Automatic Domain Adaptation for Parsing

David McClosky[a,b] Eugene Charniak[b] Mark Johnson[c,b]

[a] Natural Language Processing Group
Stanford University

[b] Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University

[c] Department of Computing
Macquarie University

(work performed while all authors were at Brown)

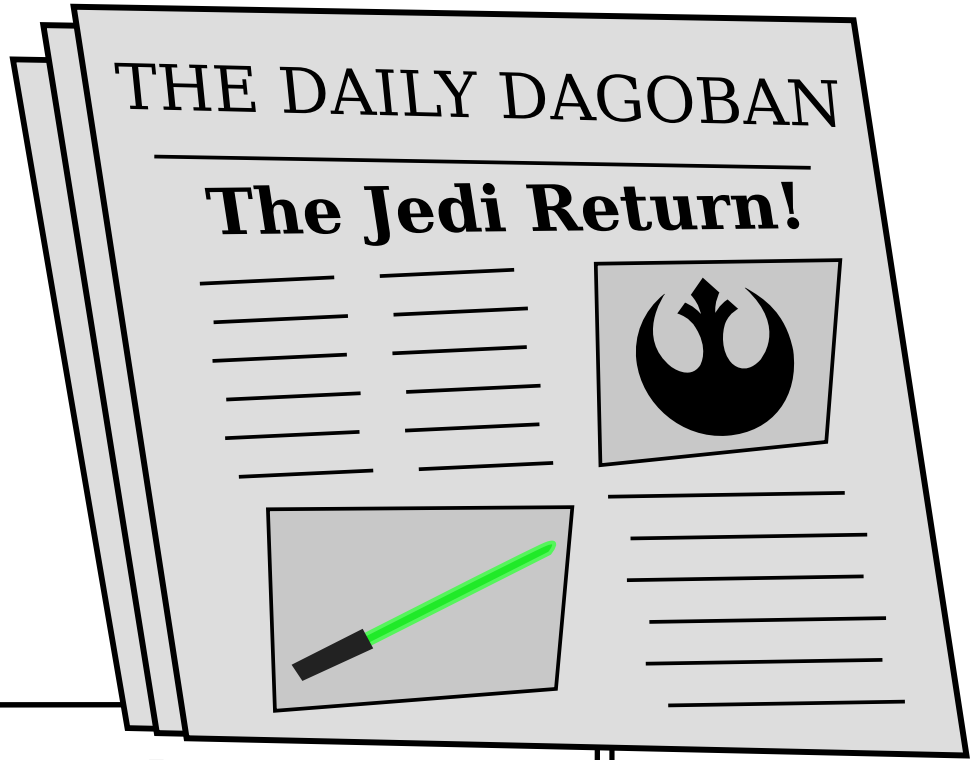
NAACL-HLT 2010 — June 2nd, 2010

Understanding language

[Lucas et al., 1977, Lucas et al., 1980, Lucas et al. 1983]

Keeping up to date with Twitter

Reading the news



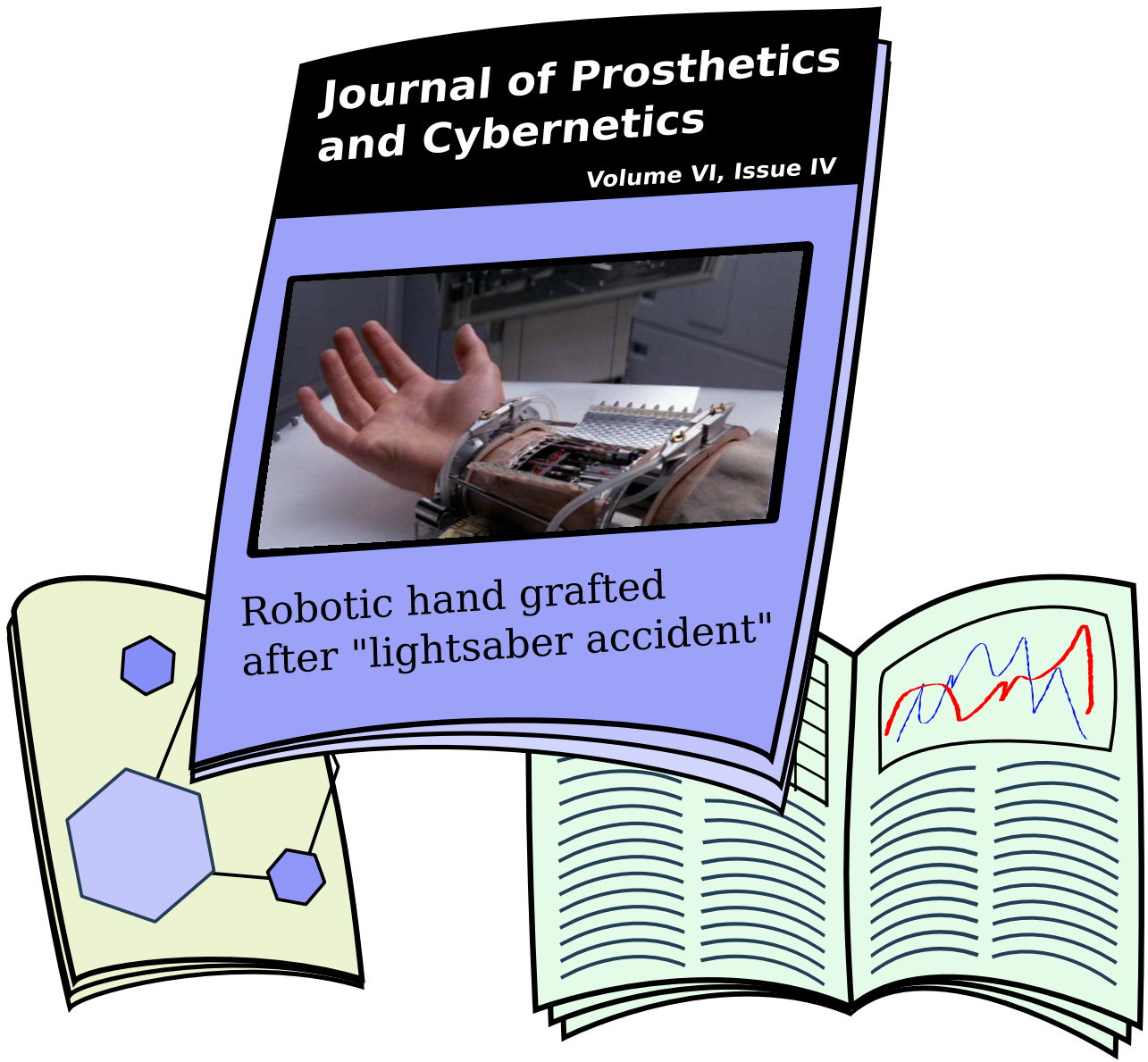
RAP **Rebel Alliance Associated Press**

Rebels make Death Star go Nova!

by Hon. Princess Leia (RAP writer), Hans Solo (RAP contributor)

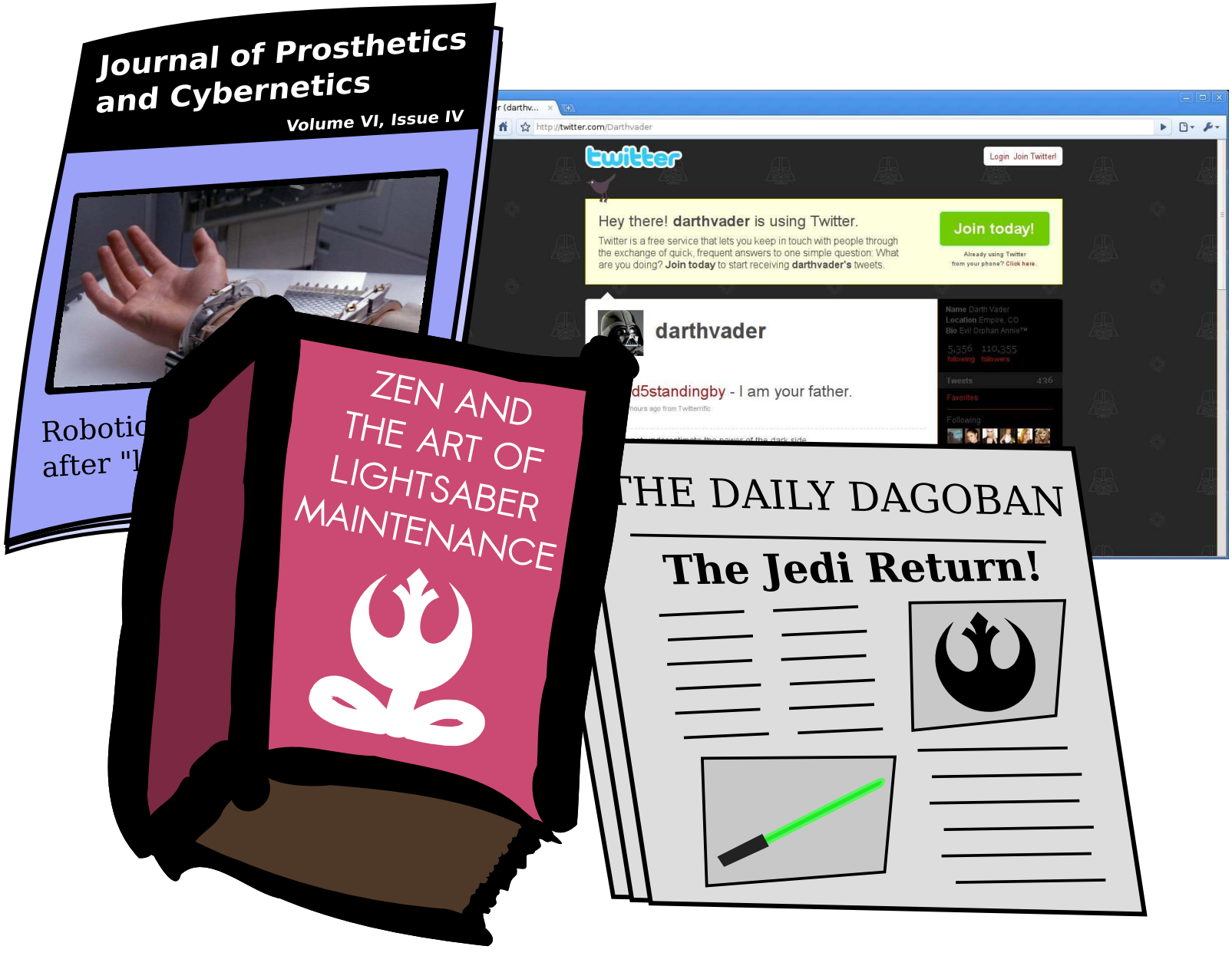
DEATH STAR -- At 3:22pm Galatic Central Time, Rebel fighters launched an assault which ultimately lead to the destruction of

Studying the latest medical journals



Casual reading

What's in a domain?



Crossdomain parsing performance

	Test
Train	WSJ
WSJ	89.7

(*f*-scores on all sentences in test sets, Charniak parser)

Crossdomain parsing performance

	Test				
Train	BROWN	GENIA	SWBD	ETT	WSJ
BROWN	86.7				
GENIA		84.6			
SWBD			88.2		
ETT				82.4	
WSJ					89.7

(*f*-scores on all sentences in test sets, Charniak parser)

Crossdomain parsing performance...not great

	Test				
Train	BROWN	GENIA	SWBD	ETT	WSJ
BROWN	86.7	73.5	77.6	80.8	79.9
GENIA	65.7	84.6	50.5	67.1	64.6
SWBD	75.8	63.6	88.2	76.2	69.8
ETT	76.2	65.7	74.5	82.4	72.6
WSJ	84.1	76.2	76.7	82.2	89.7

(*f*-scores on all sentences in test sets, Charniak parser)

Color key: $<$ 70, 70–80, $>$ 80

Automatic Domain Adaptation for Parsing

- What if we don’t know the target domain?

