Vowpal Wabbit

http://hunch.net/~vw/

git clone
git://github.com/JohnLangford/vowpal_wabbit.git

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is:
   1 | The dog ate my homework

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is: 1 | The dog ate my homework
3. Allreduce. Terascale learning paper = most scalable public algorithm.

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is:
   1 | The dog ate my homework
3. Allreduce. Terascale learning paper = most scalable public algorithm.
4. Reductions. Solve wide variety of problems well by reduction to simple problems.

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is: 1 | The dog ate my homework
3. Allreduce. Terascale learning paper = most scalable public algorithm.
4. Reductions. Solve wide variety of problems well by reduction to simple problems.
5. Interactive. Causation instead of correlation. Learn to control based on feedback.

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is:
   1 | The dog ate my homework
3. Allreduce. Terascale learning paper = most scalable public algorithm.
4. Reductions. Solve wide variety of problems well by reduction to simple problems.
5. Interactive. Causation instead of correlation. Learn to control based on feedback.
6. Learn2Search. See Hal.

Vowpal Wabbit: Research Delivery Vehicle

1. Online by default. A good solution to an ML problem is always an hour or less away.
2. Hashing. Raw text is fine. A valid input is:
   1 | The dog ate my homework
3. Allreduce. Terascale learning paper = most scalable public algorithm.
4. Reductions. Solve wide variety of problems well by reduction to simple problems.
5. Interactive. Causation instead of correlation. Learn to control based on feedback.
6. Learn2Search. See Hal.
7. others....

A user base becomes addictive

1. Mailing list of about 400

A user base becomes addictive

1. Mailing list of about 400
2.

amazon AMERICAN EXPRESS AOL Baidu 百度 facebook FTI CONSULTING GraphLab IBM Microsoft YAHOO! Яндекс

An example

`vw -c rcv1.train.raw.txt -b 22 --ngram 2 --skips 4 -l 0.25 --binary` provides stellar performance in 12 seconds.

Learning Reductions

The core idea: reduce complex problem A to simpler problem B then use solution on B to get solution on A.

Problems:

1. How do you make it efficient enough?
2. How do you make it natural to program?

The Reductions Interface

```
void learn(learner& base, example* ec)
{
   base.learn(ec); // The recursive call
   if ( ec->final_prediction > 0) //Thresholding
      ec->final_prediction = 1;
   else
      ec->final_prediction = -1;
   label_data* ld = (label_data*)ec->ld;//New loss
   if (ld->label == ec->final_prediction)
      ec->loss = 0.;
   else
      ec->loss = 1.;
}
```

Structured Prediction

= joint prediction with a joint loss

Structured Prediction

= joint prediction with a joint loss

Example: Part of Speech Tagging

Pierre	Vinken	,	61	years	old
Proper N.	Proper N.	Comma	Number	Noun	Adj.

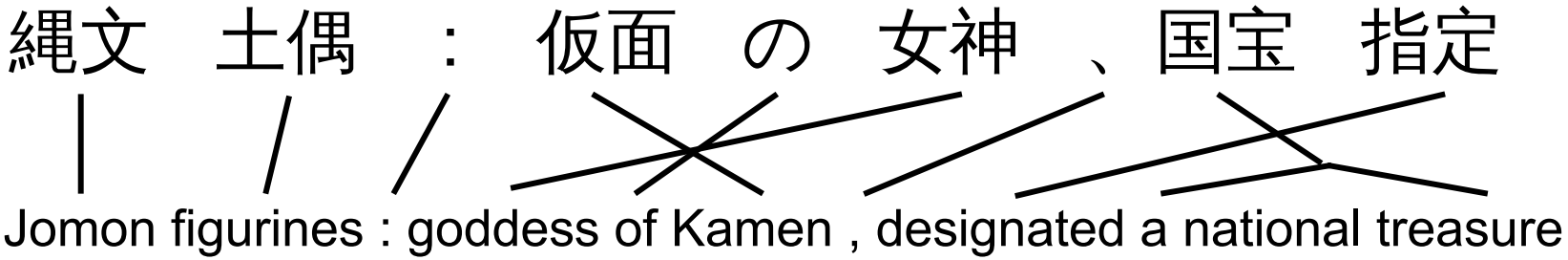
Structured Prediction

= joint prediction with a joint loss

Example: Part of Speech Tagging

Pierre	Vinken	,	61	years	old
Proper N.	Proper N.	Comma	Number	Noun	Adj.

Example 2: Machine Translation



How can you best do structured prediction?

We care about:

1. Programming complexity.

How can you best do structured prediction?

