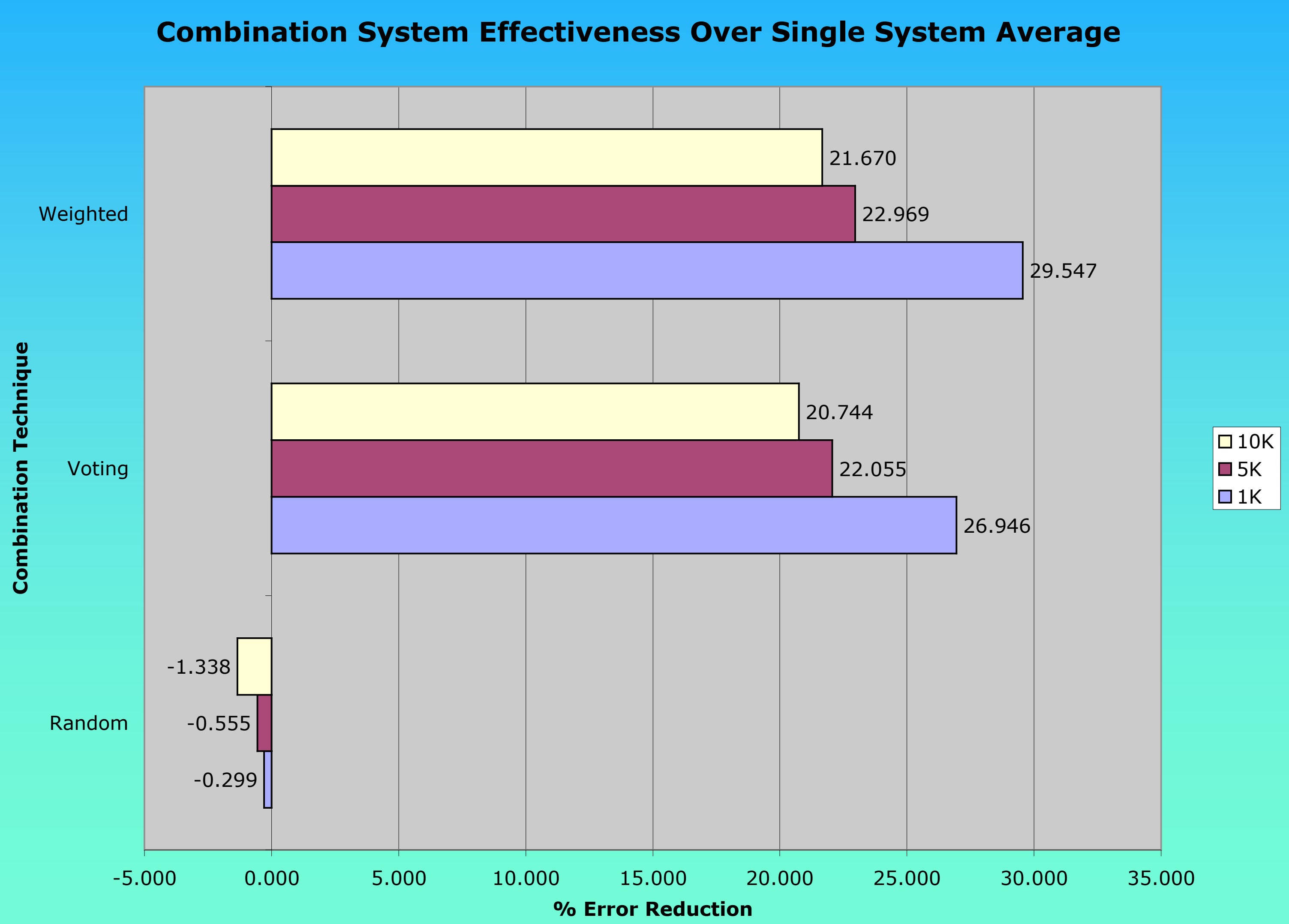
Bagging and Stability

- 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

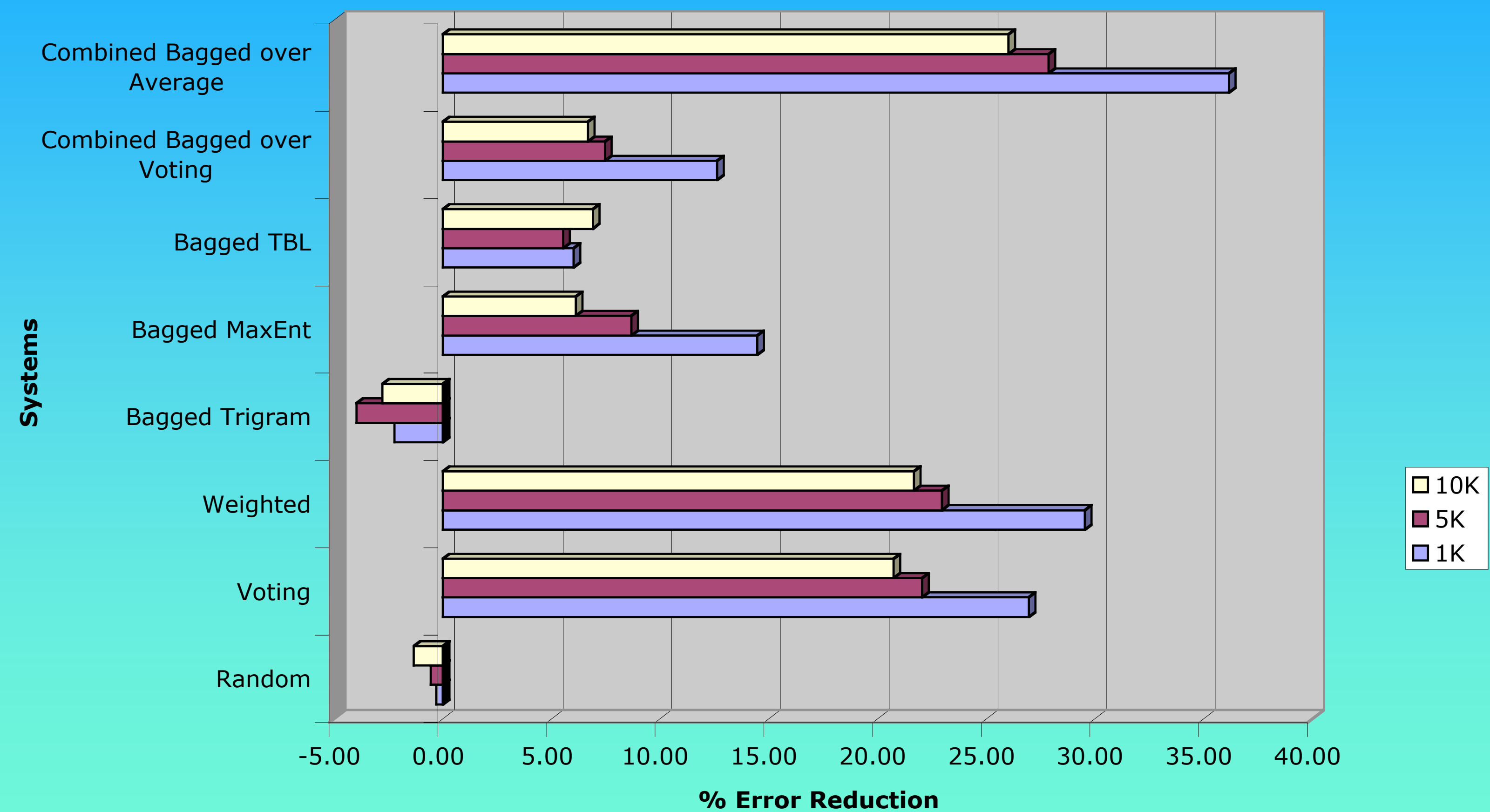


Hybrid Bagging

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