Automatic Domain Adaptation for Parsing

- What if we don’t know the target domain?
  - Parsing the web or any other large heterogeneous corpus

Automatic Domain Adaptation for Parsing

- What if we don’t know the target domain?
  - Parsing the web or any other large heterogeneous corpus
- A new ~~hope~~ parsing task:

Automatic Domain Adaptation for Parsing

- What if we don't know the target domain?
  - Parsing the web or any other large heterogeneous corpus
- A new ~~hope~~ parsing task:
  - labeled and unlabeled corpora (source domains)

Automatic Domain Adaptation for Parsing

- What if we don't know the target domain?
  - Parsing the web or any other large heterogeneous corpus
- A new ~~hope~~ parsing task:
  - labeled and unlabeled corpora (source domains)
  - corpora to parse (target text)

Automatic Domain Adaptation for Parsing

- What if we don't know the target domain?
  - Parsing the web or any other large heterogeneous corpus
- A new ~~hope~~ parsing task:
  - labeled and unlabeled corpora (source domains)
  - corpora to parse (target text)
- Combine source domains to best parse each target text

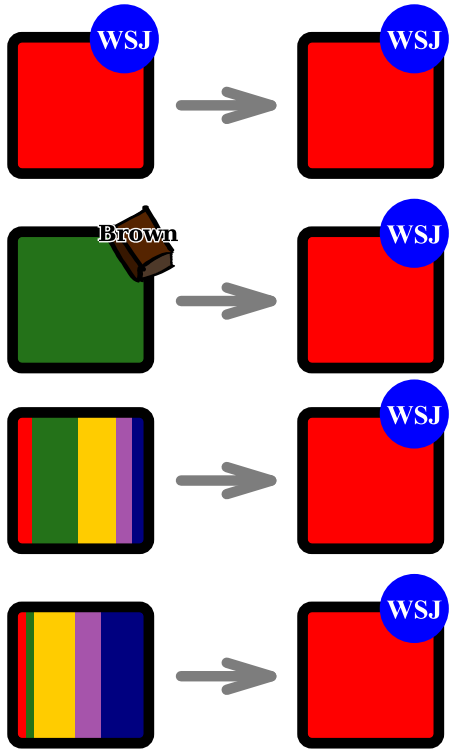
Automatic Domain Adaptation for Parsing

- What if we don't know the target domain?
  - Parsing the web or any other large heterogeneous corpus
- A new ~~hope~~ parsing task:
  - labeled and unlabeled corpora (source domains)
  - corpora to parse (target text)
- Combine source domains to best parse each target text
- Evaluation: parse unknown and foreign domains

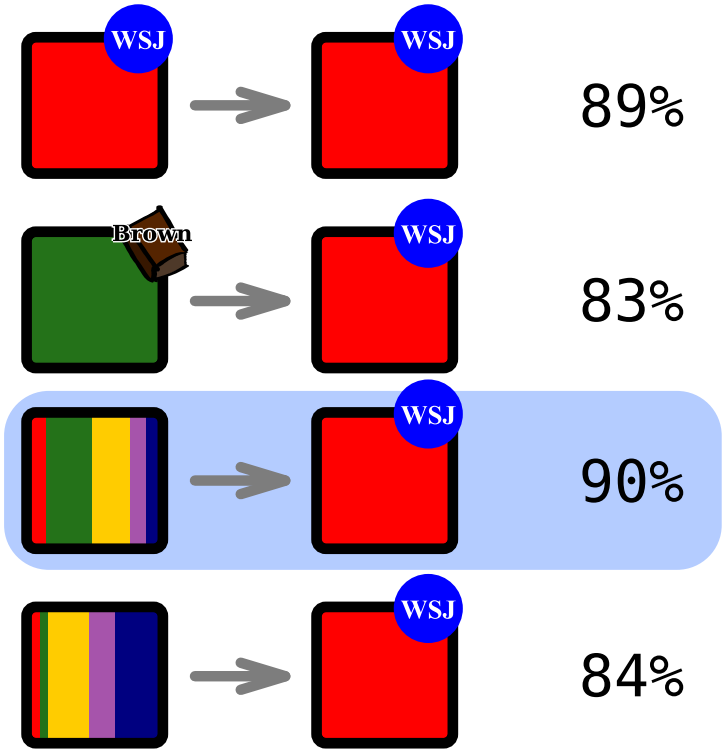
Related work

- **Subdomain Sensitive Parsing**
  [Plank and Sima'an, LREC 2008]
  - Extract subdomains from WSJ using domain-specific LMs
  - Use above to train domain-specific parsing models
- **Multitask learning**
  [Daumé III, 2007], [Finkel and Manning, 2009]
  - Each domain is a separate (related) task
  - Share non-domain specific information across domains
- **Predicting parsing performance**
  [Ravi, Knight, and Soricut, EMNLP 2008]
  - Use regression to predict *f*-score of a parse
  - Predicted accuracies can be used to rank models

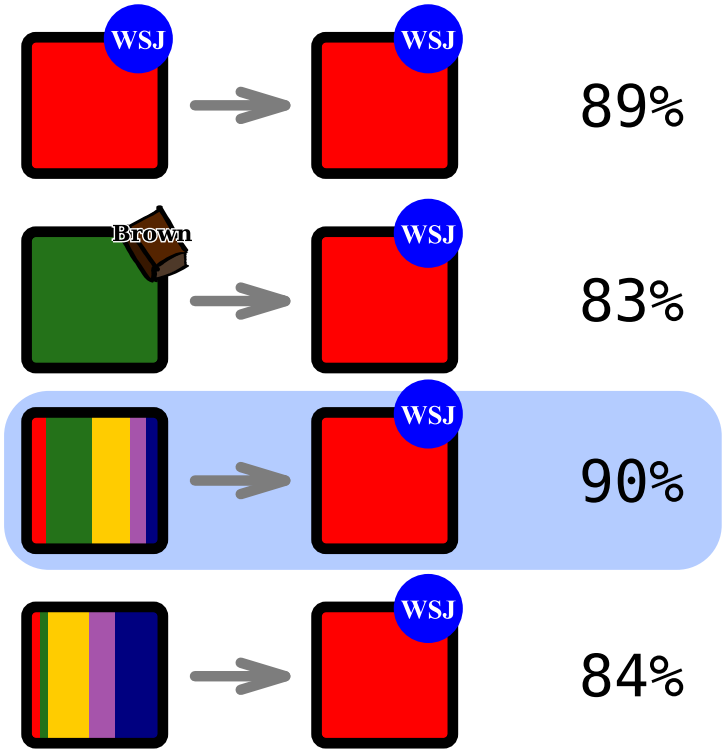
Crossdomain accuracy prediction



Crossdomain accuracy prediction

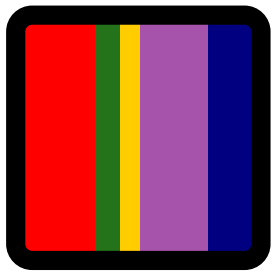


Crossdomain accuracy prediction

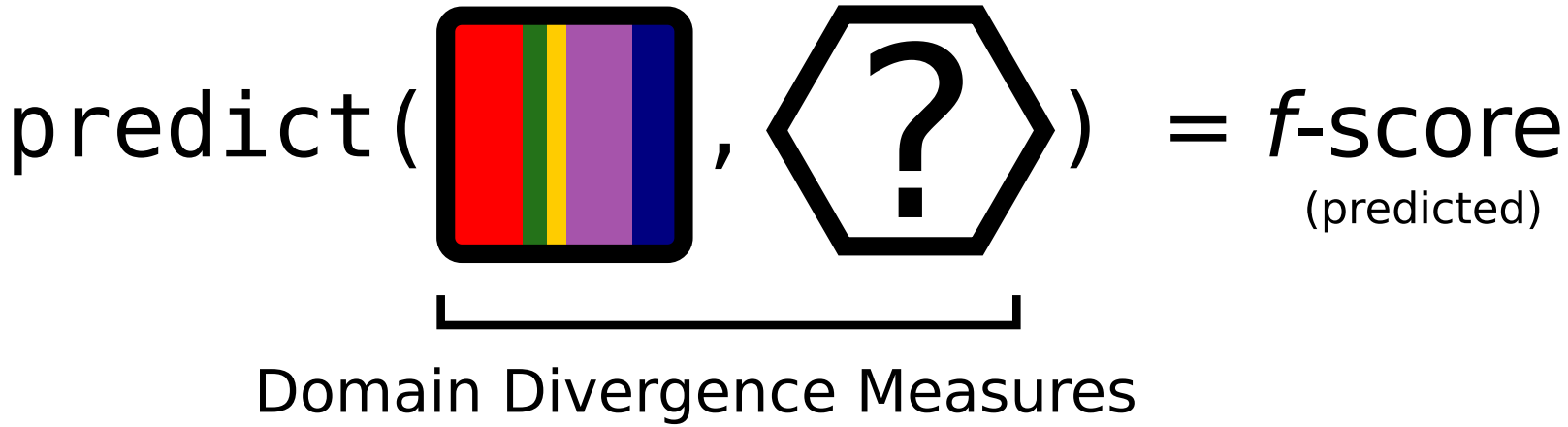


predict(, ?) = *f*-score

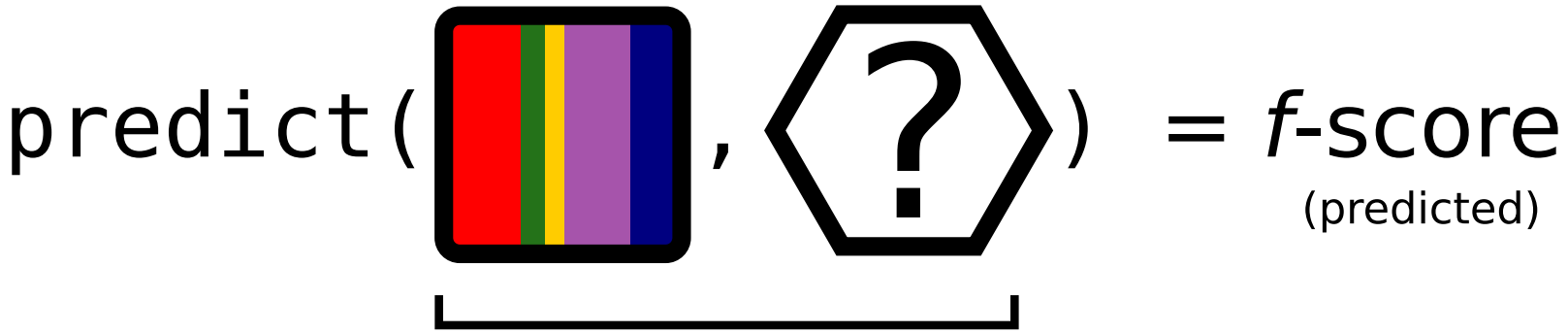
Prediction by regression

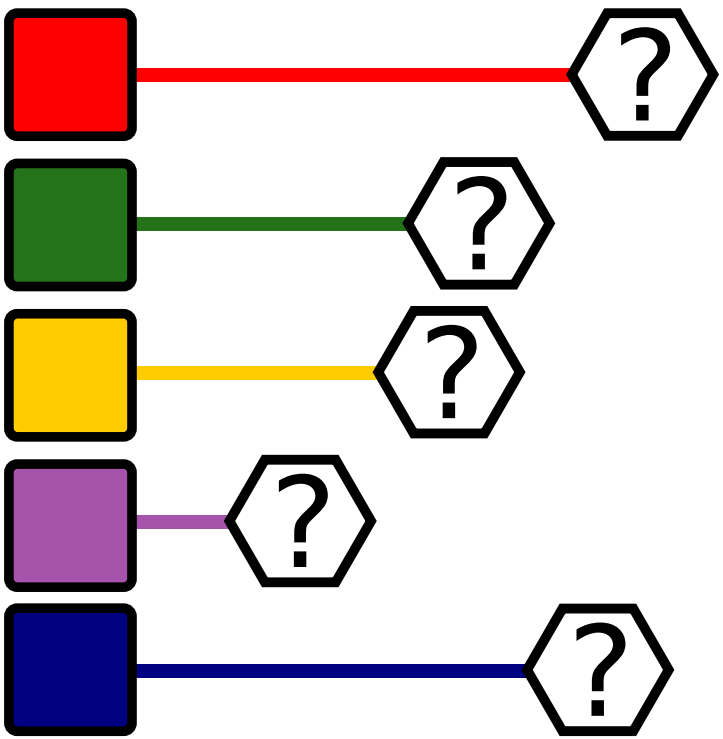
predict( , ) = *f*-score
(predicted)