We care about:

1. Programming complexity. Most structured predictions are not addressed with structured learnign algorithms, because it it too complex to do so.

How can you best do structured prediction?

We care about:

1. Programming complexity. Most structured predictions are <u>not</u> addressed with structured learnign algorithms, because it it too complex to do so.
2. Prediction accuracy. It had better work well.

How can you best do structured prediction?

We care about:

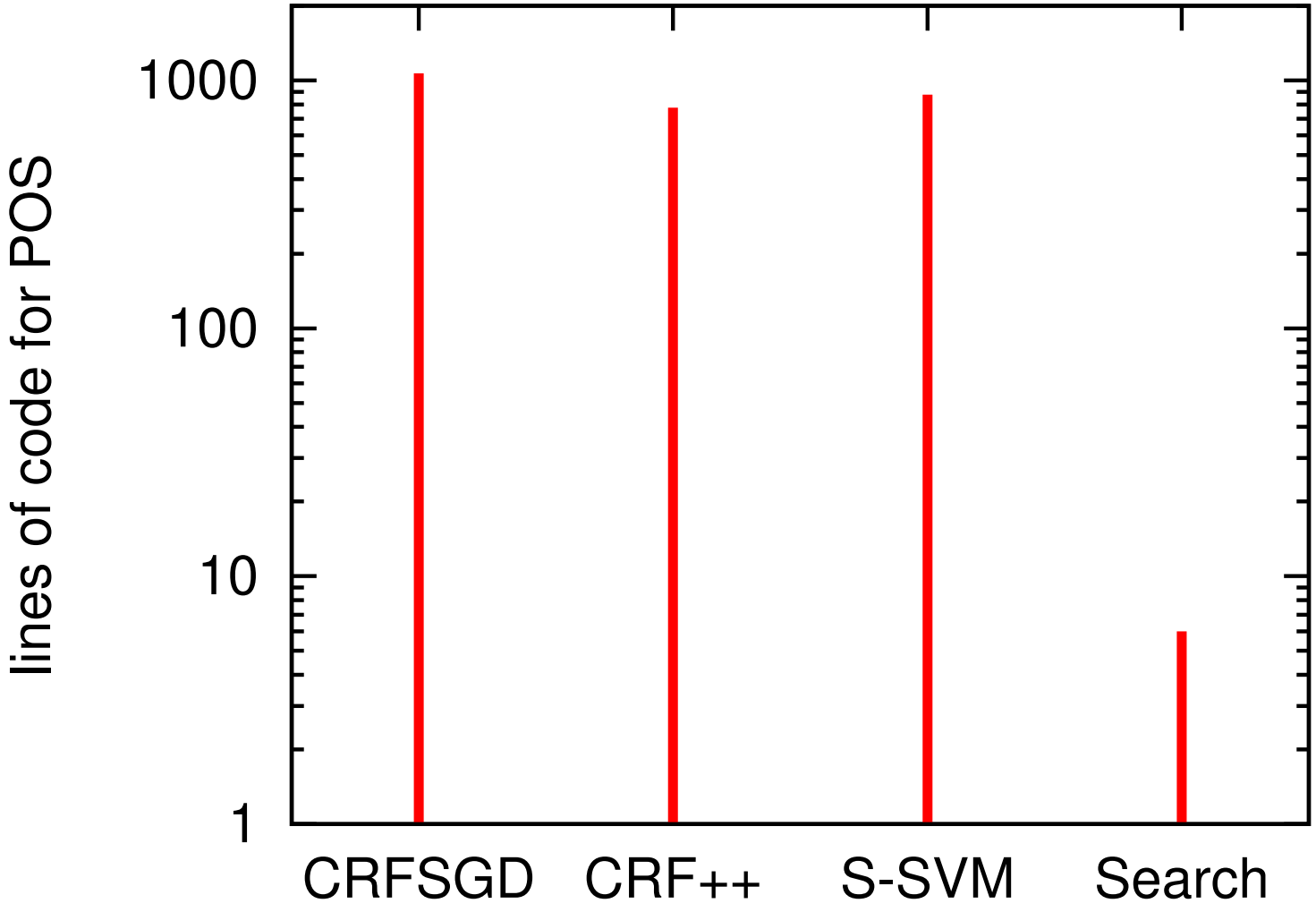
1. Programming complexity. Most structured predictions are not addressed with structured learnign algorithms, because it it too complex to do so.
2. Prediction accuracy. It had better work well.
3. Train speed. Debug/development productivity + maximum data input.

