When is Self-training Effective for Parsing?

David McClosky
dmcc@cs.brown.edu

Brown Laboratory for Linguistic Information Processing (BLLIP)

Joint work with Eugene Charniak and Mark Johnson

Outline

- What is self-training?
- Previous work
- Experimental setup
- Four hypotheses
- Conclusions

Outline

- **What is self-training?**
- Previous work
- Experimental setup
- Four hypotheses
- Conclusions

Self-training Requirements

- Labeled data
- Large amount of unlabeled data
- Statistical model:
  - `model  = train(labeled data)`
  - `labels = label(model, unlabeled data)`

Self-training Pseudocode

```
base = train(labeled)
autolabeled = label(base, unlabeled)
combined = labeled + autolabeled
selftrained = train(combined)
```

(Not pictured: data selection, iteration, weighting, etc.)

Outline

- What is self-training?
- **Previous work**
- Experimental setup
- Four hypotheses
- Conclusions

Previous Work

	Parser type	Seed size	Iterations	Improved?
Charniak (1997)	Generative	Large	Single	No
McClosky et al. (2006)	Gen.+Disc.	Large	Single	**Yes**
Steedman et al. (2003)	Generative	Small	Multiple	No
Reichart and Rappoport (2007)	Generative	Small	Single	**Yes**

(large = ~40k sentences, small = <1k sentences)

Summary of self-training for parsing experiments

Previous Work

	Parser type	Seed size	Iterations	Improved?
Charniak (1997)	Generative	Large	Single	No
McClosky et al. (2006)	Gen.+Disc.	Large	Single	**Yes**
Steedman et al. (2003)	Generative	Small	Multiple	No
Reichart and Rappoport (2007)	Generative	Small	Single	**Yes**

(large = ~40k sentences, small = <1k sentences)

Summary of self-training for parsing experiments

- In large seed case, generative + discriminative parser is necessary.

Previous Work

	Parser type	Seed size	Iterations	Improved?
Charniak (1997)	Generative	Large	Single	No
McClosky et al. (2006)	Gen.+Disc.	Large	Single	**Yes**
Steedman et al. (2003)	Generative	Small	Multiple	No
Reichart and Rappoport (2007)	Generative	Small	Single	**Yes**

(large = ~40k sentences, small = <1k sentences)

Summary of self-training for parsing experiments

- In large seed case, generative + discriminative parser is necessary.
- Performing only one iteration is better than multiple iterations.

Previous Analysis

	Seed size	Predictor	
		Length	# unknown words
McClosky et al. (2006)	Large	+	-
Reichart and Rappoport (2007)	Small	+	+

Previous Analysis

	Seed size	Predictor	
		Length	# unknown words
McClosky et al. (2006)	Large	+	-
Reichart and Rappoport (2007)	Small	+	+

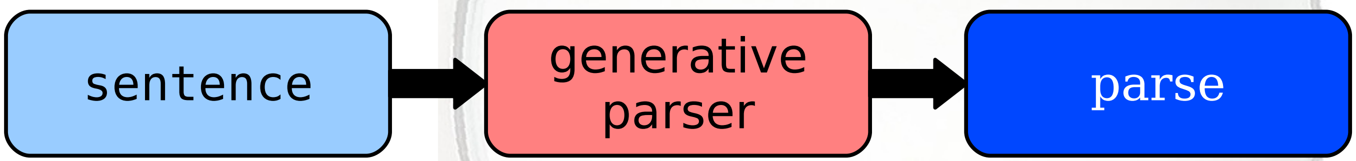
- Unknown words are a good predictor of self-training's success **only** in the small seed case.

Outline

- What is self-training?
- Previous work
- **Experimental setup**
- Four hypotheses
- Conclusions

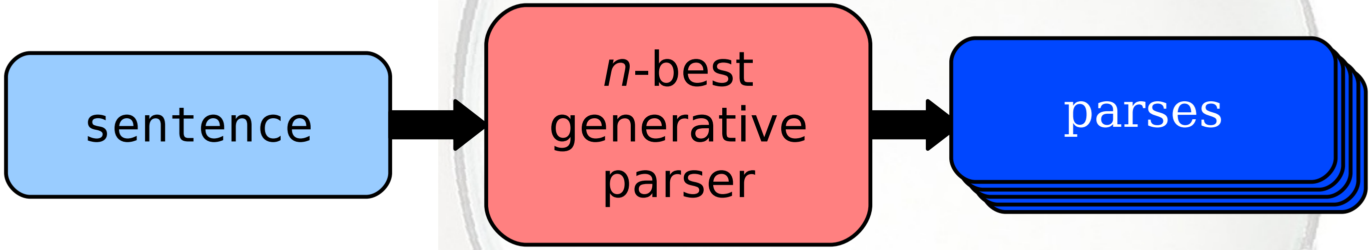
Experimental Setup

- Labeled data: WSJ (Marcus et al., 1993)
- Unlabeled data: NANC (Graff, 1995)
- Parser: Charniak and Johnson (2005) reranking parser



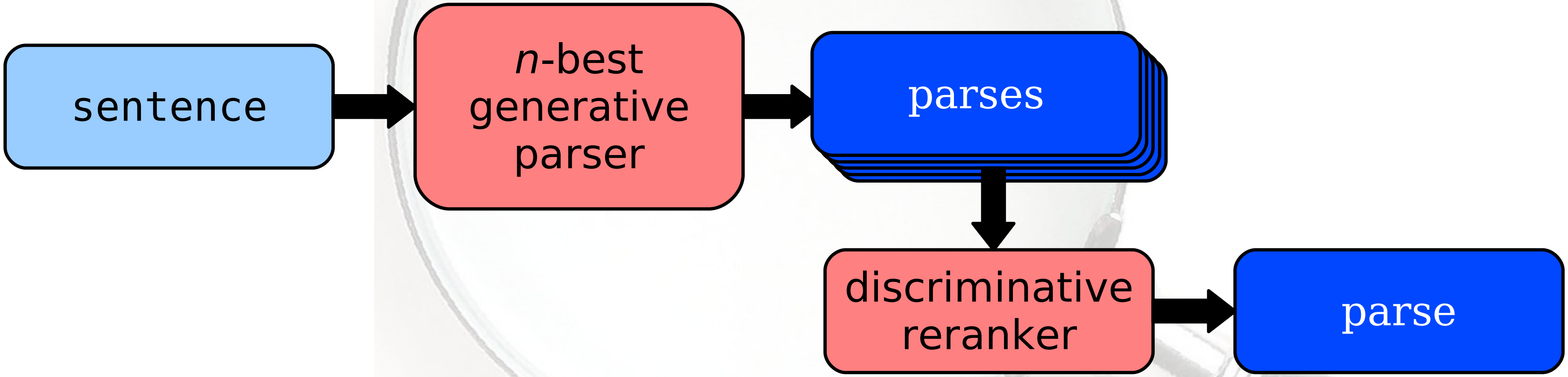
Experimental Setup

- Labeled data: WSJ (Marcus et al., 1993)
- Unlabeled data: NANC (Graff, 1995)
- Parser: Charniak and Johnson (2005) reranking parser



Experimental Setup

- Labeled data: WSJ (Marcus et al., 1993)
- Unlabeled data: NANC (Graff, 1995)
- Parser: Charniak and Johnson (2005) reranking parser



Outline

- What is self-training?
- Previous work
- Experimental setup
- **Four hypotheses**
- Conclusions