Regression features

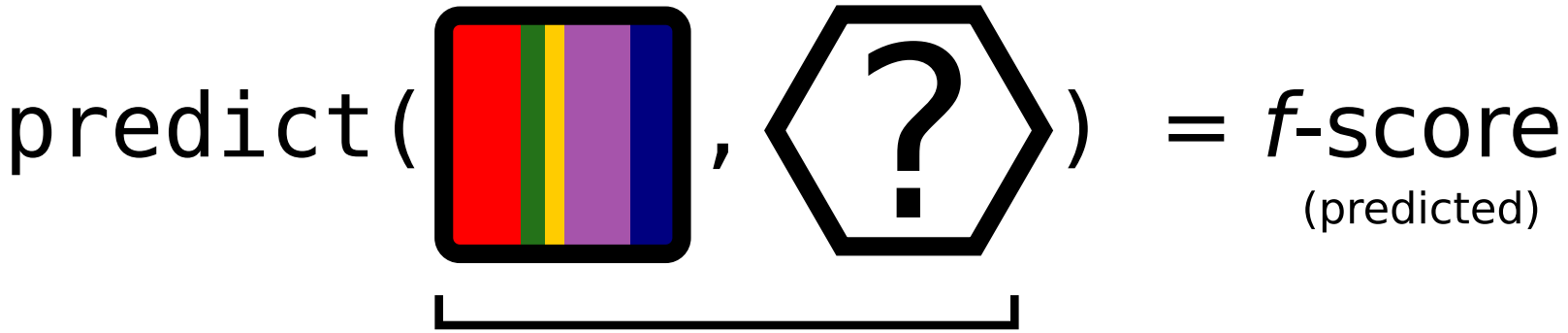


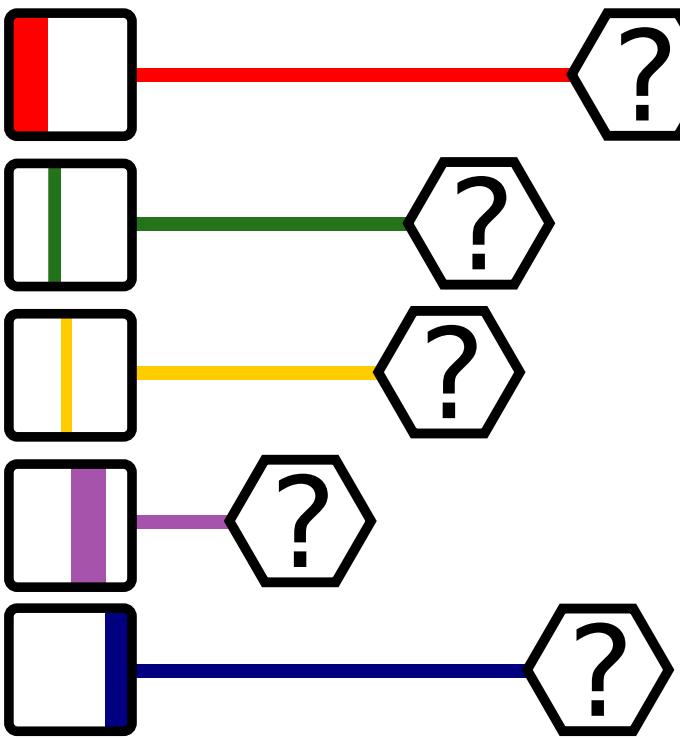
Regression features



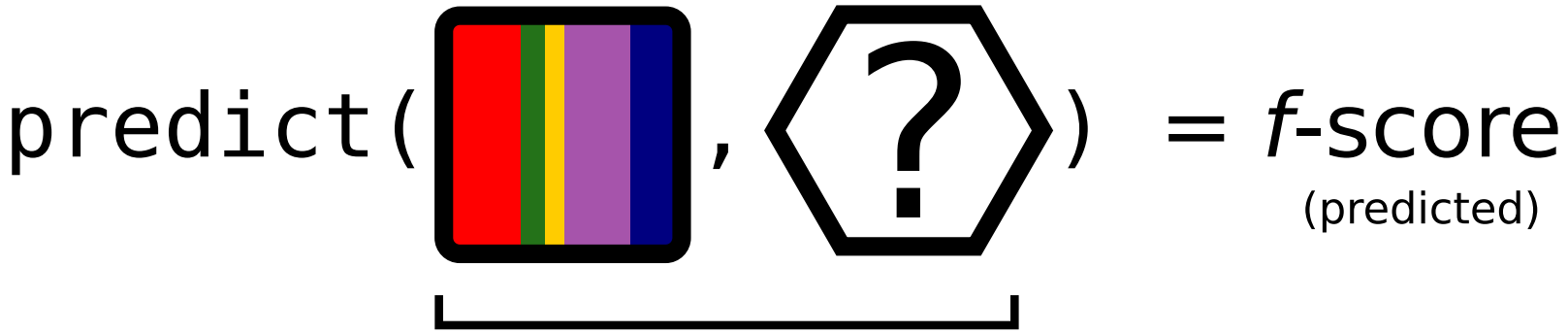


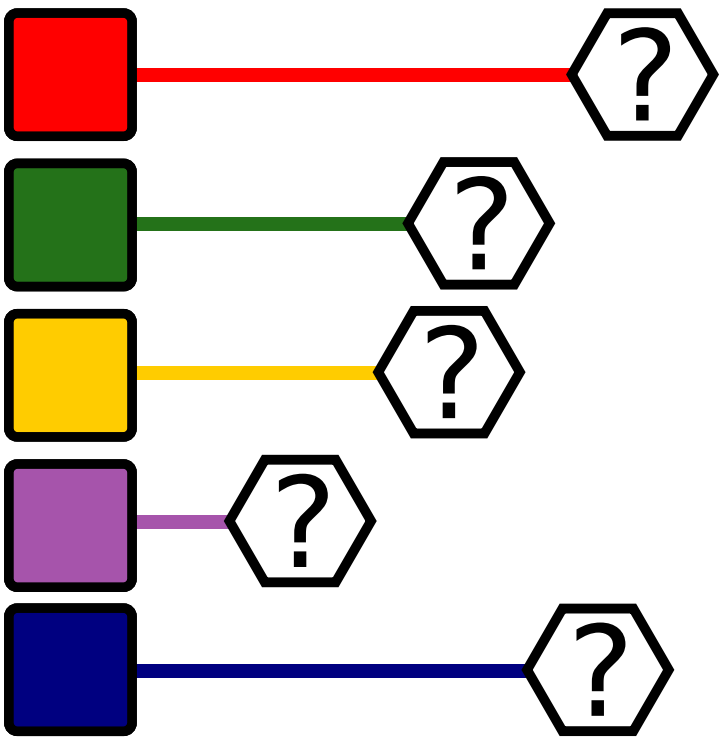
Regression features



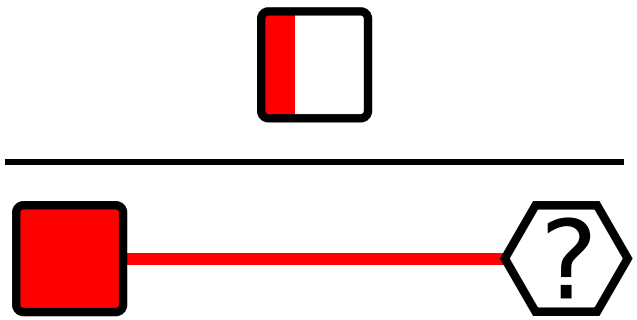


Regression features

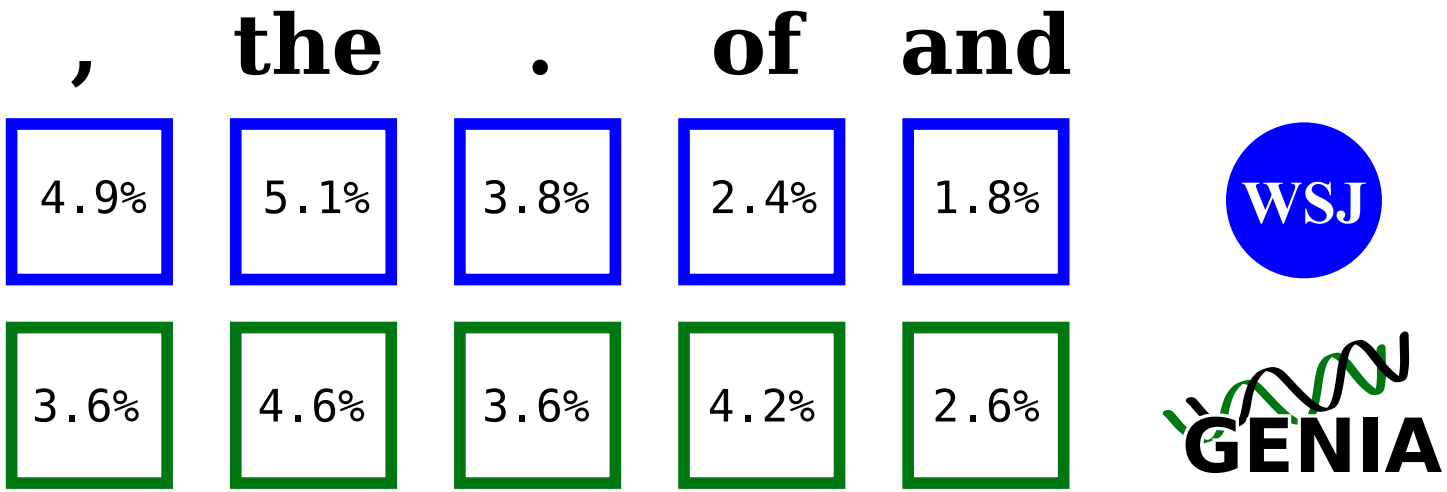




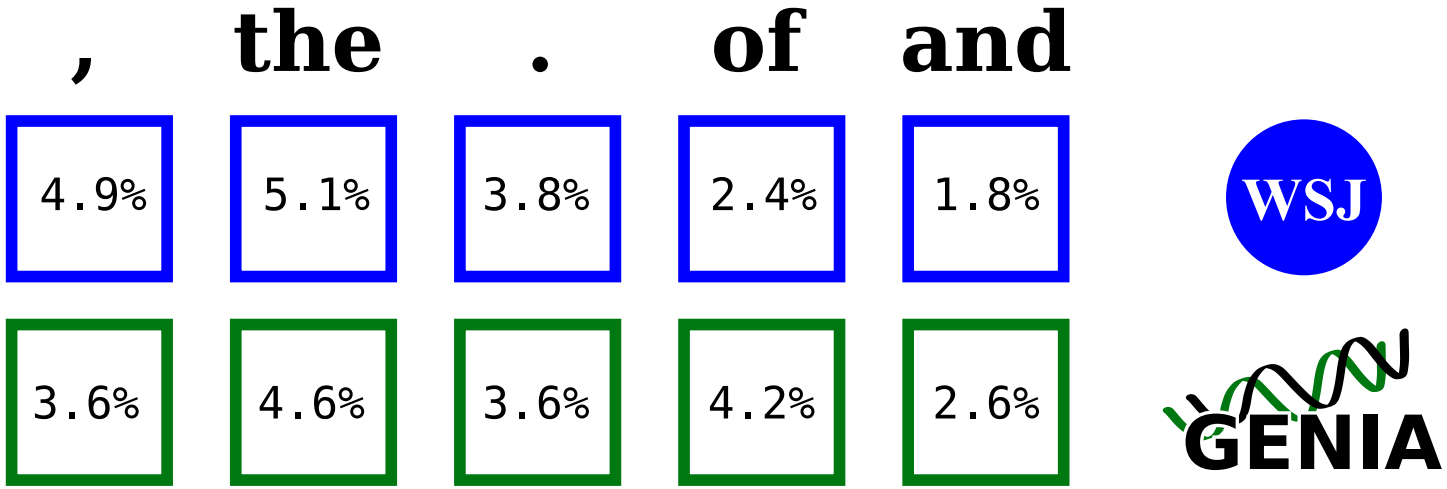
Divide mixture weight by divergence:



