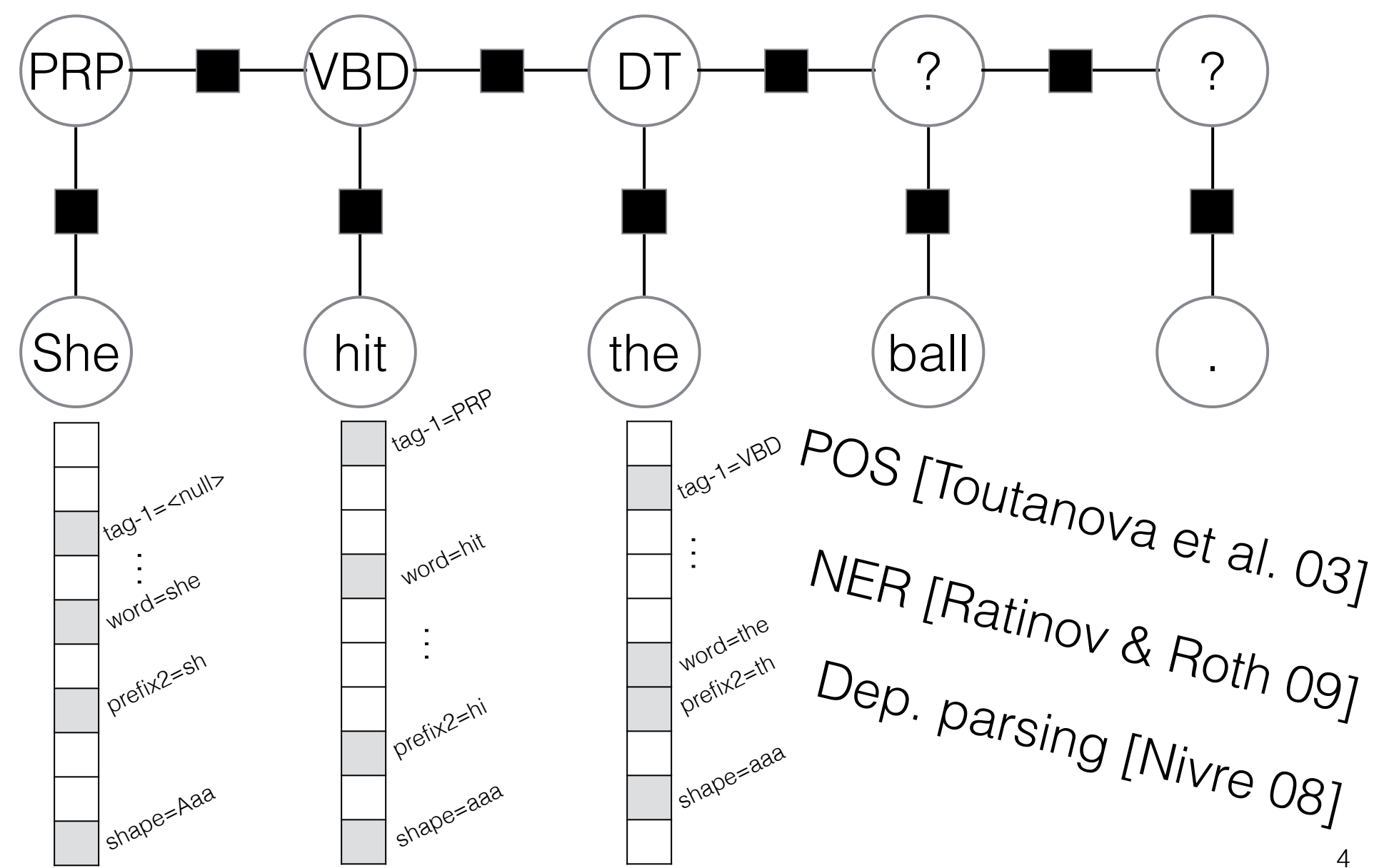


Learning Dynamic Feature Selection for Fast Sequential Prediction

Emma Strubell, Luke Vilnis, Kate Silverstein & Andrew McCallum
 College of Information and Computer Sciences, University of Massachusetts Amherst