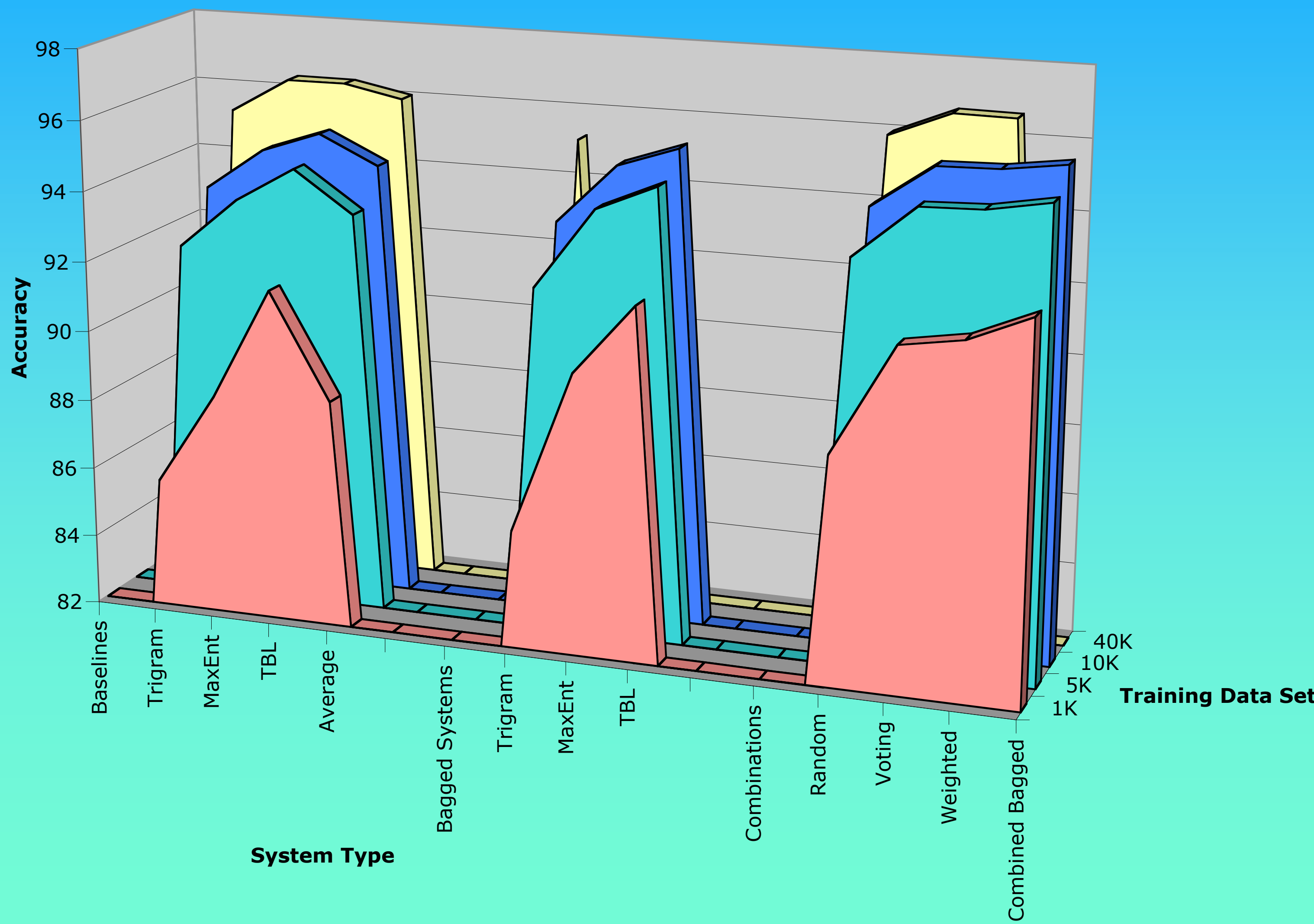
	1K	5K	10K
Accuracy	92.6568	95.2401	95.8752
Error Reduction Over Voting	12.63	7.45	6.67
Error Reduction Over Average	36.17!	27.86	26.03

Overall Improvement Comparison



	Random	Voting	Weighted	Bagged Trigram	Bagged MaxEnt	Bagged TBL	Combined Bagged over Voting	Combined Bagged over Average
10K	-1.34	20.74	21.67	-2.78	6.13	6.90	6.67	26.03
5K	-0.55	22.06	22.97	-3.97	8.68	5.55	7.45	27.86
1K	-0.30	26.95	29.55	-2.22	14.47	6.03	12.63	36.17