Cosine Similarity



Cosine Similarity



$$\text{cosine similarity} = \frac{\text{WSJ} \cdot \text{GENIA}}{\|\text{WSJ}\| \, \|\text{GENIA}\|} \approx 0.956$$

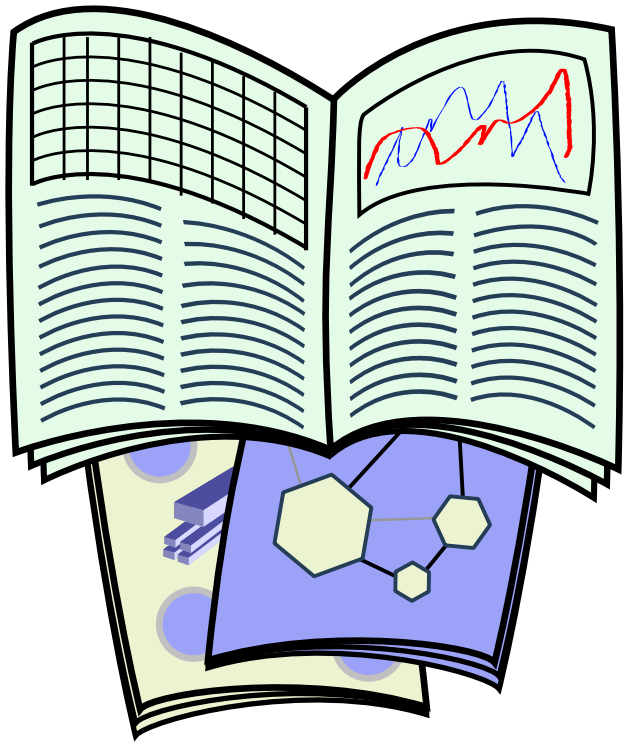
Unknown words

= vocabulary

Unknown words

WSJ

GENIA

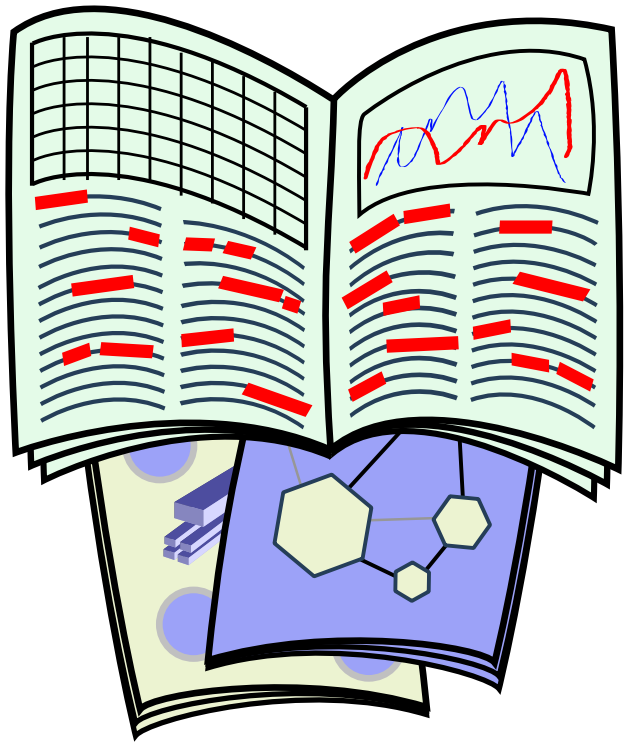


= vocabulary

Unknown words

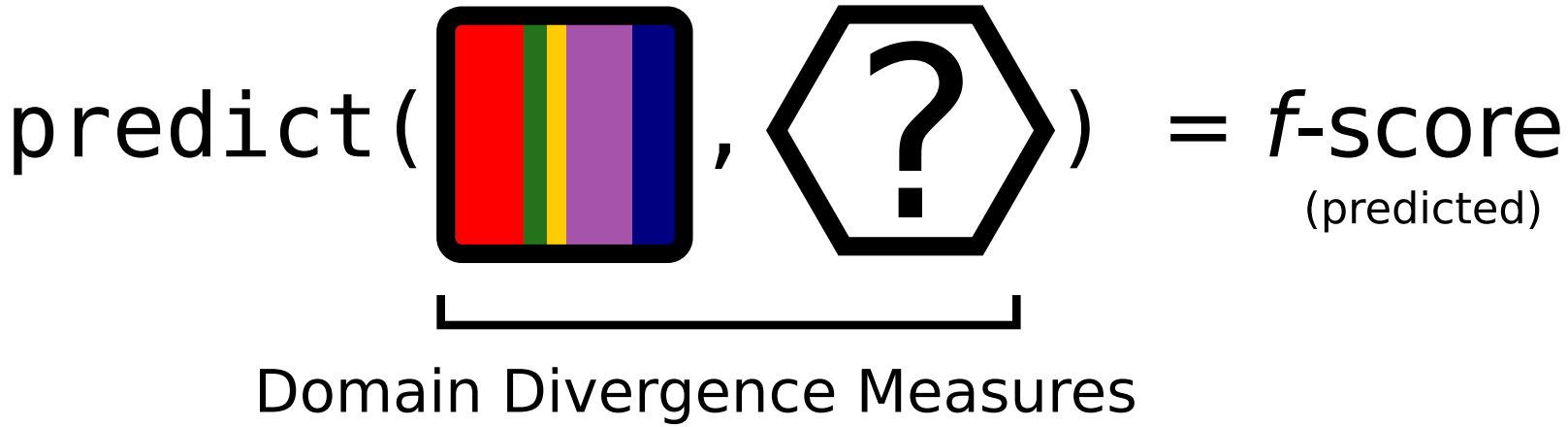
WSJ

GENIA

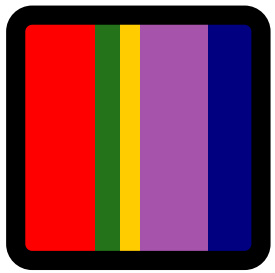


= vocabulary

Regression features

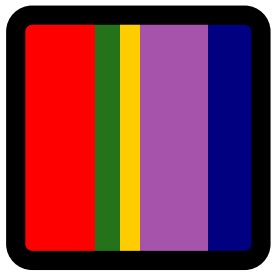


Regression features

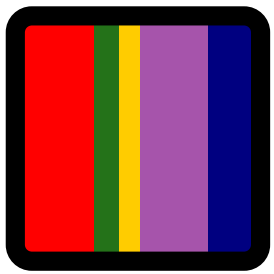
predict( , ) = *f*-score (predicted)

Source domain features

Regression features

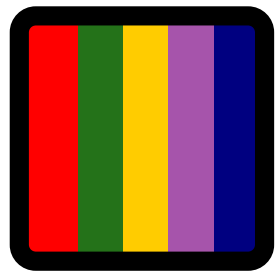
predict( , ) = *f*-score (predicted)

Source domain features



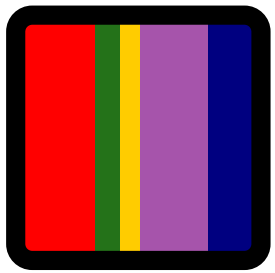
source domain mixture

vs.

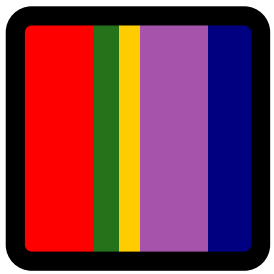


uniform

Regression features

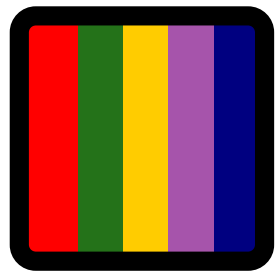
predict( , ) = *f*-score (predicted)

Source domain features



source domain mixture

vs.



uniform

Entropy: $\mathrm{H}(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i)$

Features considered

- Domain divergence measures
  - n-gram language model (PPL, PPL1, probability)
  - Cosine similarity for frequent words ($k \in \{5, 50, 500, 5000\}$)
  - Cosine similarity for punctuation
  - Average length differences (absolute, directed)
  - % unknown words (source $\rightarrow$ target, target $\rightarrow$ source)
- Source domain features
  - Source domain probabilities
  - Source domain non-zero probability
  - # source domains
  - % self-trained corpora
  - Source domain entropy

Features considered

- Domain divergence measures
  - n-gram language model (PPL, PPL1, probability)
  - **Cosine similarity for frequent words**
    ($k \in \{5, \mathbf{50}, 500, 5000\}$)
  - Cosine similarity for punctuation
  - Average length differences (absolute, directed)
  - **% unknown words** (source $\rightarrow$ target, **target $\rightarrow$ source**)
- Source domain features
  - Source domain probabilities
  - Source domain non-zero probability
  - # source domains
  - % self-trained corpora
  - **Source domain entropy**

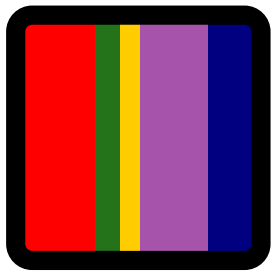
Cosine similarity illustrated ($k = 5000$)

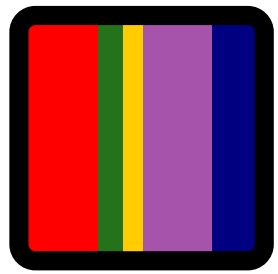
Source domain	Target domain: BNC	GENIA	BROWN	SWBD	ETT	WSJ
GENIA	0.894	0.998	0.860	0.676	0.887	0.881
PUBMED	0.911	0.977	0.875	0.697	0.895	0.897
BROWN	0.976	0.862	0.999	0.828	0.917	0.960
GUTENBERG	0.982	0.868	0.977	0.839	0.929	0.957
SWBD	0.779	0.663	0.825	0.992	0.695	0.789
ETT	0.971	0.896	0.937	0.766	0.992	0.959
WSJ	0.968	0.880	0.963	0.803	0.941	0.997
NANC	0.983	0.888	0.979	0.801	0.950	0.987

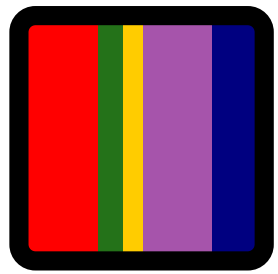
Unknown words illustrated (*target* → *source*)

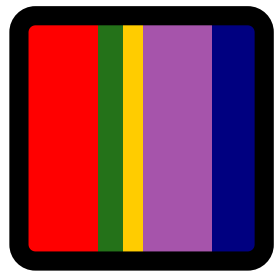
Source domain	Target domain: BNC	GENIA	BROWN	SWBD	ETT	WSJ
GENIA	33.3	10.8	40.5	45.8	43.1	38.9
PUBMED	32.5	21.5	36.5	45.4	42.0	35.5
BROWN	14.3	38.5	10.7	21.5	22.7	18.3
GUTENBERG	16.0	36.9	14.3	23.7	23.2	20.0
SWBD	9.0	30.6	6.1	4.6	11.1	11.4
ETT	18.1	35.3	17.4	22.1	10.3	16.6
WSJ	23.1	41.1	22.5	30.1	25.4	14.2
NANC	20.4	39.8	19.3	27.1	24.5	18.3

Model and estimation

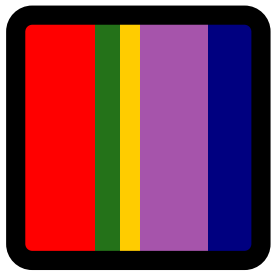
predict( , ) =

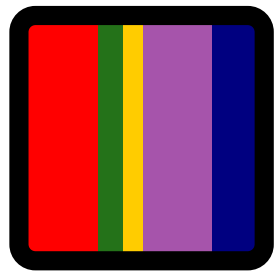
$\vec{\lambda}_c$ cosinesim( , )

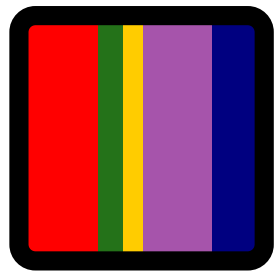
$+ \vec{\lambda}_w$ unkwords( , )