Hypotheses for Self-training

1 Phase Transition

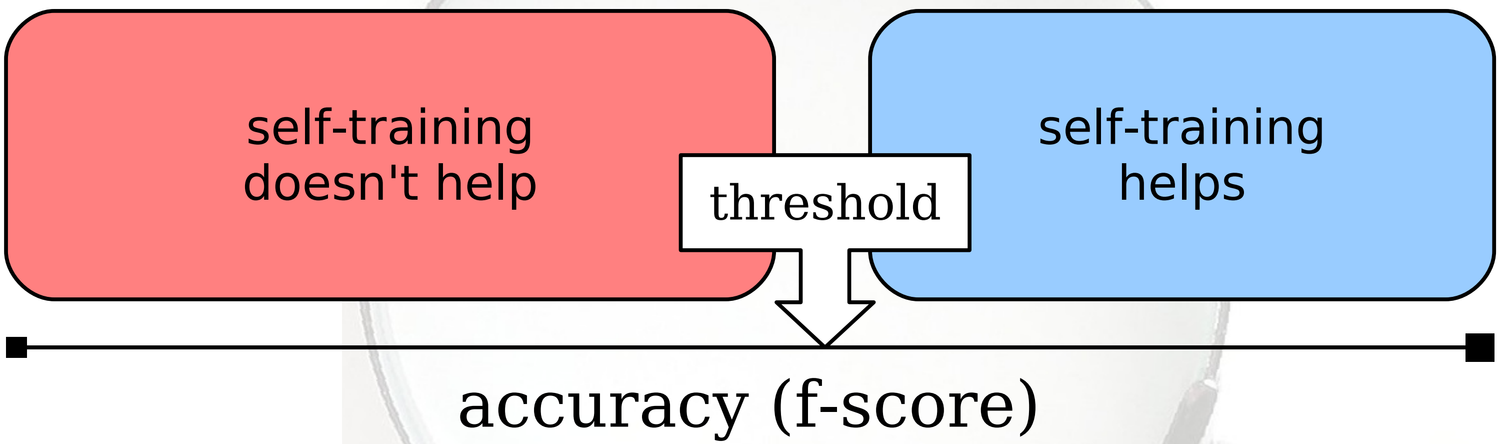
2 Search Errors

3 Non-generative Reranker Features

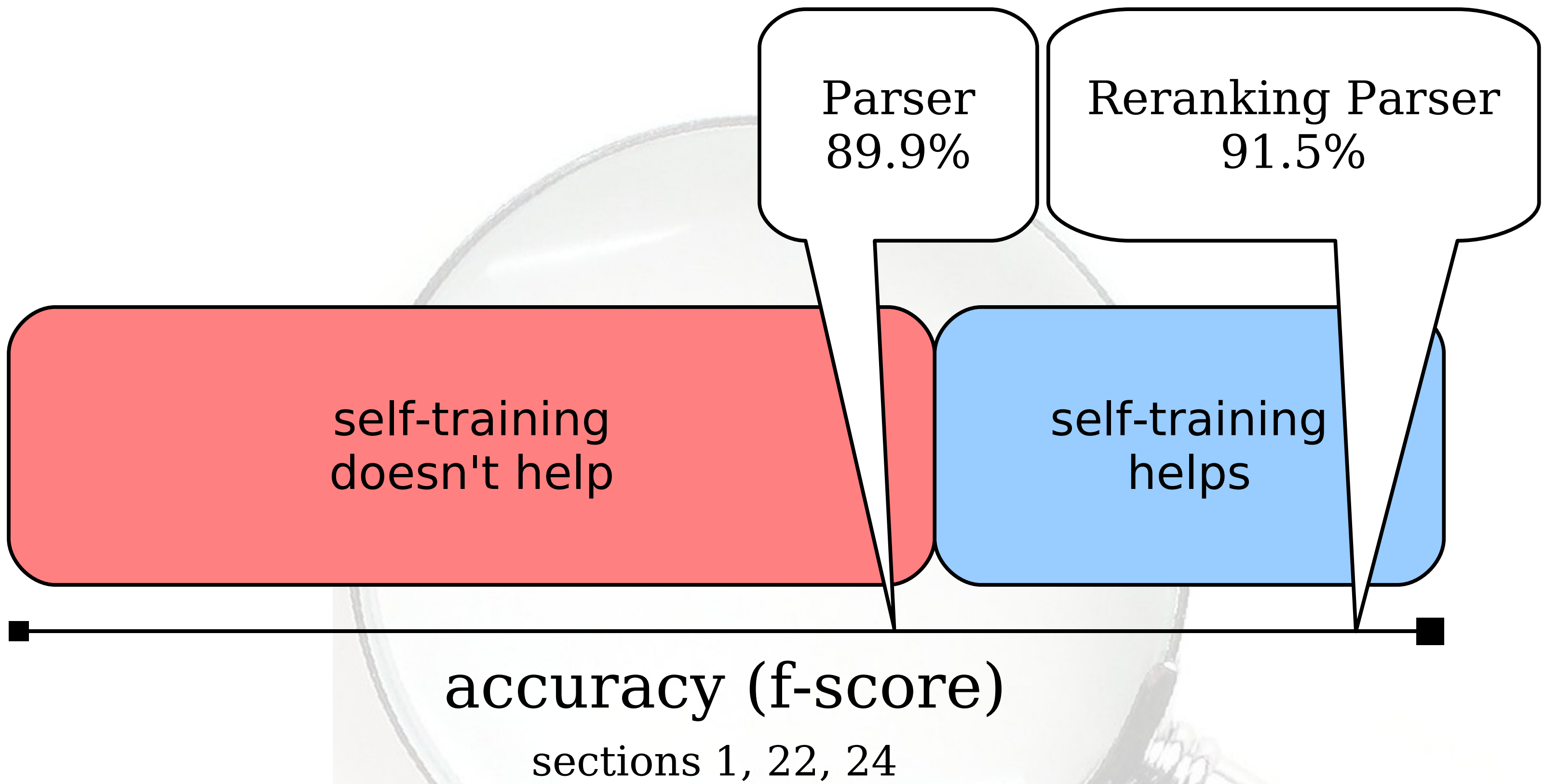
4 Bilexical Dependencies

Phase Transition

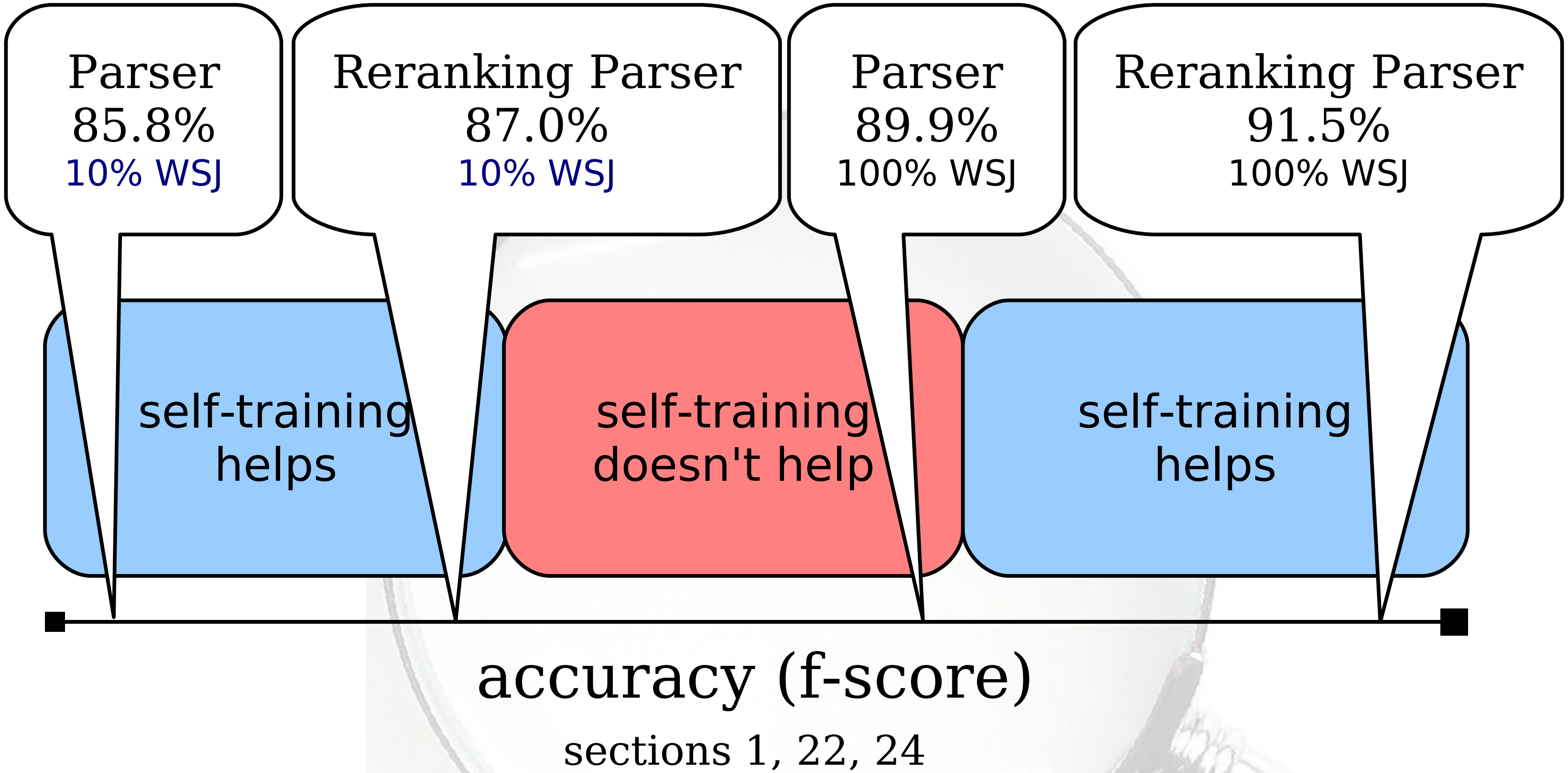
Self-training works after a phase transition