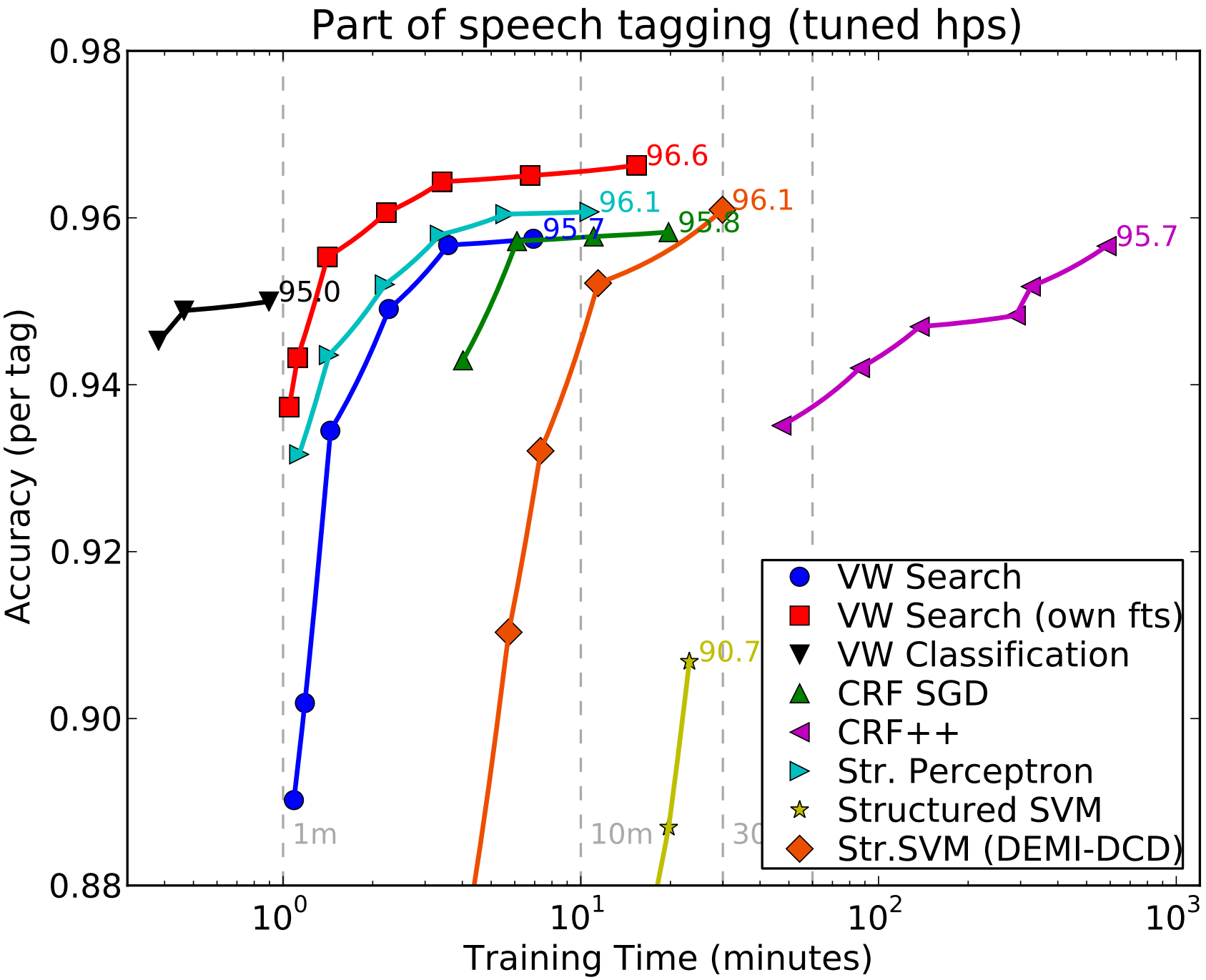
How can you best do structured prediction?