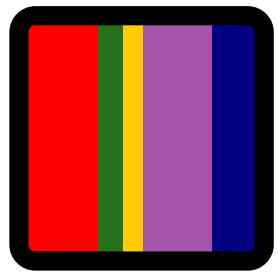
$+ \lambda_u$ entropy() + b

Model and estimation

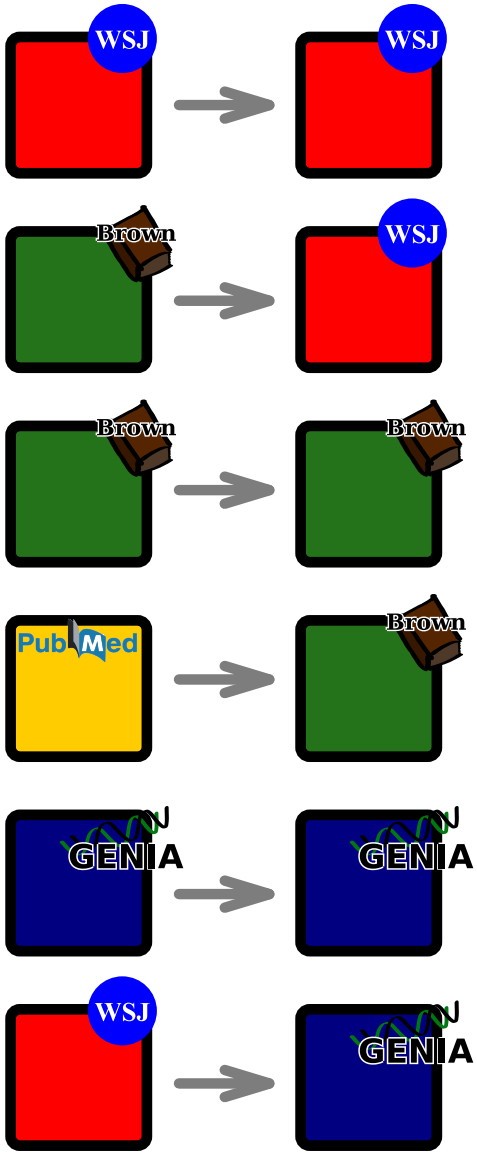
predict( , ) =

$\vec{\lambda}_c$ cosinesim( , )

$+ \vec{\lambda}_w$ unkwords( , )

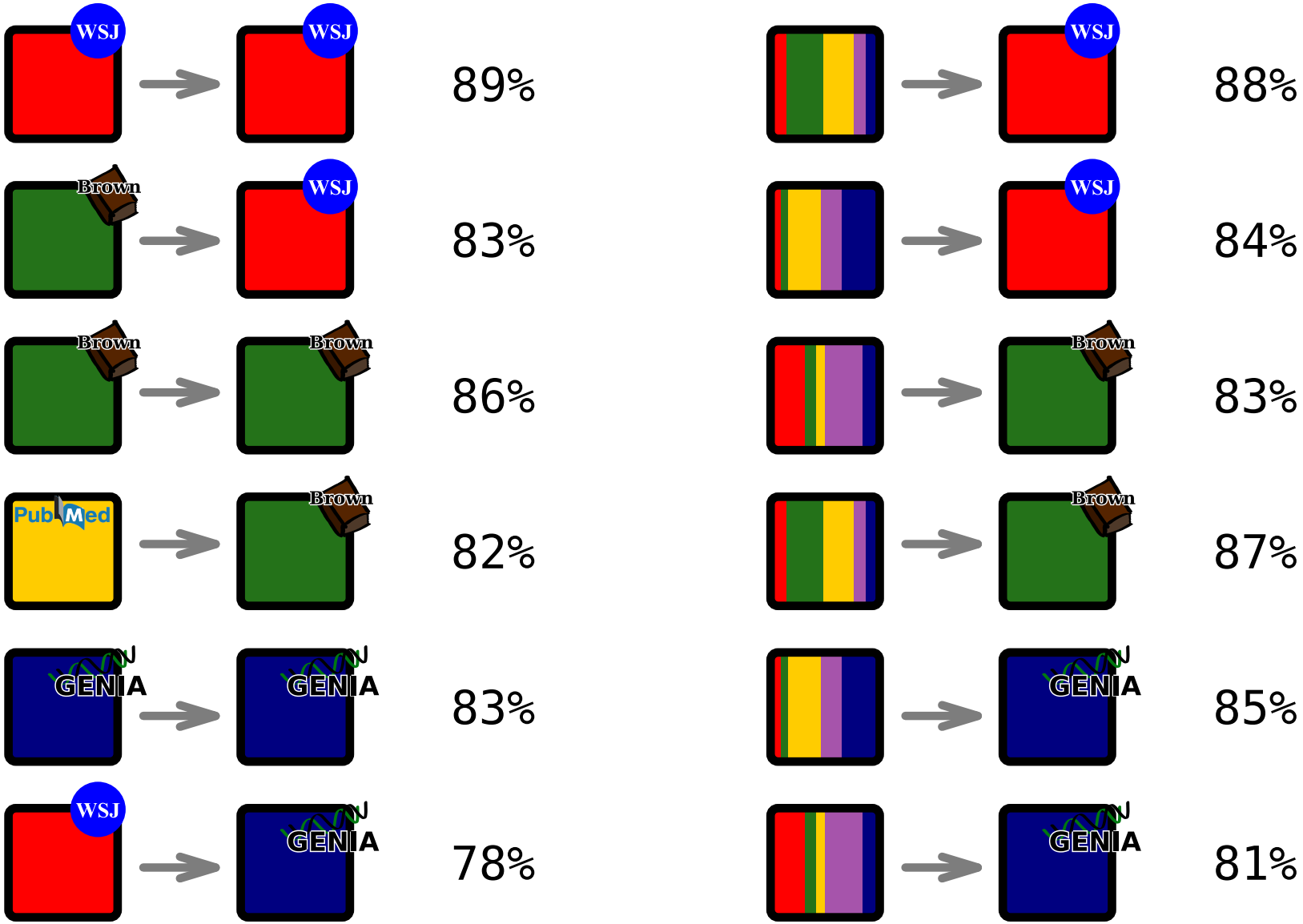
$+ \lambda_u$ entropy() + b

Training data



* numbers on this slide are cooked

Training data



* numbers on this slide are cooked

Corpora used

Corpus	Source domain	Target domain
BNC		●
BROWN	●	●
ETT	●	●
GENIA	●	●
PUBMED	●	
SWBD	●	●
WSJ	●	●
NANC	●	
GUTENBERG	●	
PUBMED	●	

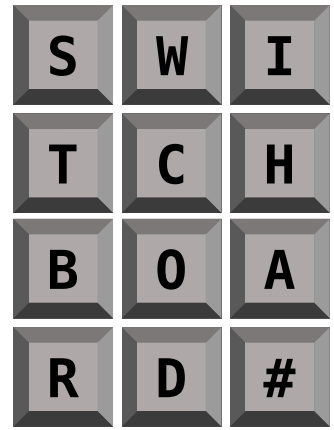
Corpora used

Corpus	Source domain	Target domain
BNC		●
BROWN	●	●
ETT	●	●
GENIA	●	●
PUBMED	●	
SWBD	●	●
WSJ	●	●
NANC	●	
GUTENBERG	●	
PUBMED	●	

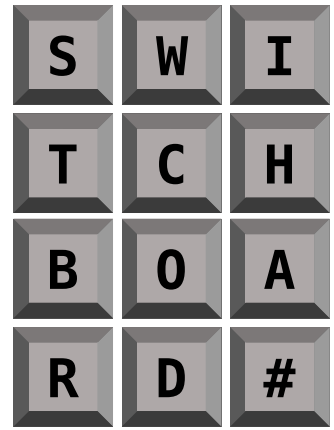
Corpora used

Corpus	Source domain	Target domain
BNC		●
BROWN	●	●
ETT	●	●
GENIA	●	●
PUBMED	●	
SWBD	●	●
WSJ	●	●
NANC	●	
GUTENBERG	●	
PUBMED	●	