Phase Transition

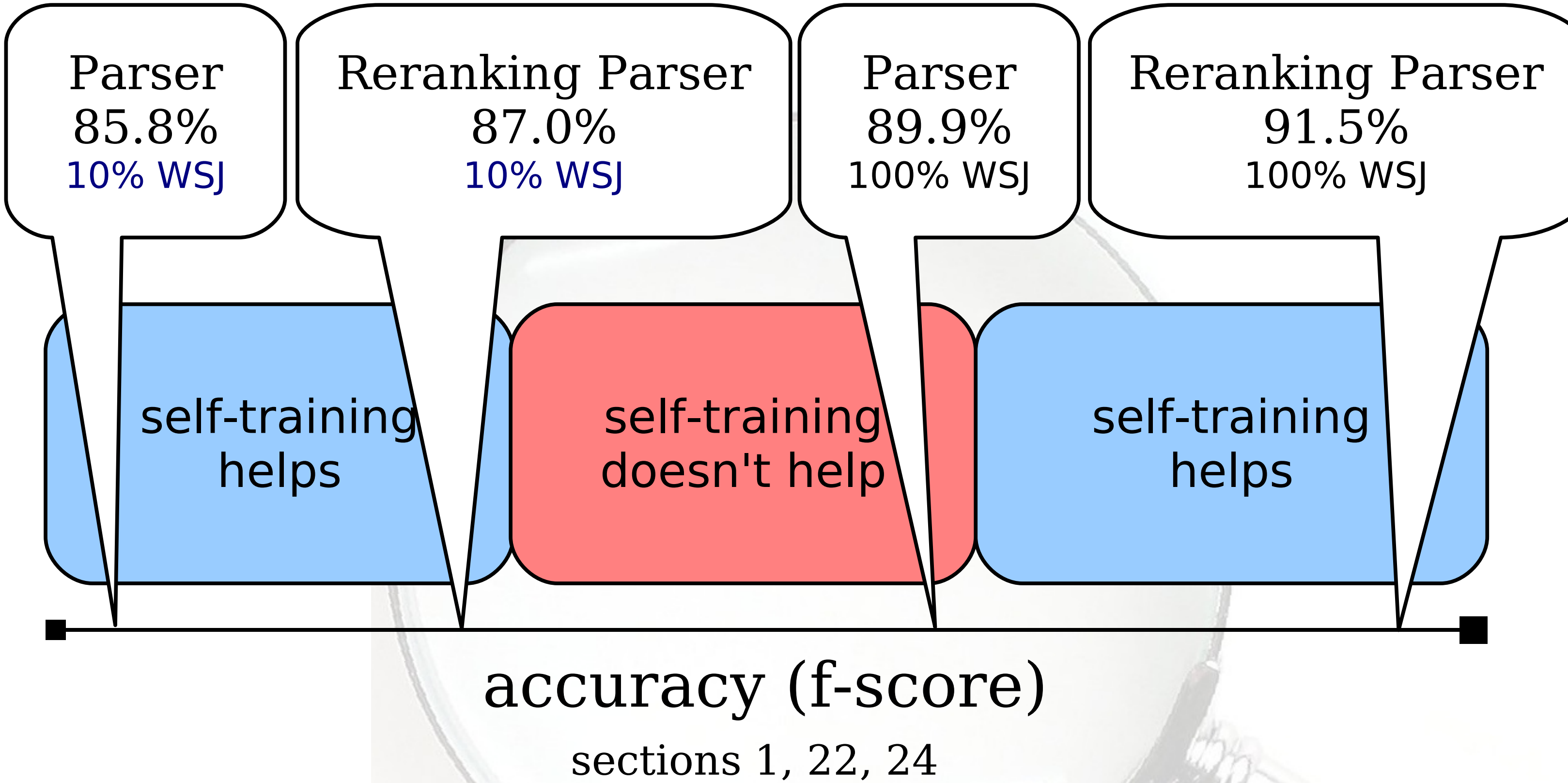


Phase Transition



➔ There is no phase transition for self-training.

Phase Transition



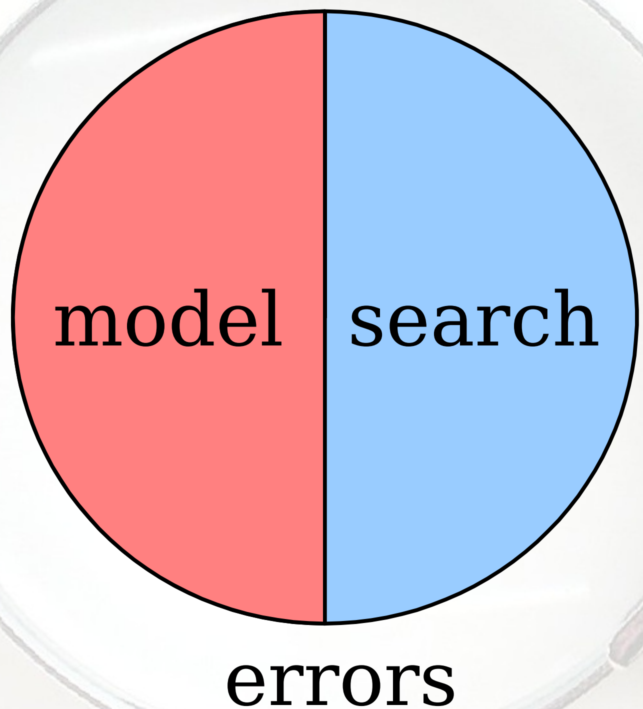
See also: Reichart and Rappoport (2007)

➔ There is no phase transition for self-training.

Search Errors

- Self-trained models have fewer search errors

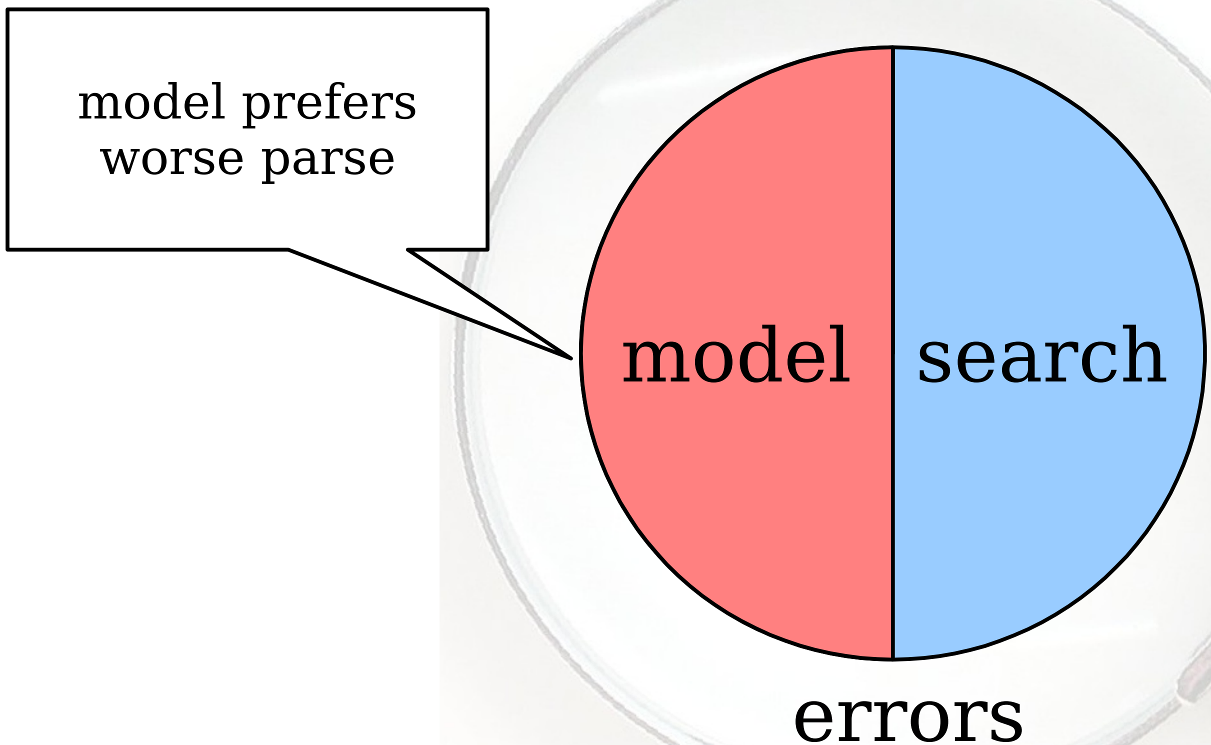
(Daniel Marcu, p.c.)



Search Errors

- Self-trained models have fewer search errors

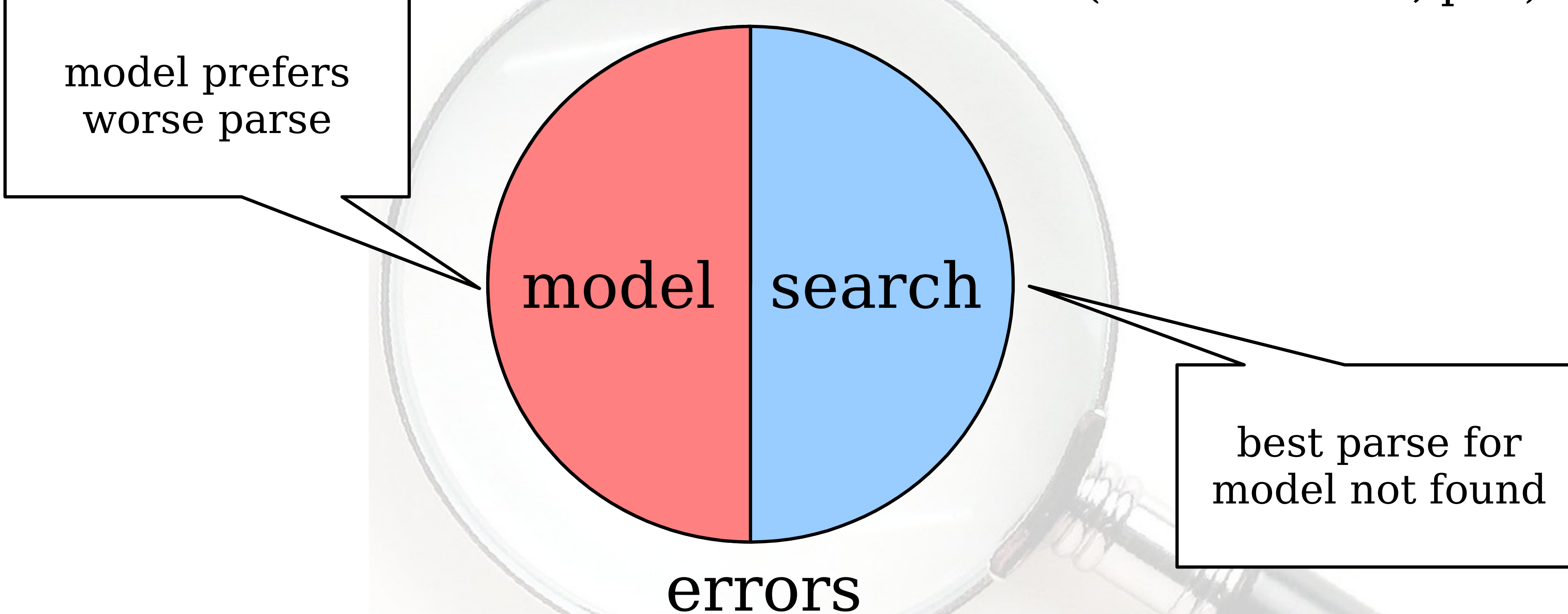
(Daniel Marcu, p.c.)



Search Errors

- Self-trained models have fewer search errors

(Daniel Marcu, p.c.)