Overall Results



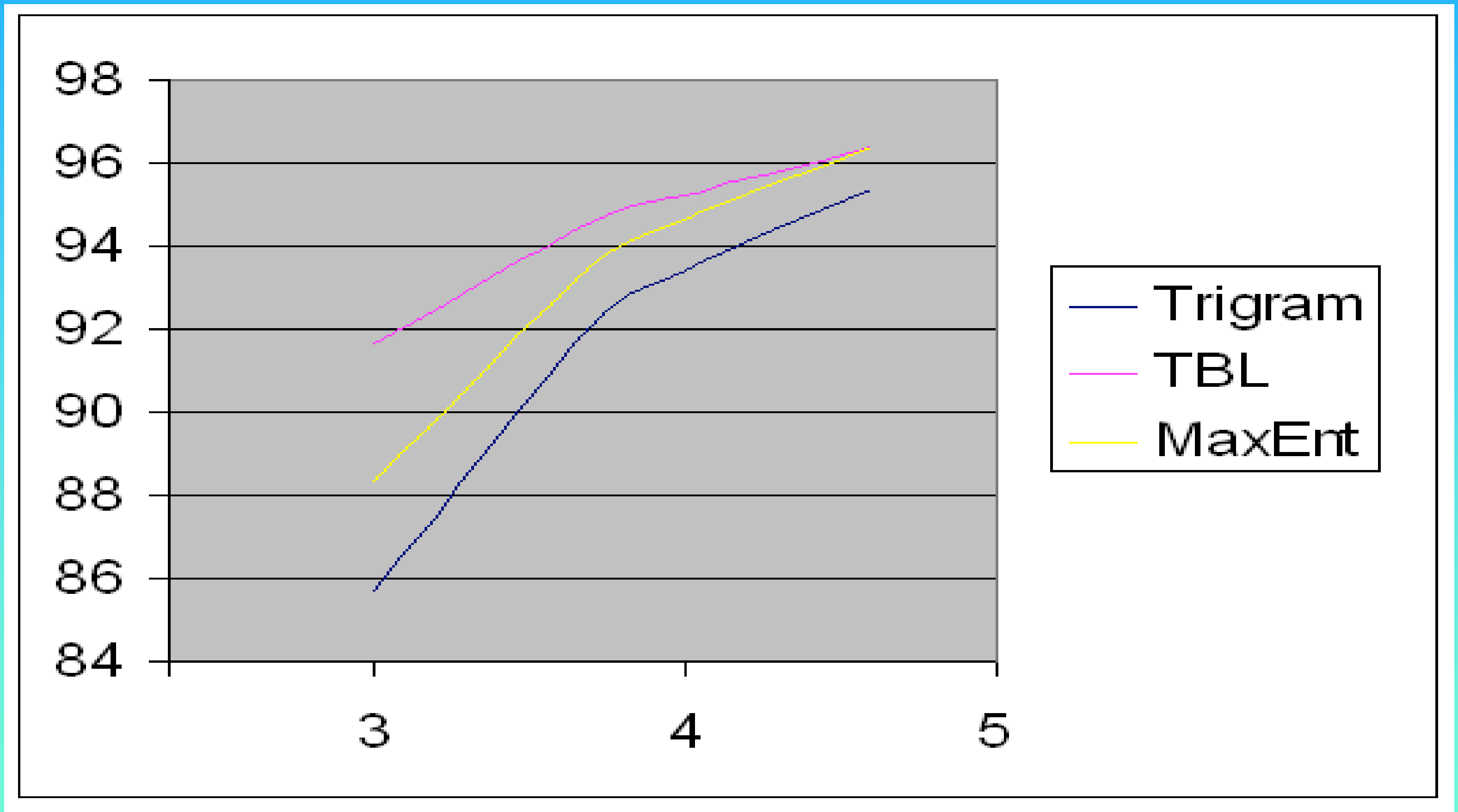
Baseline Data

	1K	5K	10K	40K
Trigram	85.680	92.116	93.438	95.359
MaxEnt	88.304	93.548	94.629	96.342
TBL	91.639	94.566	95.225	96.349