Round-robin evaluation



Round-robin evaluation

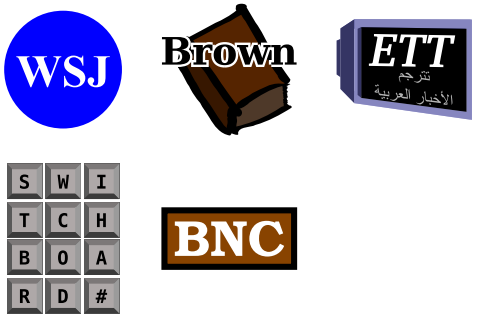


Evaluation for GENIA

train

sources

targets

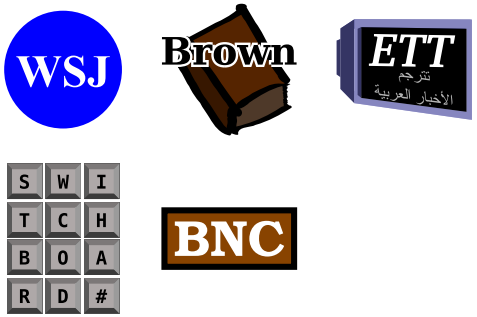


Evaluation for GENIA

train

sources

targets



---

test

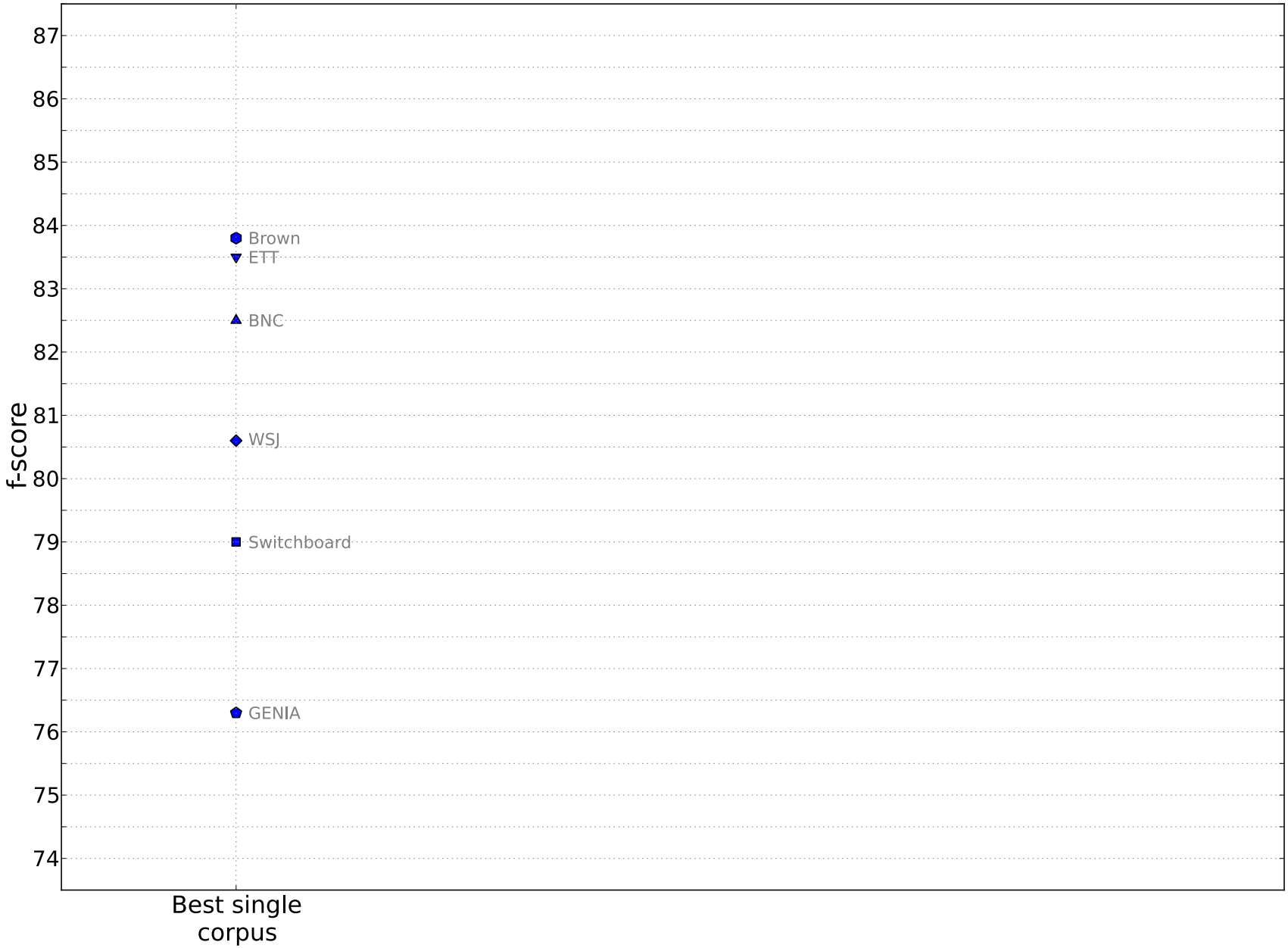
sources

target

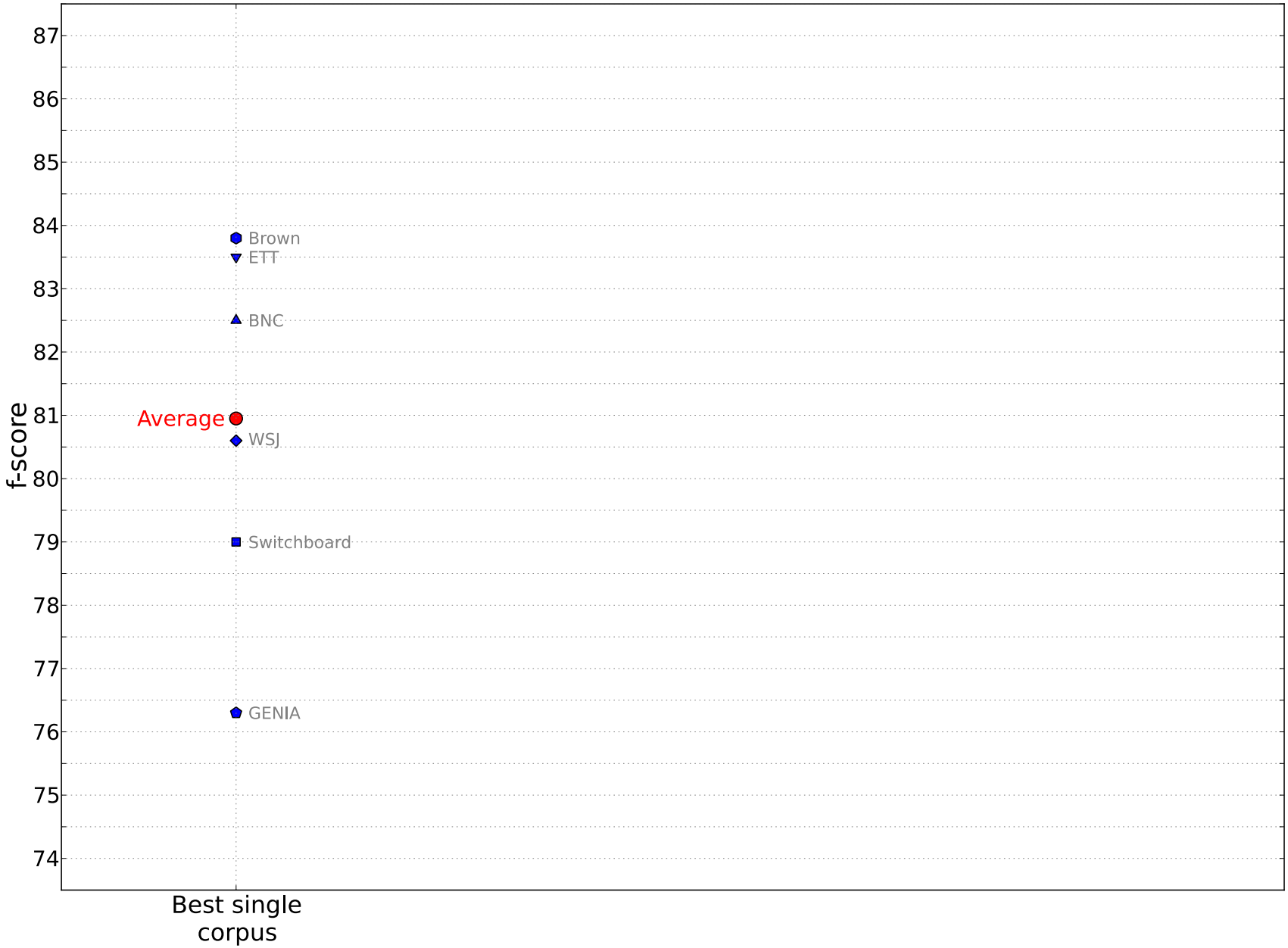
Baselines

- Standard baselines
  - Uniform with labeled corpora
  - Uniform with labeled and self-trained corpora
  - Fixed set: `WSJ`
- Oracle baselines
  - Best single corpus
  - Best seen