Some notation

orig ← parses from original parser

st ← parses from self-trained parser

$\mathrm{top}_{m}(\mathbf{P})$ ← best parse in **P** for reranker model m

(m in **{orig,st}**)

Comparing n-best lists

Overlap of **st** and **orig**	66.0%
$\mathrm{top}_{st}(\mathbf{st}) = \mathrm{top}_{orig}(\mathbf{orig})$	42.4%
$\mathrm{top}_{st}(\mathbf{st})$ in **orig**	60.3%
Search errors	2.5%

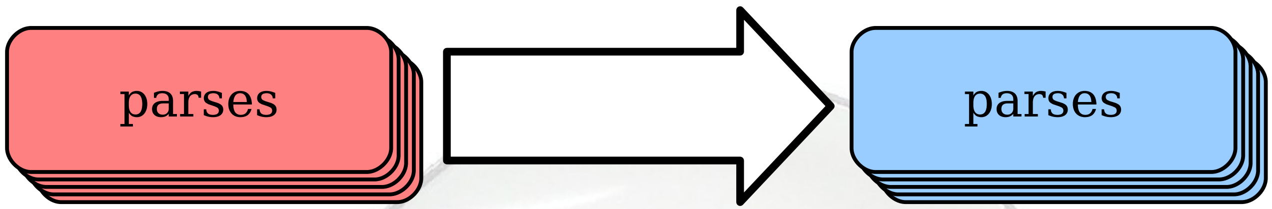
(statistics on 5,039 sentences in sections 1, 22, 24)

- Search errors =

  $\mathrm{top}_{st}(\mathbf{st})$ not in **orig** and

  $\mathrm{top}_{orig}(\mathbf{st} \cup \mathbf{orig}) = \mathrm{top}_{st}(\mathbf{st})$

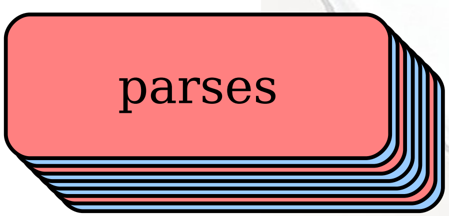
Decreasing Search Errors



self-trained parser

original parser

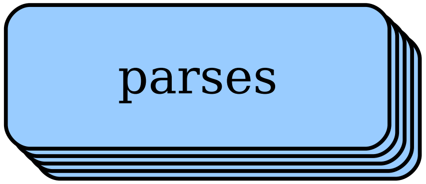
- Add parses from self-trained n-best list to original parser's n-best list, rescoring by original parser



augmented original parser

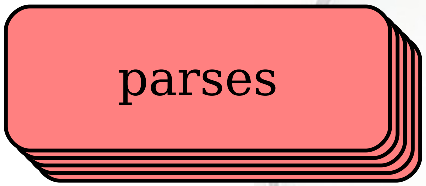
zero probability under original parser

Evaluation



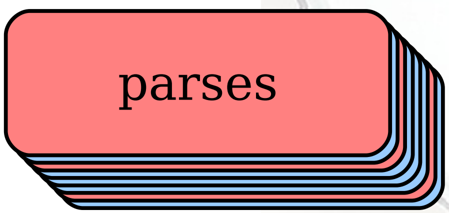
original
reranking parser

91.5%



self-trained
reranking parser

92.0%

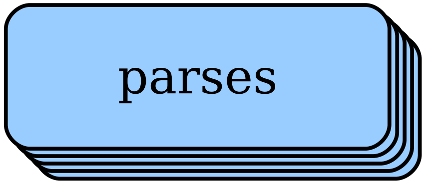


augmented original
reranking parser

91.7%

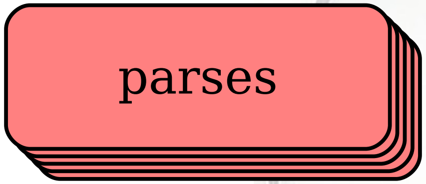
(reranking parser f-score on sections 1, 22, 24)

Evaluation



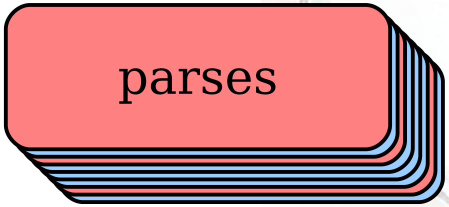
original reranking parser

91.5%



self-trained reranking parser

92.0%



augmented original reranking parser

91.7%

➔ The original parser makes both **model** and **search** errors relative to the self-trained model.

(reranking parser f-score on sections 1, 22, 24)

Reranker features

Non-generative reranker features help self-training more

reranker features
(1.3 million total)

Reranker features

Non-generative reranker features help self-training more

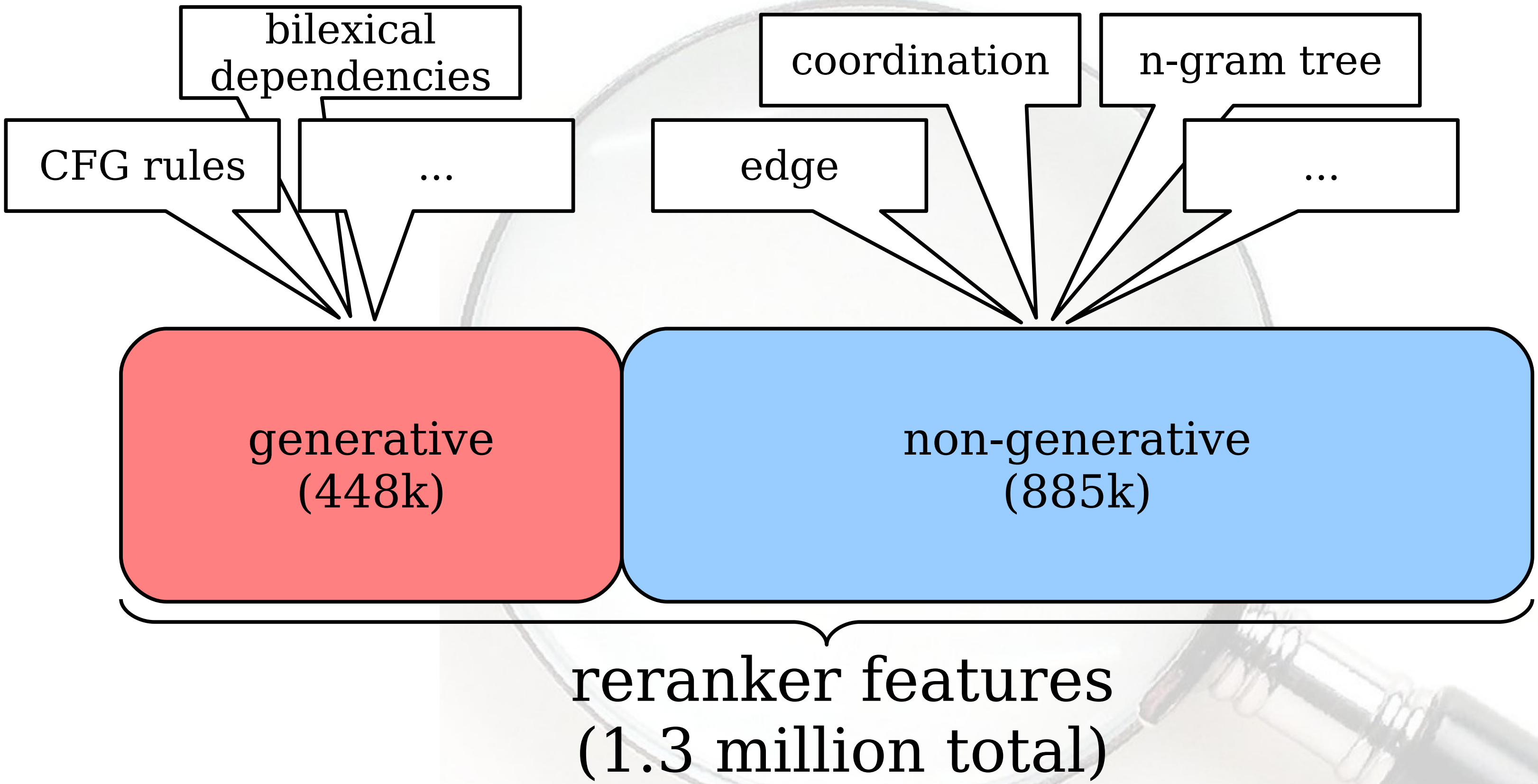
generative
(448k)

non-generative
(885k)

reranker features
(1.3 million total)