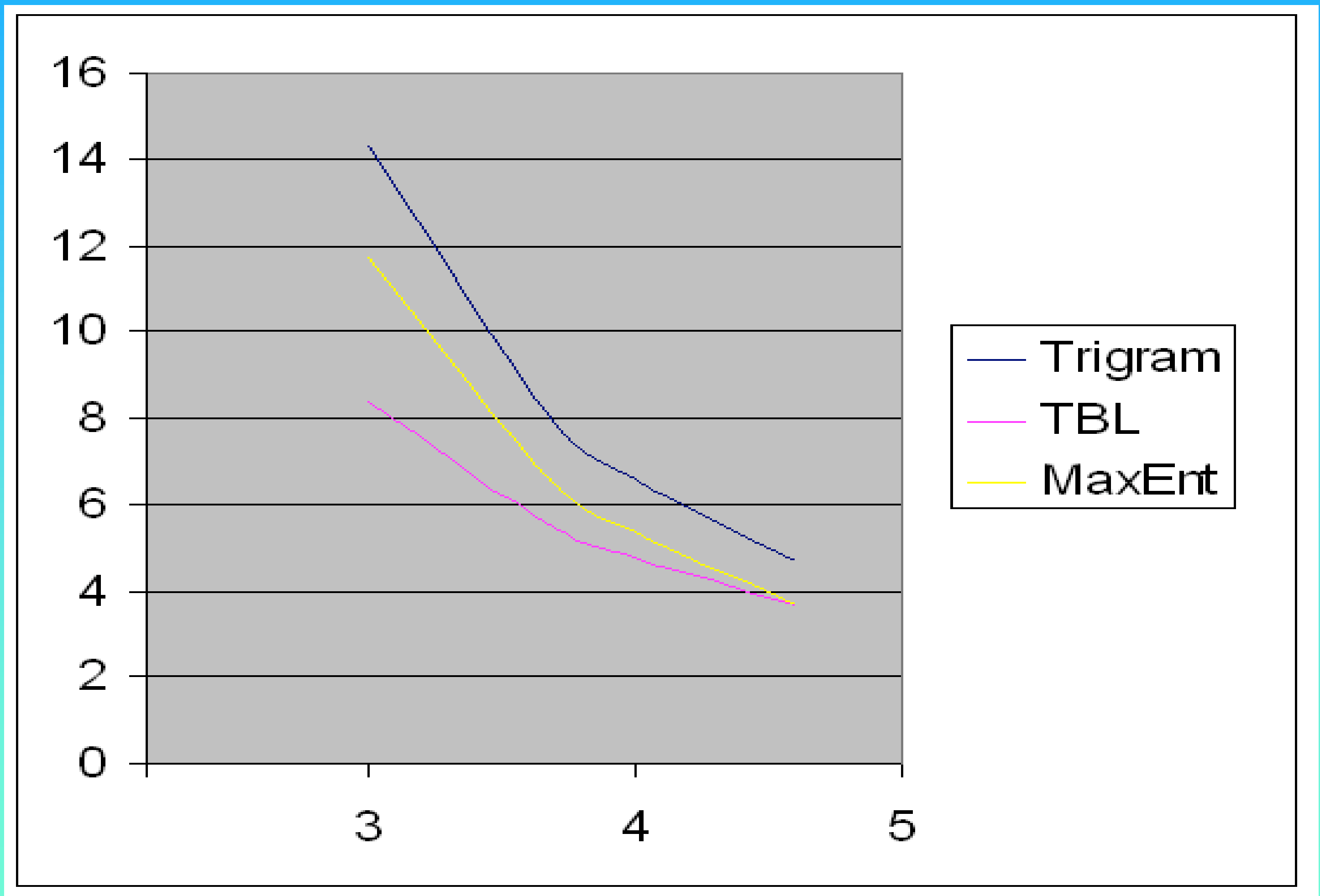
Data Trends

- 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.