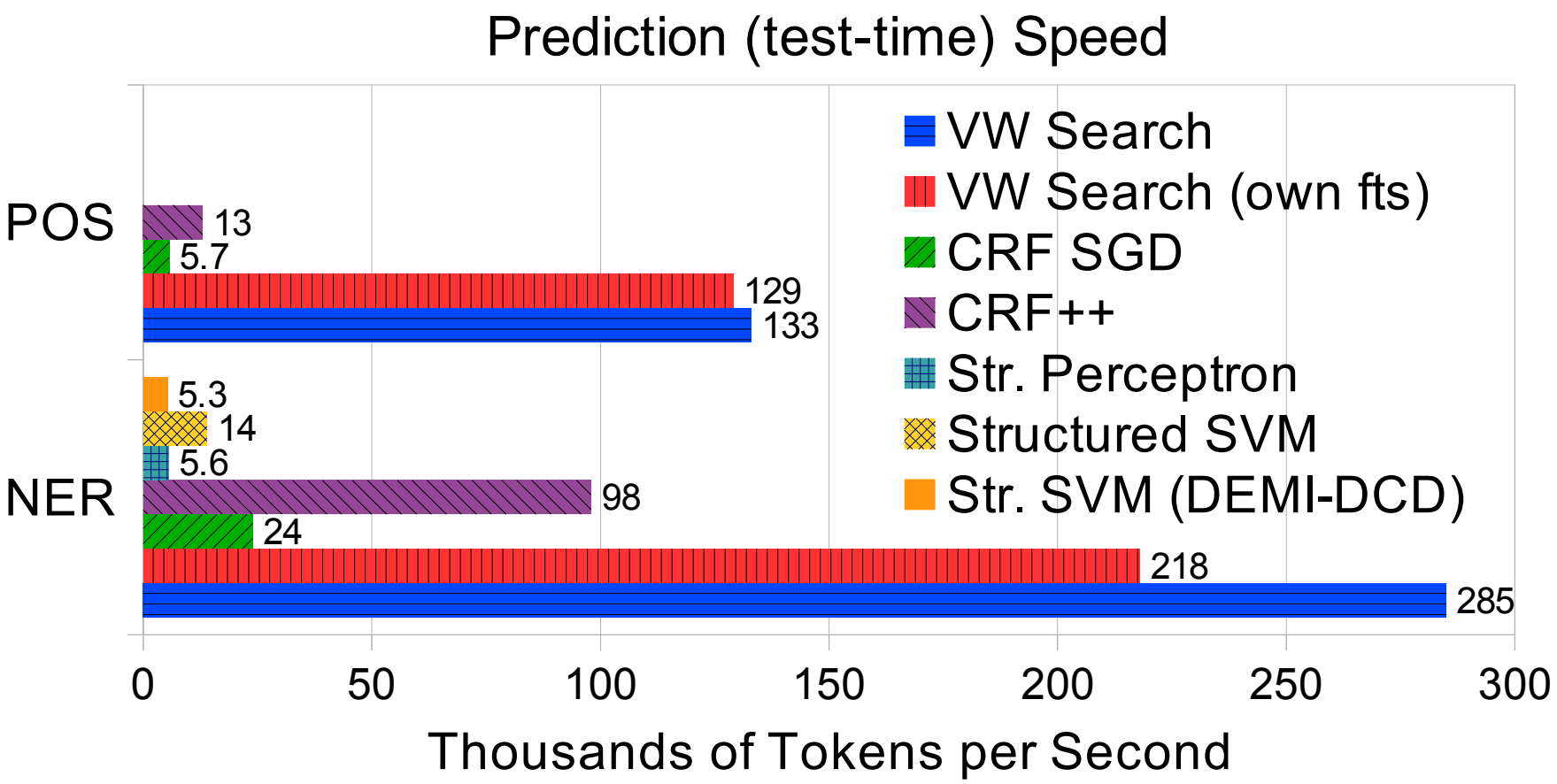
We care about:

1. Programming complexity. Most structured predictions are not addressed with structured learnign algorithms, because it it too complex to do so.
2. Prediction accuracy. It had better work well.
3. Train speed. Debug/development productivity + maximum data input.
4. Test speed. Application efficiency

A program complexity comparison




Part of speech tagging (tuned hps)
Accuracy (per tag)
Training Time (minutes)
0.98
0.96
0.94
0.92
0.90
0.88
10^0
10^1
10^2
10^3
1m
10m
96.6
96.1
95.7
95.8
96.1
95.0
95.7
90.7
VW Search
VW Search (own fts)
VW Classification
CRF SGD
CRF++
Str. Perceptron
Structured SVM
Str.SVM (DEMI-DCD)


Prediction (test-time) Speed
POS
NER
13
5.7
129
133
5.3
14
5.6
98
24
218
285
VW Search
VW Search (own fts)
CRF SGD
CRF++
Str. Perceptron
Structured SVM
Str. SVM (DEMI-DCD)
0
50
100
150
200
250
300
Thousands of Tokens per Second

An example

```
vw -b 24 -d wsj.train.vw -c –search_task sequence –search 45
–search_alpha 1e-8 –search_neighbor_features -1:w,1:w
–affix -1w,+1w
```

Good performance in about 2.5 minutes.