Reranker features

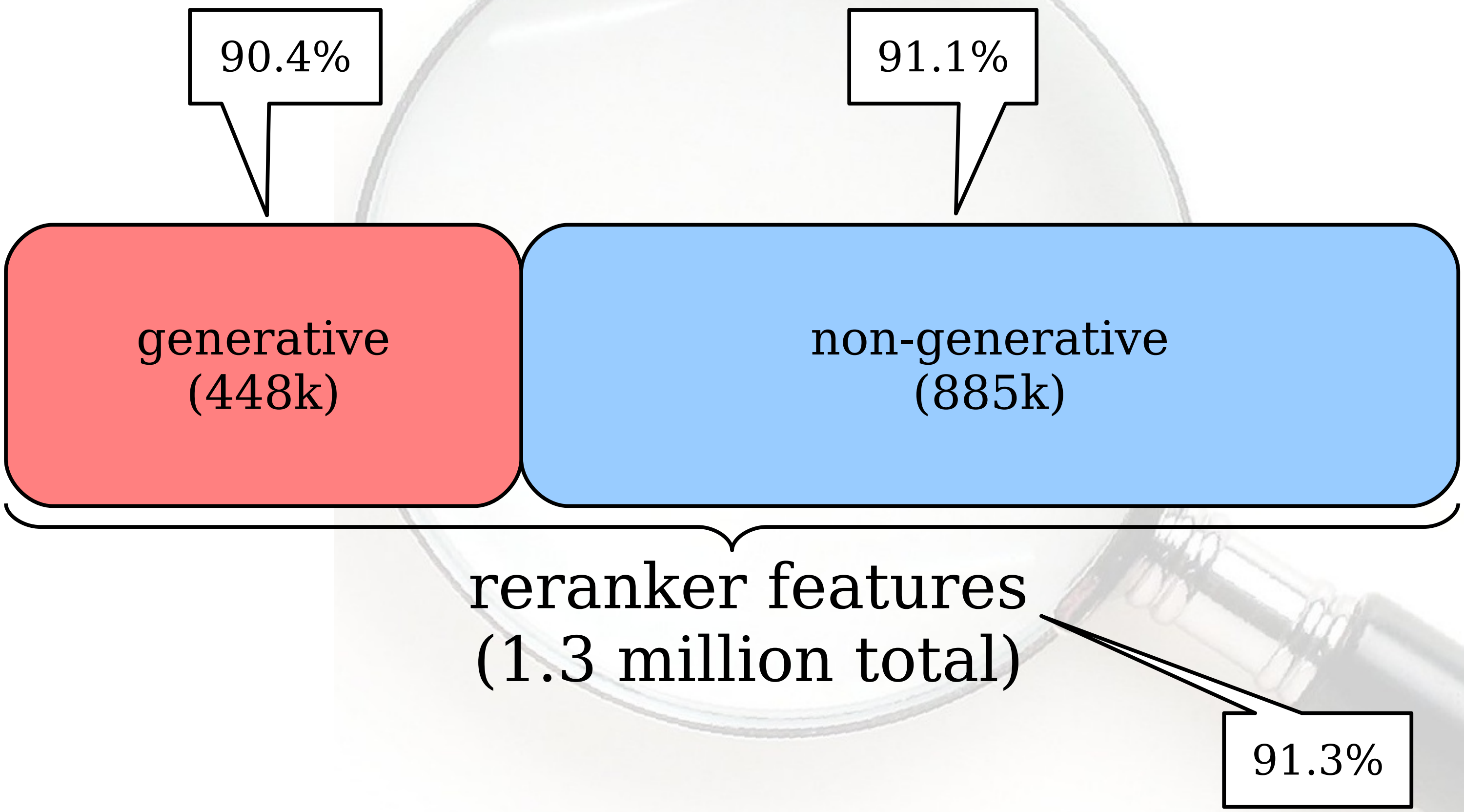
Non-generative reranker features help self-training more



Reranker features

🔍 Non-generative reranker features help self-training more

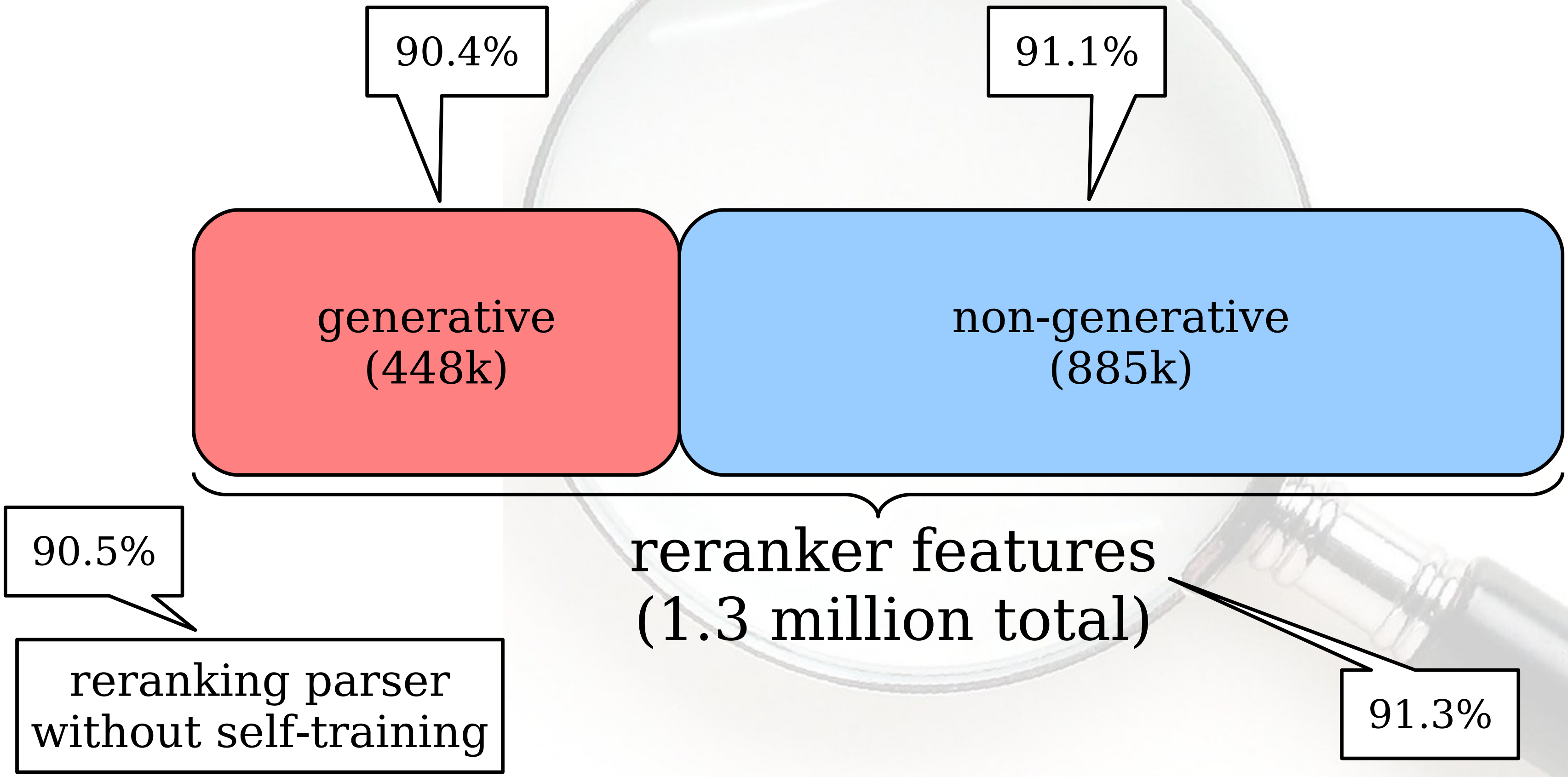
(reranking parser f-score on section 24)



Reranker features

Non-generative reranker features help self-training more

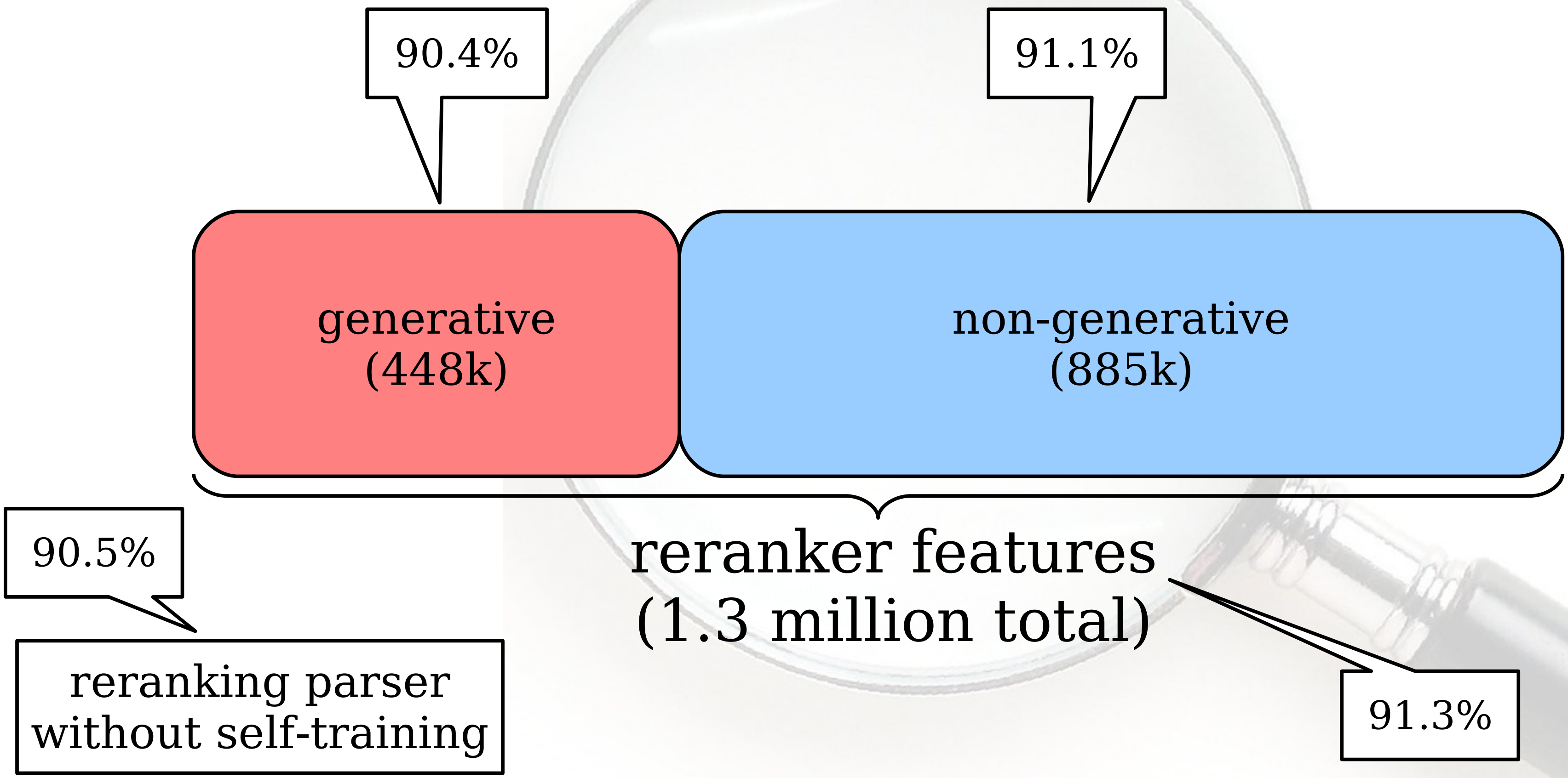
(reranking parser f-score on section 24)



Reranker features

➔ Non-generative reranker features are essential for self-training with a reranker.