Problem



- Want fast and accurate NLP
- In many cases, fewer features needed for accurate prediction

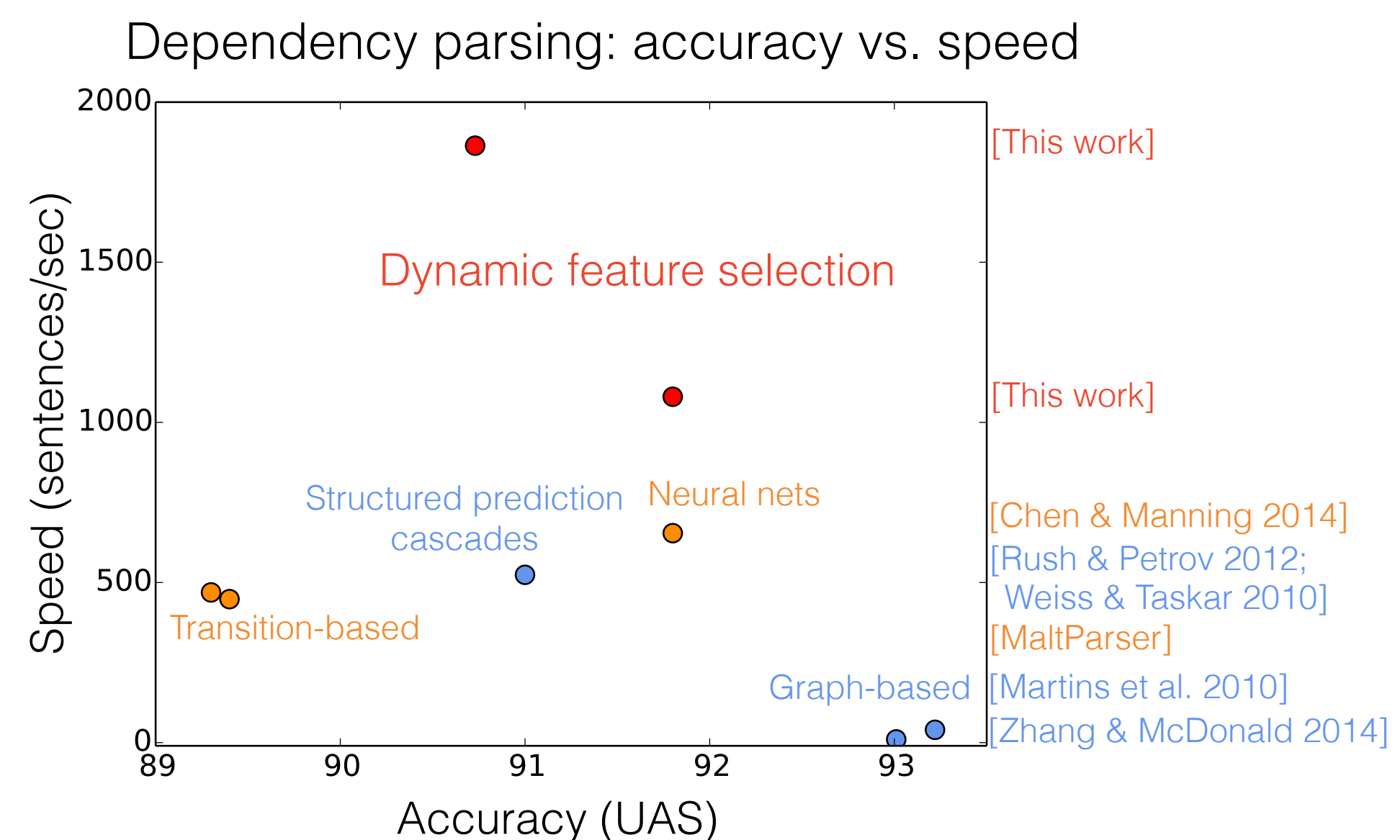
Solution

Define a linear model over *feature templates* $\{\Phi_j(x, y)\}$:

$$y^* = \arg \max_{y \in \mathcal{Y}} \mathbf{w} \cdot \Phi(x, y) \quad (1)$$

$$\mathbf{w} \cdot \Phi(x, y) = \sum_j \mathbf{w}_j \cdot \Phi_j(x, y) \quad (2)$$

Approximate (1) using as few terms as possible from (2).