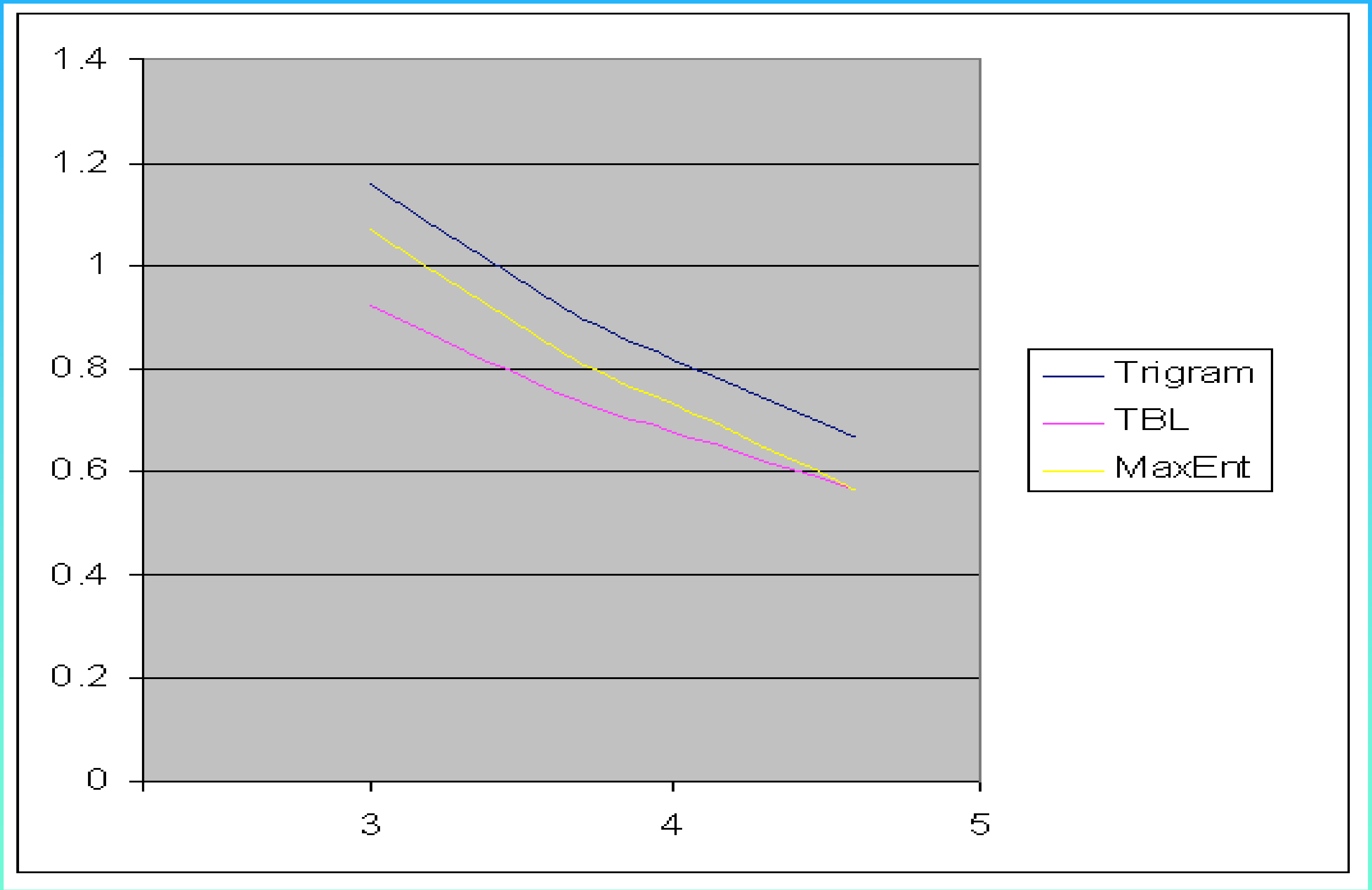
Accuracy vs. Log (Training Size)



Error vs. Log (Training Size)



Log (Error) vs. Log (Training Size)