(reranking parser f-score on section 24)

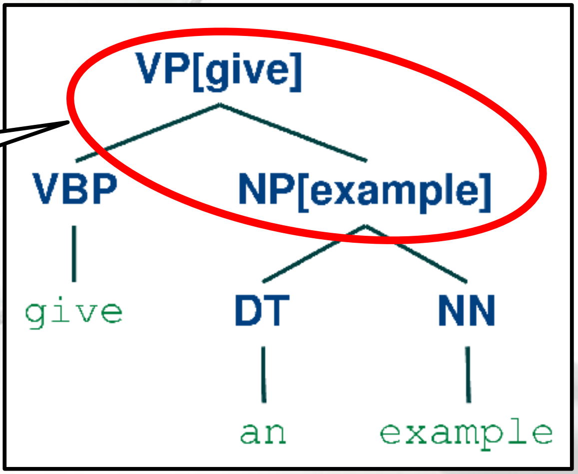


Bilexical Dependencies

Self-training teaches the parser about bilexical dependencies

(Mitch Marcus, p.c.)

bilexical dependency
or “bihead”



Bilexical Dependencies

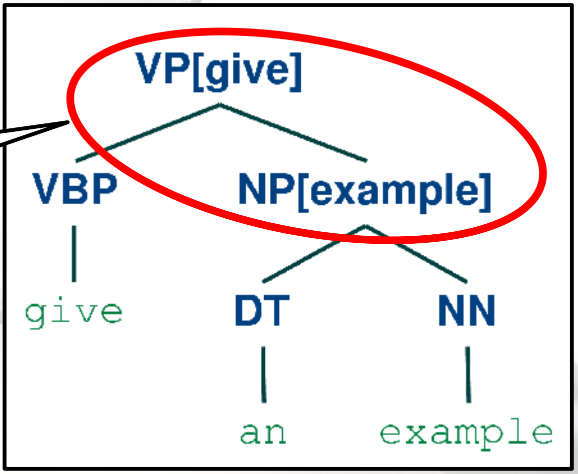
Self-training teaches the parser about bilexical dependencies

(Mitch Marcus, p.c.)

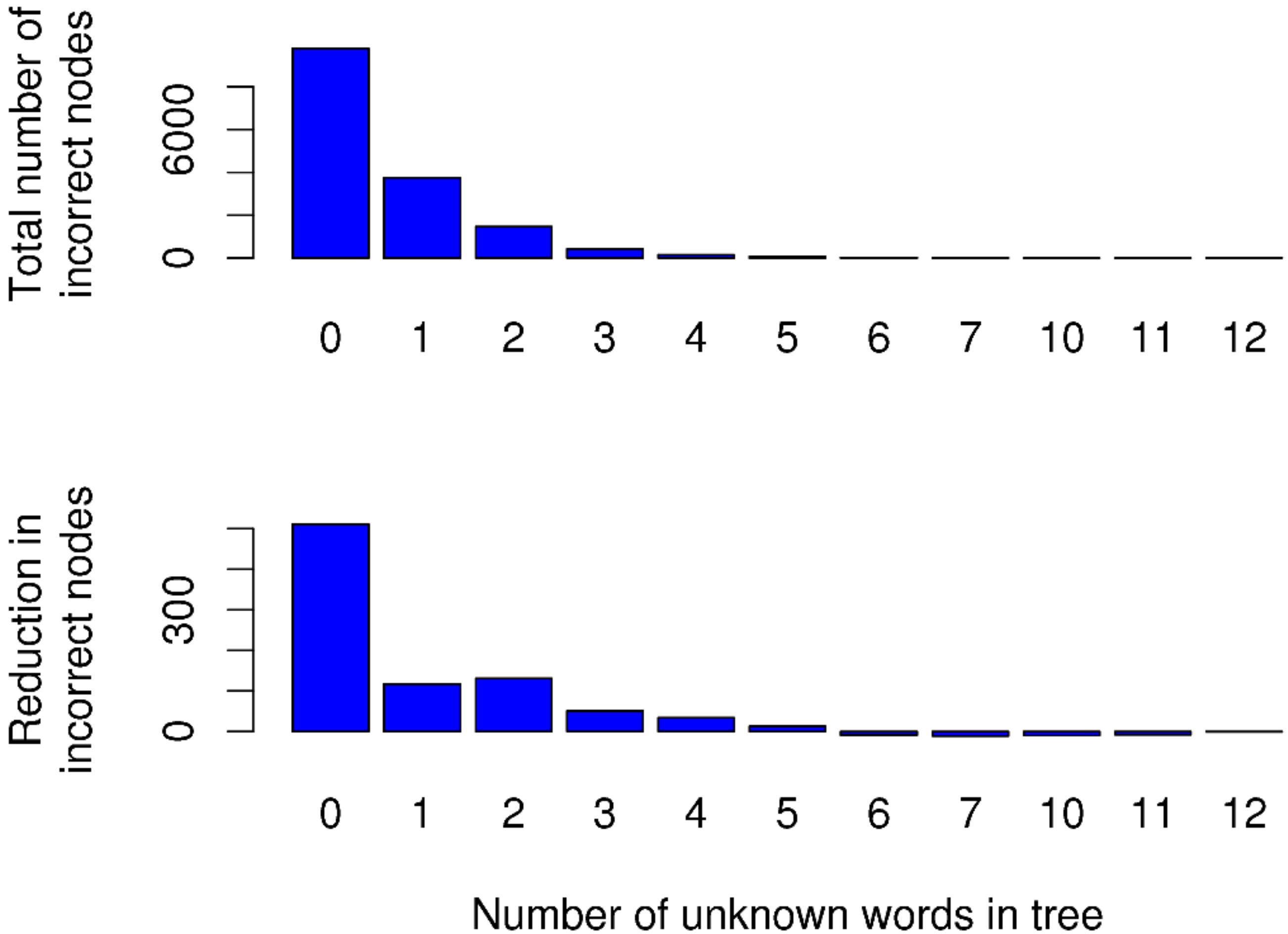
Two ways to test:

- Factor analysis (as in previous work)
- Transfer biheads distribution from self-trained model to original model

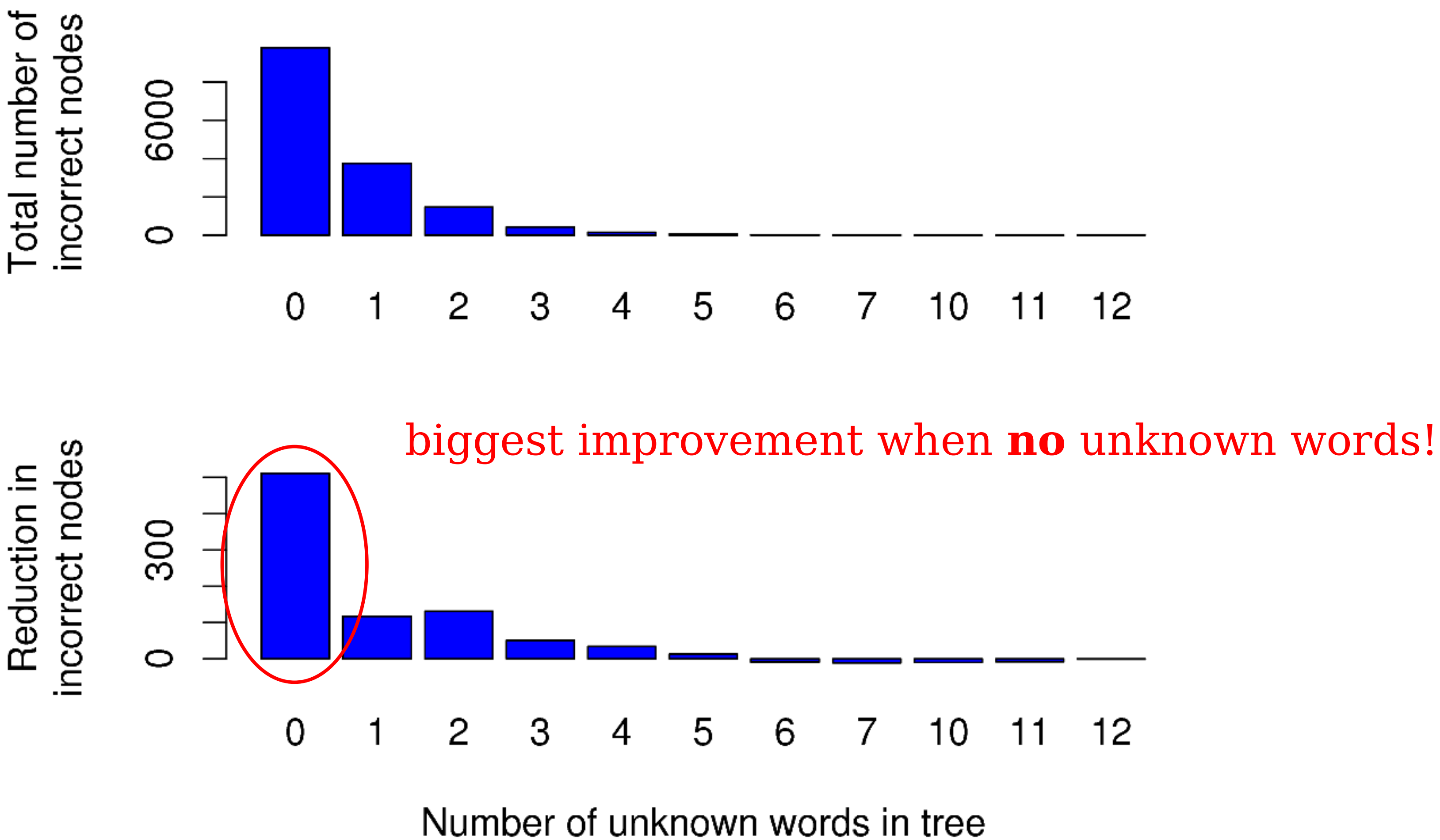
bilexical dependency
or "bihead"