Dynamic feature selection

```
def predict(token) :=
  for(template <- feature_templates)
  feature_index := compute_feature(token, template)
  scores += weights[feature_index]
  if(confident)
    return argmax(scores)
```

	JJ	NN	CC	VBN	...	DT	NNP	PRP	VBD	
word@0	-0.23	1.69	0.61	-0.48	1.44	0.36	-0.73	1.19	0.11	-0.98
1 word@0=the	0.01	1.92	0.84	-0.25	1.67	0.59	-0.5	1.42	0.34	-0.75
tag@-1	-0.84	1.08	0	1.92	0.84	-0.25	1.67	0.59	-0.5	1.42
1 tag@-1=VBD	-0.99	0.93	-0.15	1.77	0.69	-0.4	1.52	0.44	-0.65	1.27
shape@0	-0.12	1.8	0.71	-0.38	1.54	0.46	9.75	-0.34	1.58	0.5
1 shape@0=aaa	-1.8	0.12	-0.96	0.96	-0.13	1.79	0.71	-0.38	1.54	0.46
suffix@+1	-0.48	1.44	0.36	-0.73	1.19	0.11	-0.98	0.94	-0.15	1.77
1 suffix@+1=all	-7.61	0.3	-0.79	1.13	0.04	1.96	0.88	-0.21	1.71	0.63
	1.1	0.02	1.94	0.86	-0.23	1.69	0.61	-0.48	1.44	0.36
	-0.48	1.44	0.36	-0.62	1.3	0.21	-0.88	1.04	-0.04	1.88
	0.2	-0.88	1.04	-0.04	1.88	0.8	-0.29	1.63	0.54	-0.54
	1.94	0.86	-0.23	1.69	0.61	-0.48	1.44	0.36	-0.73	1.19
scores	-1.92	5.76	1.43	0.19	4.87	0.53	9.56	2.23	0.89	5.57

Learning

Consider model predictions $P_{i,y}$ for each template prefix (Eq. 2).
 Hinge loss with margin m on prefix score i :

$$h(P_i, y) = \max\{0, \max_{y' \neq y} P_{i,y'} - P_{i,y} + m\} \quad (2)$$

Per-example gradient:

$$\frac{\partial \ell}{\partial \mathbf{w}_j} = \sum_{k=j}^{i_y^*} \Phi_j(x, y_{\text{loss}}(P_k, y)) - \Phi_j(x, y)$$

where $i_y^* = \min_{i \in \{1..k\}} i$ s.t. $h(P_i, y) = 0$
 and $y_{\text{loss}}(P_i, y) = \arg \max_{y'} P_{i,y'} - m \cdot \mathbb{1}(y' = y)$