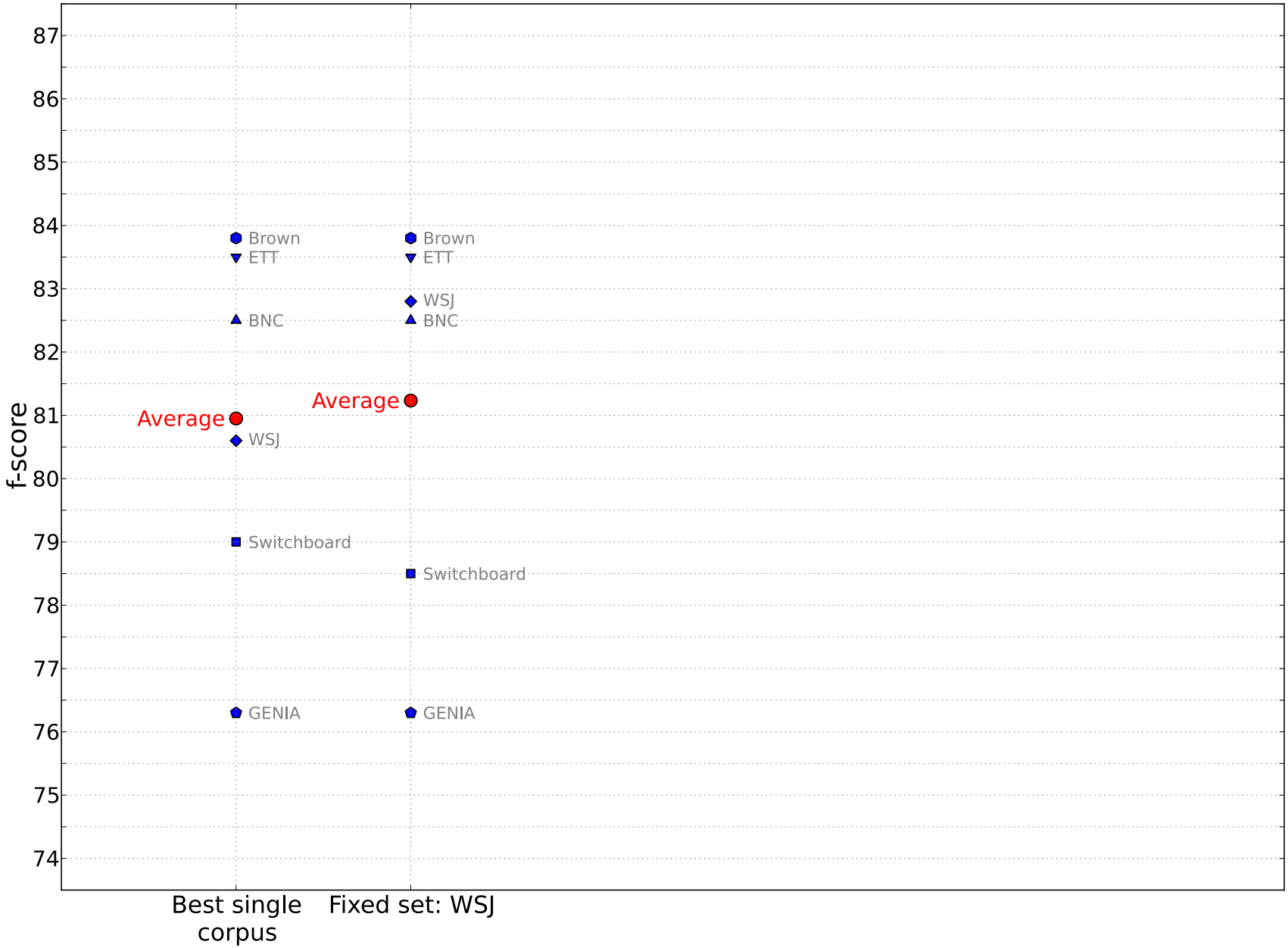
Evaluation results



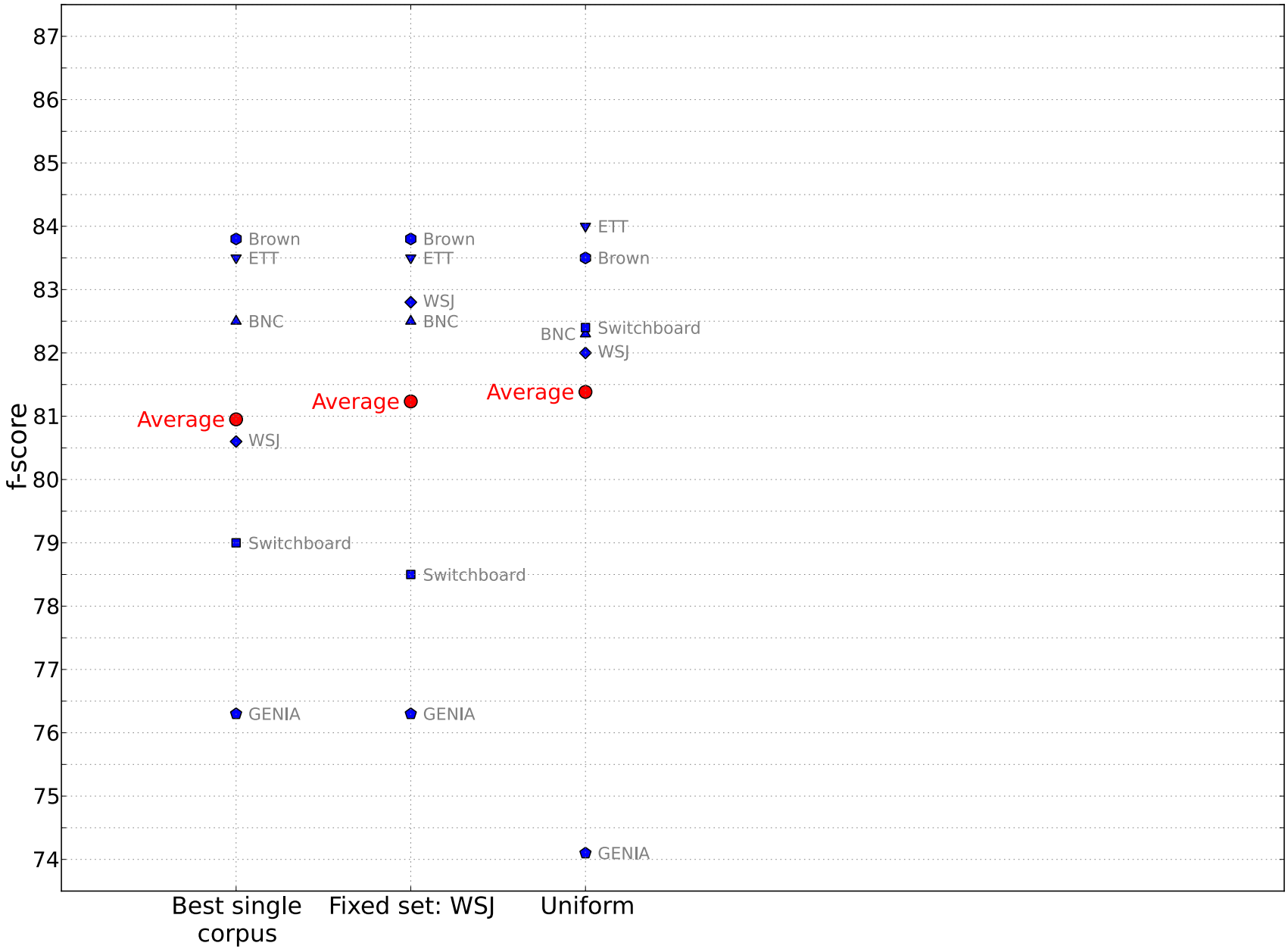
Evaluation results



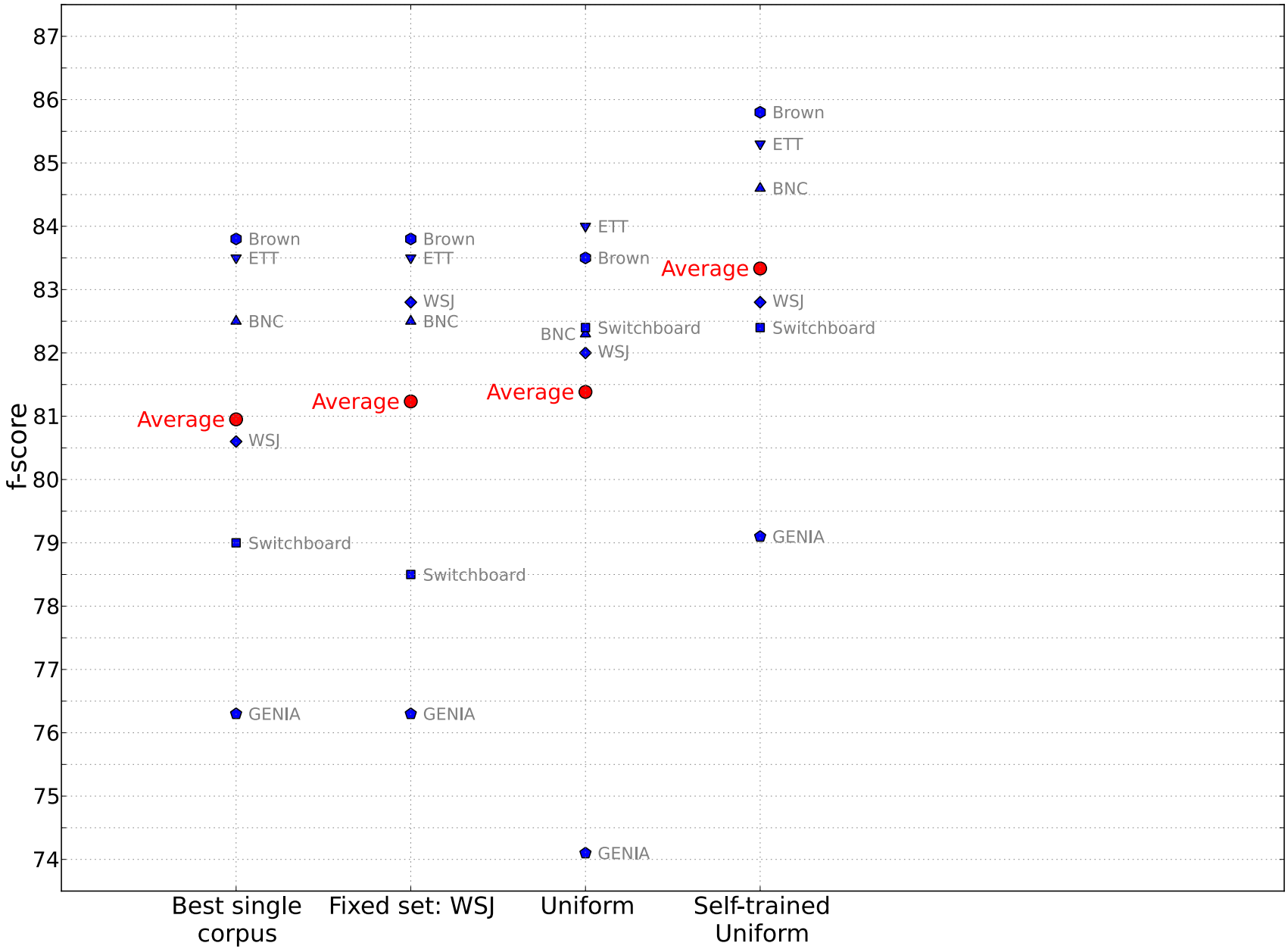
Evaluation results



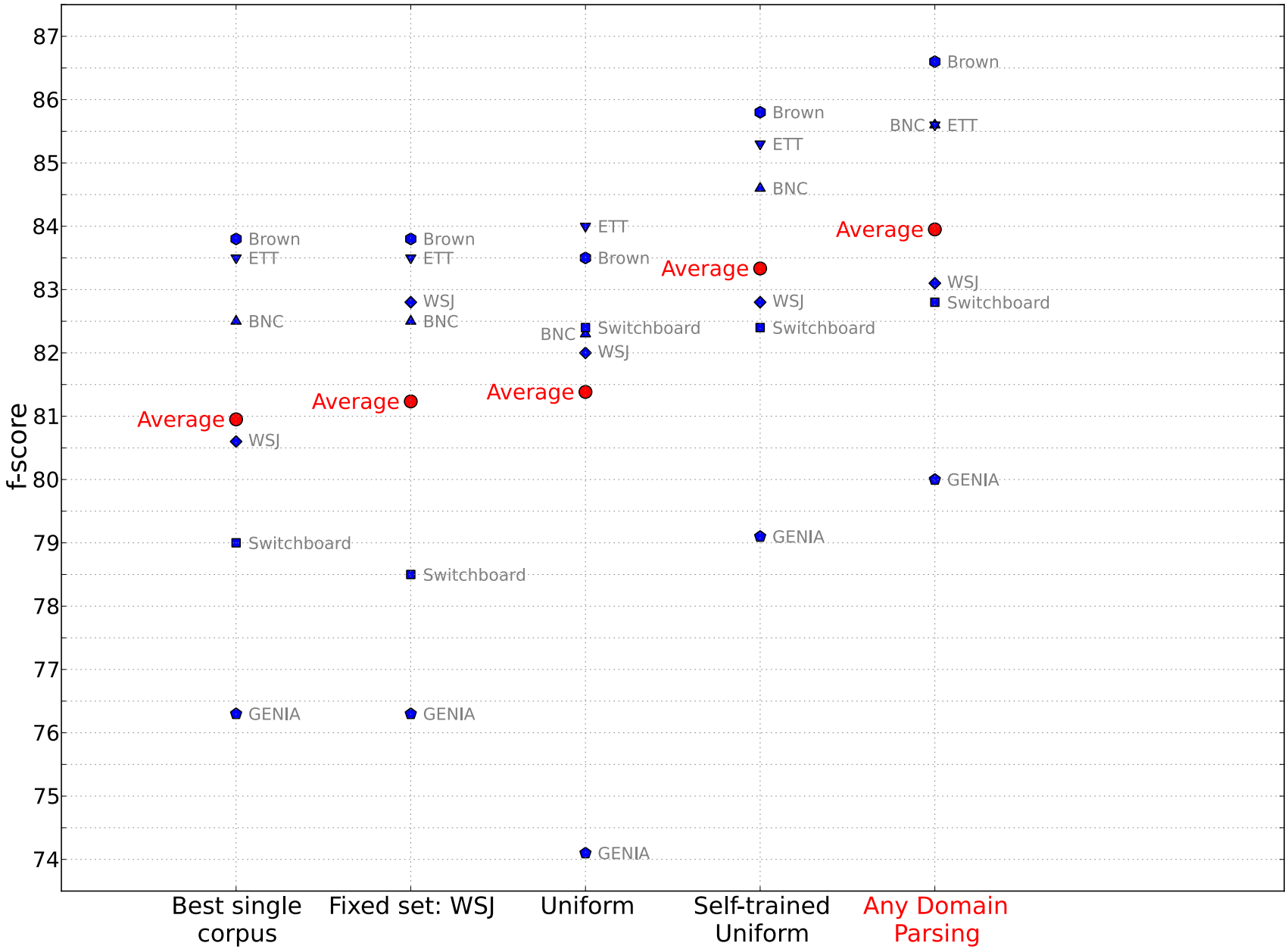
Evaluation results



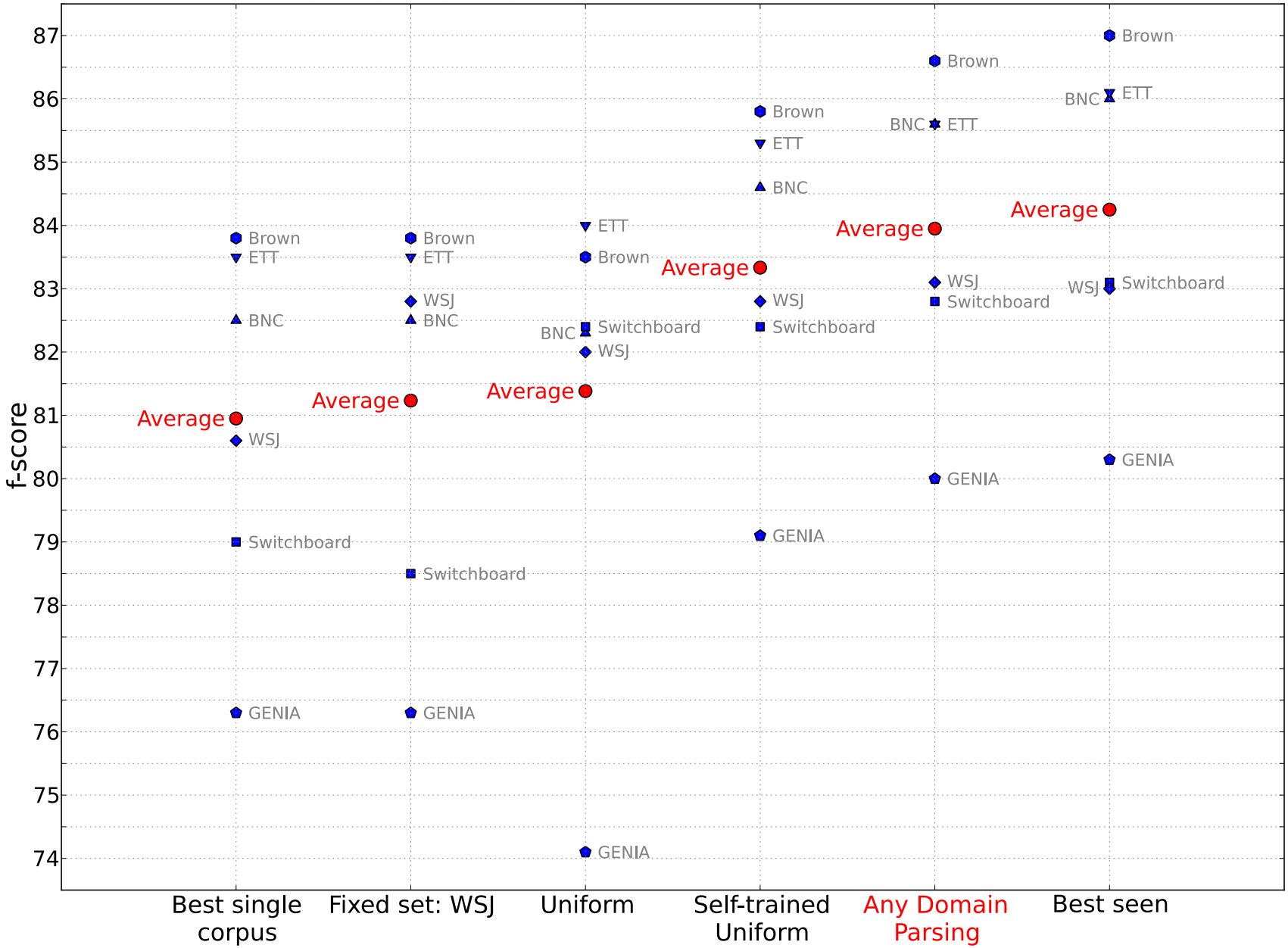
Evaluation results



Evaluation results



Evaluation results



Moral of the story

- Domain differences can be captured by surface features
- Any Domain Parsing:
  - near-optimal performance for out-of-domain evaluation
  - domain-specific parsing models are beneficial
- Self-trained corpora improve accuracy across domains

Future work

In order of decreasing $\frac{bang}{buck}$:

- Automatically adapting the reranker (and other non-linear models)
- Other parsing model combination strategies
- Applying to other tasks
- Non-linear regression
- Syntactic features

May The Force Be With You

Questions?