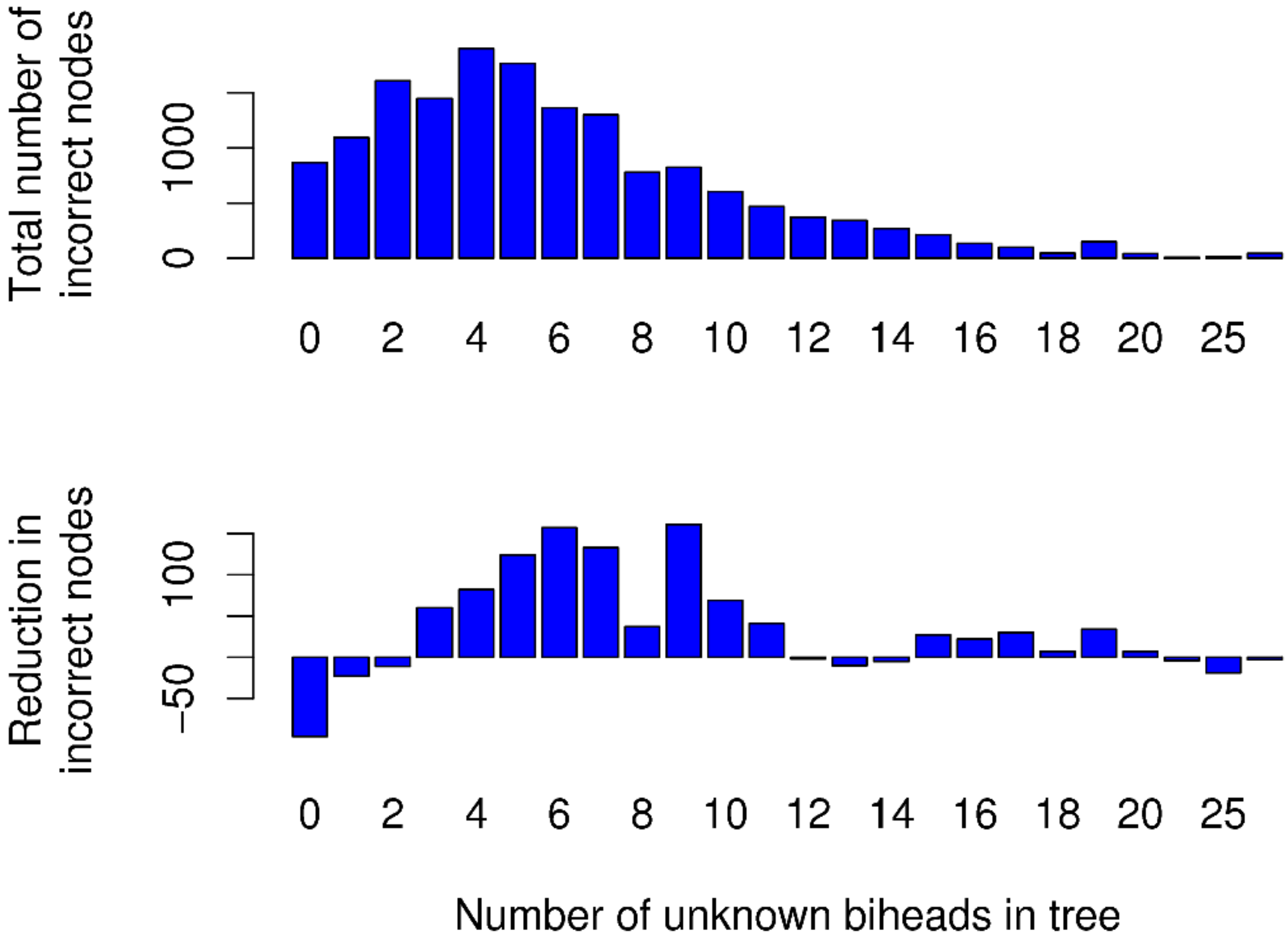
Factor analysis: Words



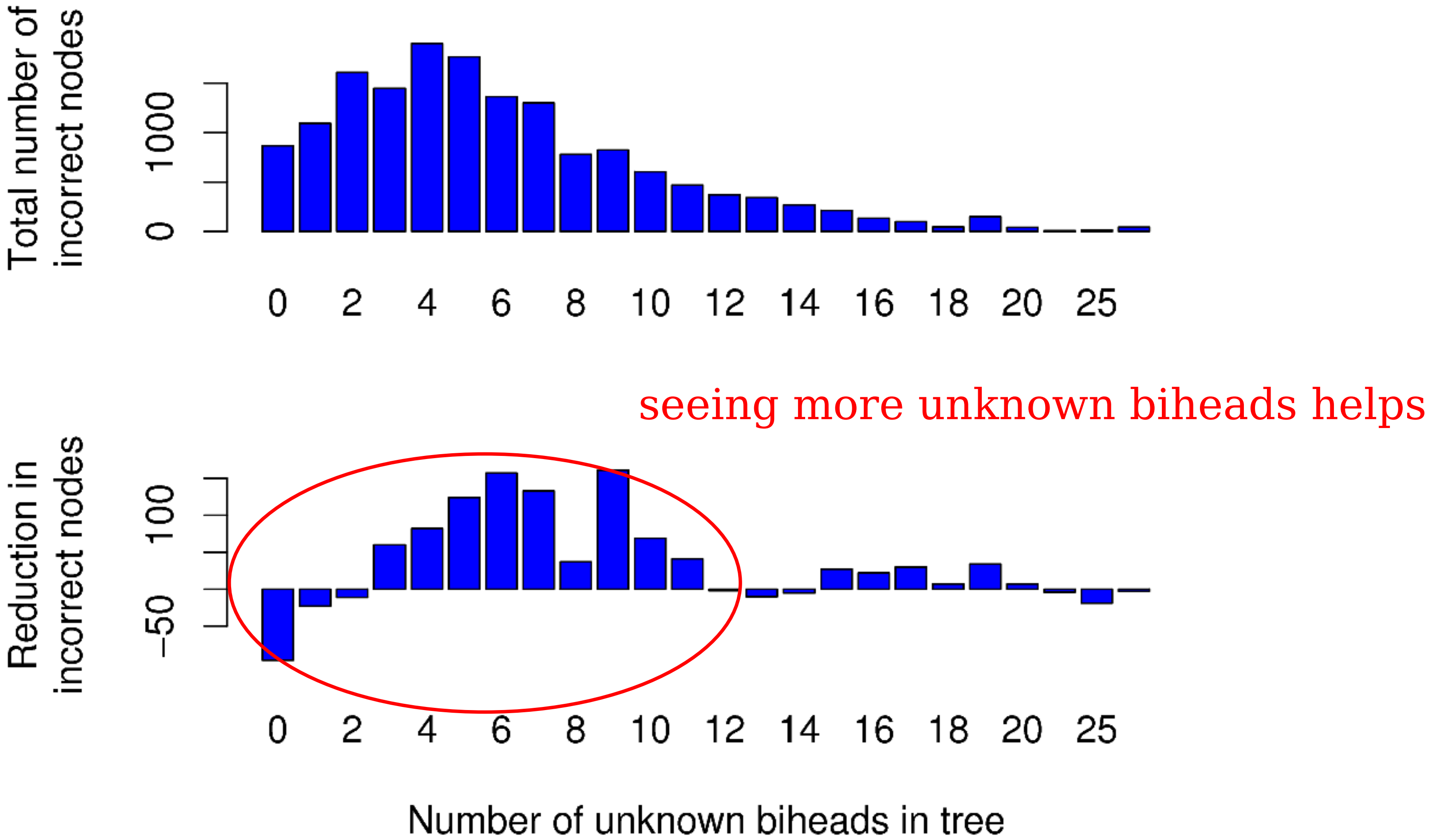
Factor analysis: Words



Factor Analysis: Biheads

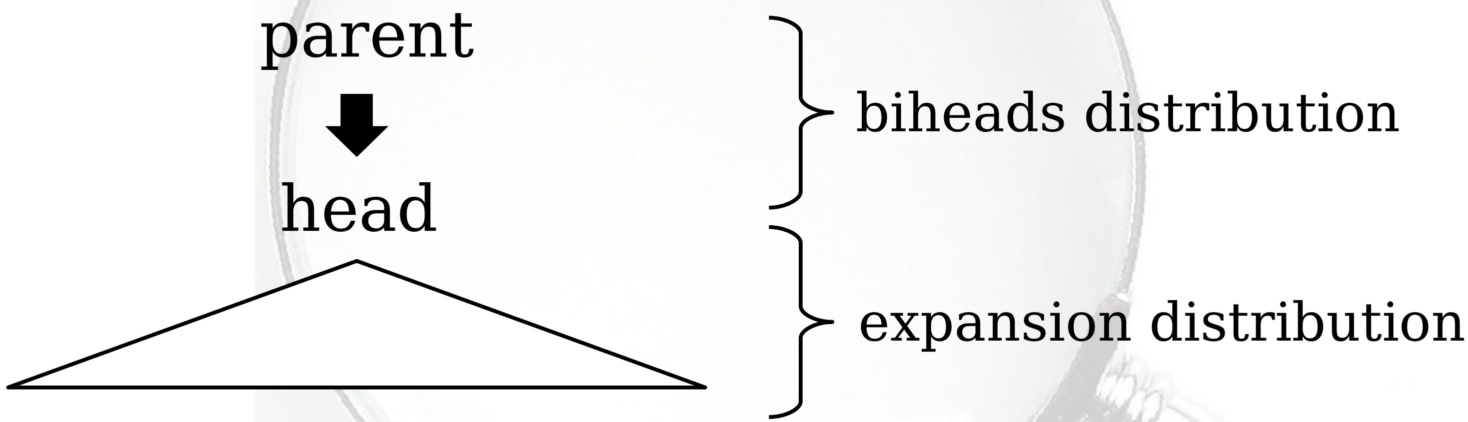


Factor Analysis: Biheads



Just biheads?