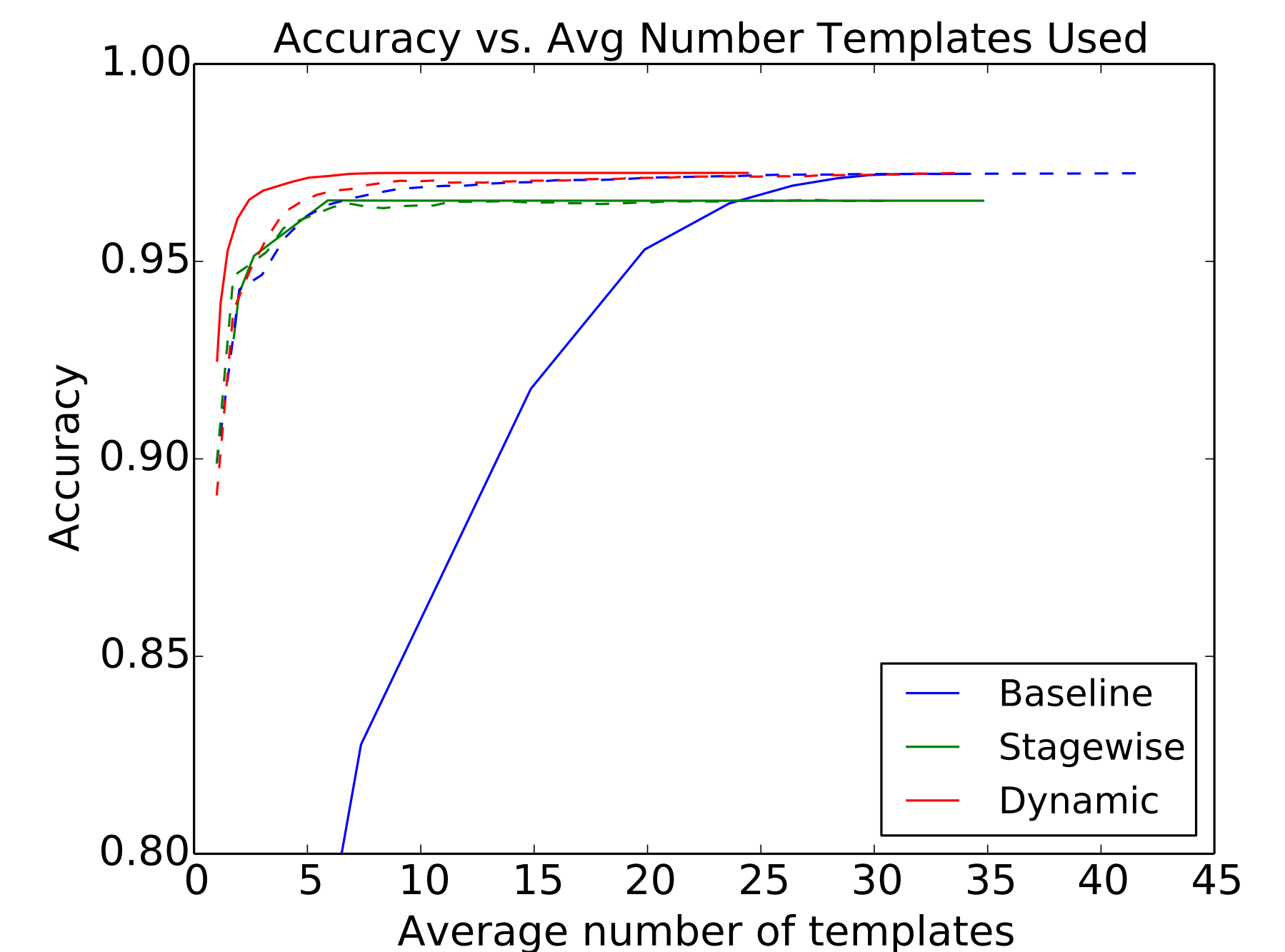
Inference

- Test time, compute prefix scores until any label has margin m .
- Train time, compute prefix scores until correct with margin m .

Experimental results

Part-of-speech tagging:

Model	Accuracy	Templates	Speedup
baseline	97.22	46	1x
dynamic conservative	97.21	6.89	3.41x
dynamic aggressive	97.02	4.33	5.22x
dynamic v. aggressive	96.09	1.92	10.36x



Transition-based dependency parsing:

Model	LAS	UAS	Templates	Speedup
baseline	90.31	91.83	60	1x
dynamic conservative	90.27	91.80	15.83	2.71x
dynamic aggressive	89.07	90.73	8.57	4.66x