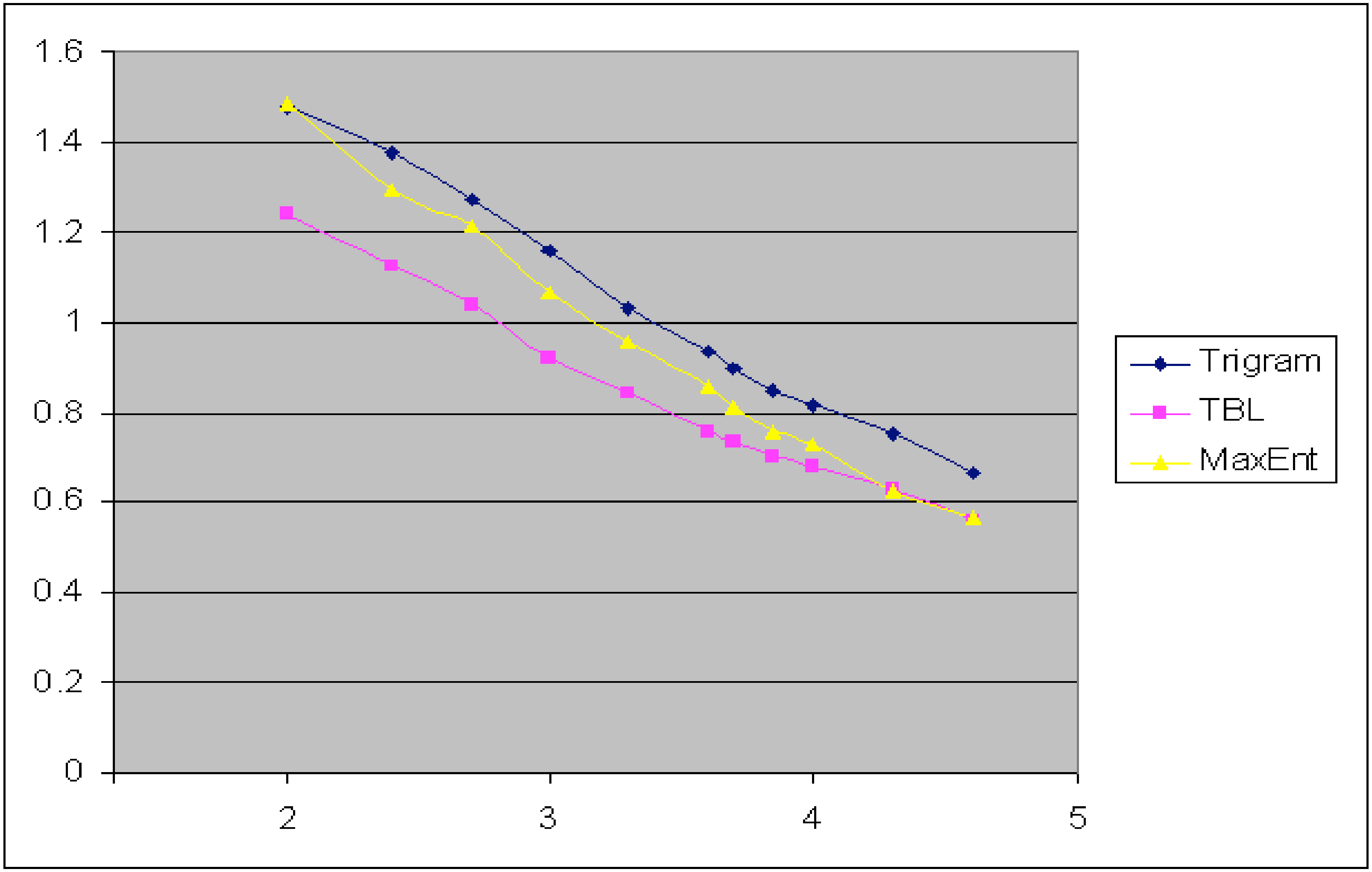
Does This Make Sense?

- We consider the proportional increase in the size of the training data:
  log (training data size)
- As that increases, for example, as it doubles we see a proportional decrease in the percentage error:
  log (100 – 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

Correlation Coefficients

• 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.