Thanks to the members of the Brown, Berkeley, and Stanford NLP groups for their feedback and support!

Brought to you by NSF grants LIS9720368 and IIS0095940 and DARPA GALE contract HR0011-06-2-0001

Extra slides

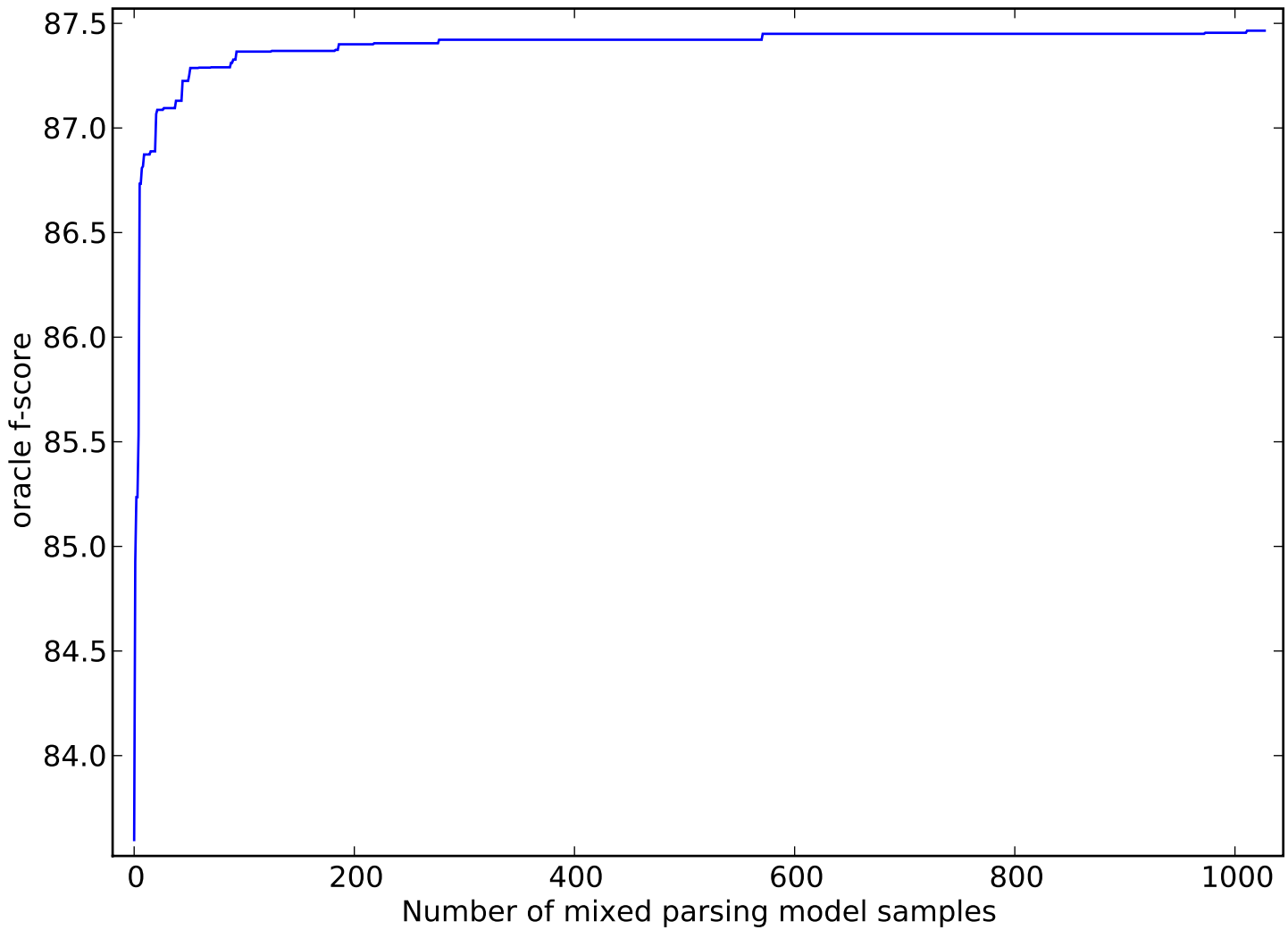
Sampling parsing models

Goal: parsing models with many different subsets of corpora

1. Sample $n = \#$ source domains from exponential distribution
2. Sample probabilities for n corpora from n-simplex
3. Sample names for n corpora

Repeat until “done”

Average oracle *f*-score

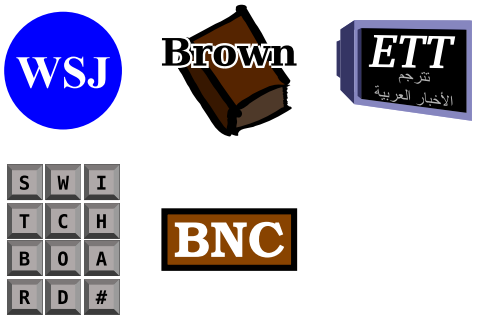


Out-of-domain evaluation for GENIA

train

sources

targets



---

test

sources

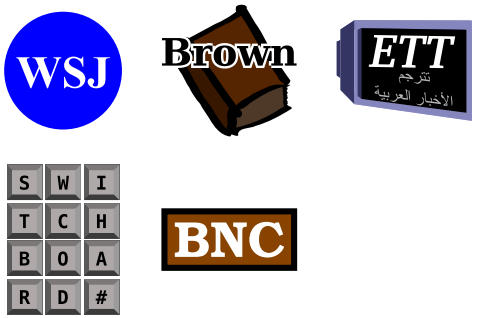
target

In-domain evaluation for GENIA

train

sources

targets



---

test

sources

target

Tuning parameters

- We want to select regression model, features
- Evaluation is round-robin
- Tuning can be done with nested round-robins
  - hold out one target corpus entirely
  - round-robin on each remaining target corpus
- This results in 30 small tuning scenarios

Tuning metrics

- Three metrics to do model/feature selection:

- These metrics are summed across all 30 tuning scenarios
- Parallelized best-first search explored 6,000 settings
- Our best setting performs well over all three metrics: cosine (k=50), unknown words (*target* $\rightarrow$ *source*), entropy

Tuning metrics

- Three metrics to do model/feature selection:
  1. mean squared error:
     $$\sum(\text{true} - \text{predicted})^2$$

- These metrics are summed across all 30 tuning scenarios
- Parallelized best-first search explored 6,000 settings
- Our best setting performs well over all three metrics: cosine (k=50), unknown words (*target* $\rightarrow$ *source*), entropy

Tuning metrics

- Three metrics to do model/feature selection:
  1. mean squared error:
     $$\sum(\text{true} - \text{predicted})^2$$
  2. modified mean squared error:
     $$\sum |\text{true} - \text{predicted}|^{1+\text{true}}$$

- These metrics are summed across all 30 tuning scenarios
- Parallelized best-first search explored 6,000 settings
- Our best setting performs well over all three metrics: cosine (k=50), unknown words (*target* $\rightarrow$ *source*), entropy

Tuning metrics

- Three metrics to do model/feature selection:
  1. mean squared error:
     $$\sum(\text{true} - \text{predicted})^2$$
  2. modified mean squared error:
     $$\sum |\text{true} - \text{predicted}|^{1+\text{true}}$$
  3. oracle loss:
     $$\max(\text{true}) - \text{evaluate}(\max(\text{predicted}))$$
- These metrics are summed across all 30 tuning scenarios
- Parallelized best-first search explored 6,000 settings
- Our best setting performs well over all three metrics: cosine (k=50), unknown words (*target* $\rightarrow$ *source*), entropy

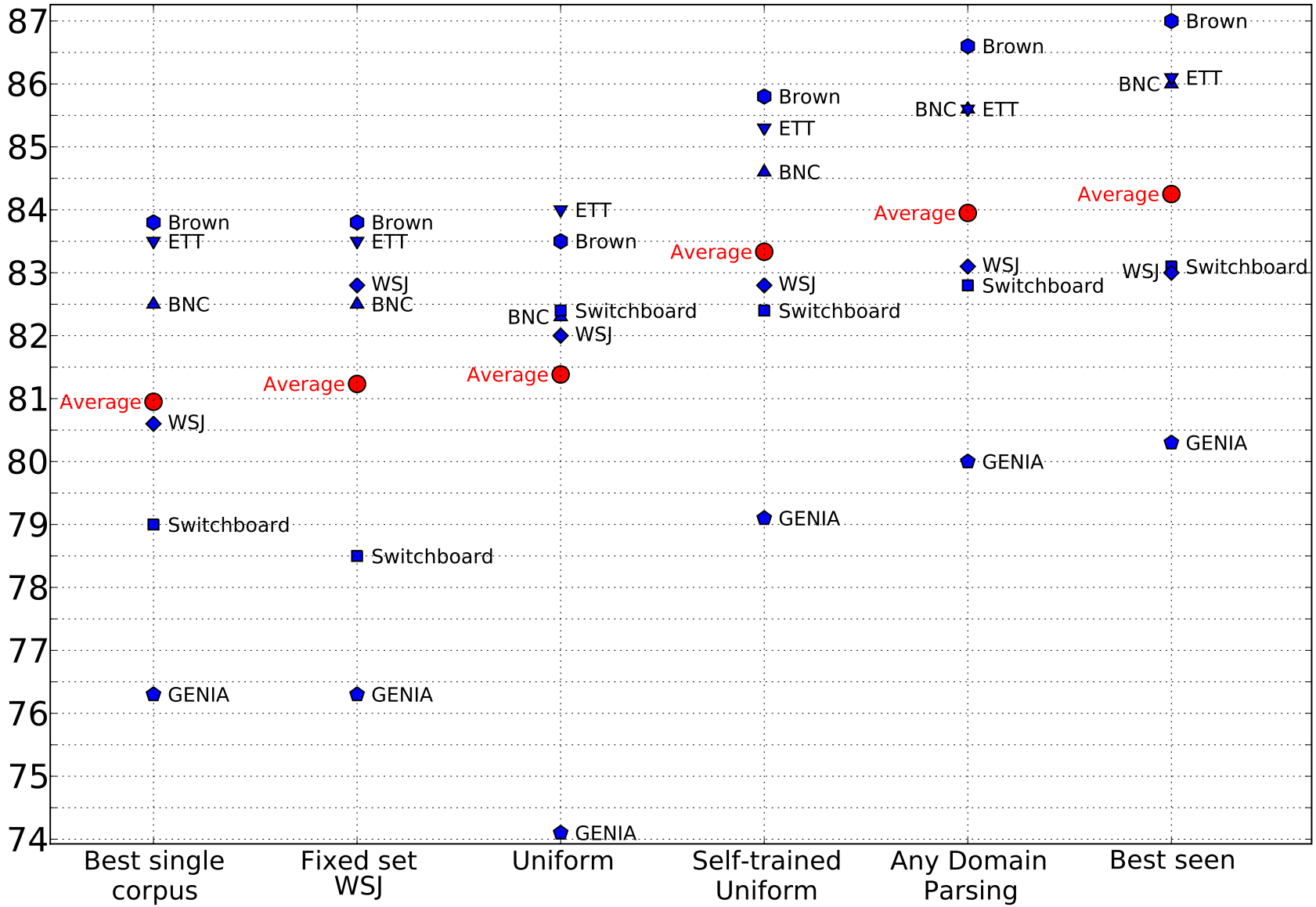
Feature interactions

$+$	Cosine ($k = 50$)
$+$	Unknown words
$-$	Relative entropy

$+$	Unknown words
$-$	Relative entropy

$-$	Cosine ($k = 50$)
$-$	Relative entropy

Out-of-domain evaluation results



In-domain evaluation results