- If self-trained model learns more about biheads, can we transfer that knowledge to original model?



dangerously oversimplified Charniak parsing model!

Just biheads?

- If self-trained model learns more about biheads, can we transfer that knowledge to original model?

original model	expansions	biheads	rest	89.8%
self-trained model	expansions	biheads	rest	90.8%

(generative parser f-score on sections 1, 22, 24)

Just biheads?

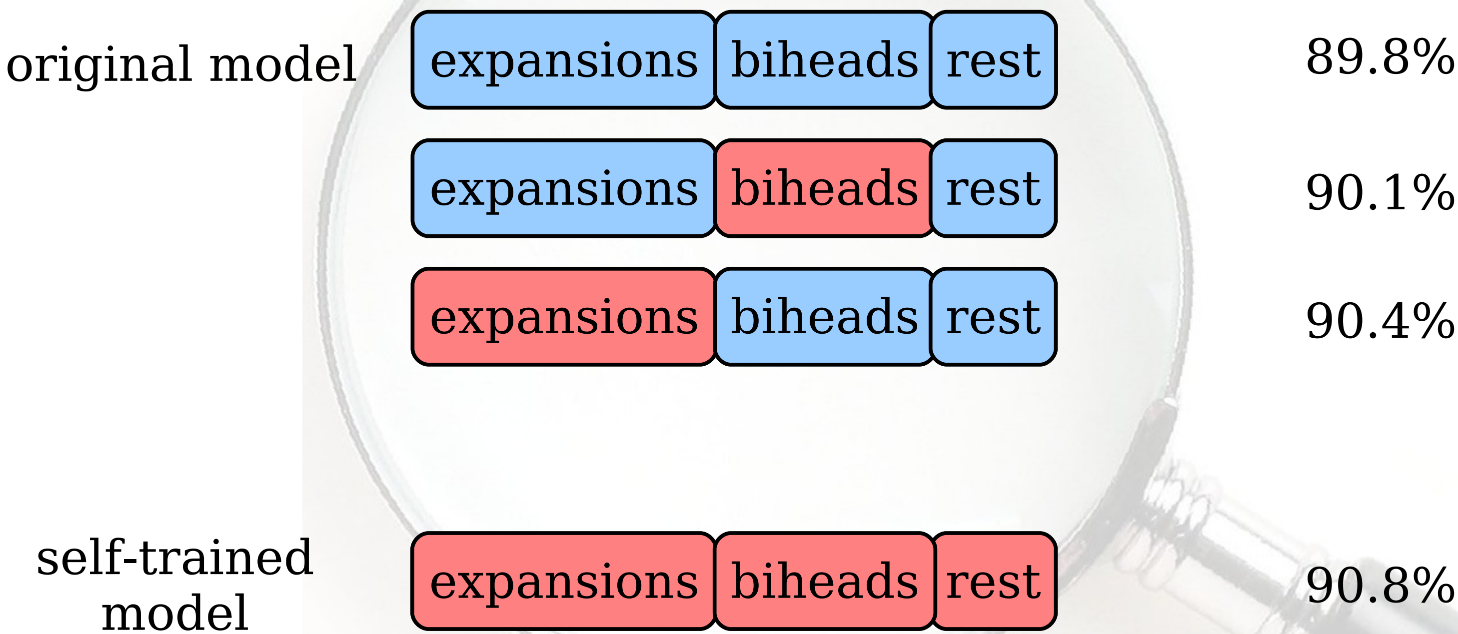
- If self-trained model learns more about biheads, can we transfer that knowledge to original model?

original model	expansions	biheads	rest	89.8%
	expansions	biheads	rest	90.1%
self-trained model	expansions	biheads	rest	90.8%

(generative parser f-score on sections 1, 22, 24)

Just biheads?

- If self-trained model learns more about biheads, can we transfer that knowledge to original model?



(generative parser f-score on sections 1, 22, 24)

Just biheads?

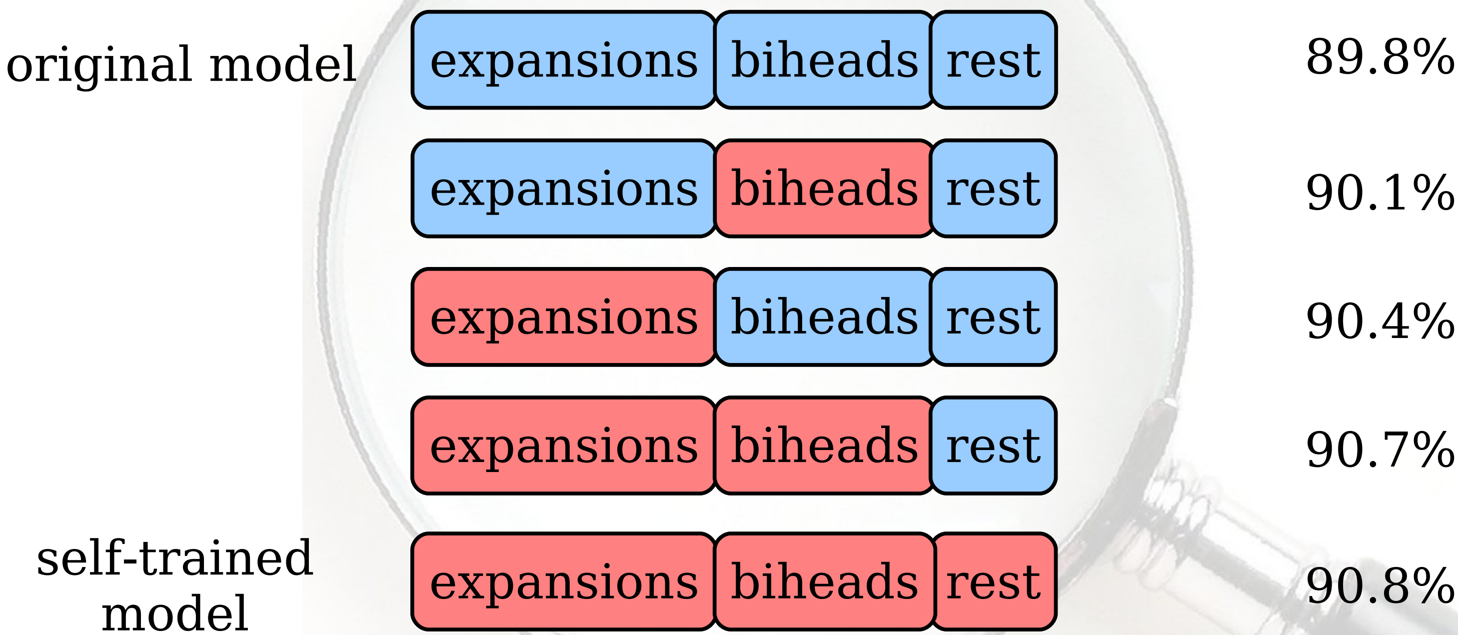
- If self-trained model learns more about biheads, can we transfer that knowledge to original model?

original model	expansions	biheads	rest	89.8%
	expansions	biheads	rest	90.1%
	expansions	biheads	rest	90.4%
	expansions	biheads	rest	90.7%
self-trained model	expansions	biheads	rest	90.8%

(generative parser f-score on sections 1, 22, 24)

Just biheads?

➔ Self-training improves biheads distribution, but **also** expansions distribution.



(generative parser f-score on sections 1, 22, 24)

Outline

- What is self-training?
- Previous work
- Experimental setup
- Four hypotheses
- **Conclusions**

Hypothesis Results

1. Phase Transition
   Disproved.
2. Search Errors
   Reducing search errors helps but model errors remain.
3. Non-generative Reranker Features
   Reranker features must be different from generative parser.
4. Bilexical Dependencies
   Biheads correlated with self-training improvements.
   Self-training helps all parts of the parser, not just biheads.

Two different cases?

	Predictor			Need reranker?
	Length	unk. words	unk. biheads	
Large seed	**+**	**-**	**+**	**+**
Small seed	**+**	**+**	**?**	**-**

A Possible Connection

- **Small seed case:** Vocabulary is sparse so unknown **words** may be resolved in unlabeled text.
- **Large seed case:** Learn new **head information** in unlabeled text. Non-generative features in the reranker needed to handle more complex constructions.

Future Work

- “Bilexical dependencies” experiments with small seed size.
- Different parsers (Collins, LTAG, discriminative, ...) with and without rerankers.
- More hypotheses...
  (Audience participation?)

Acknowledgments

This work was supported by DARPA GALE contract HR0011-06-2-0001.

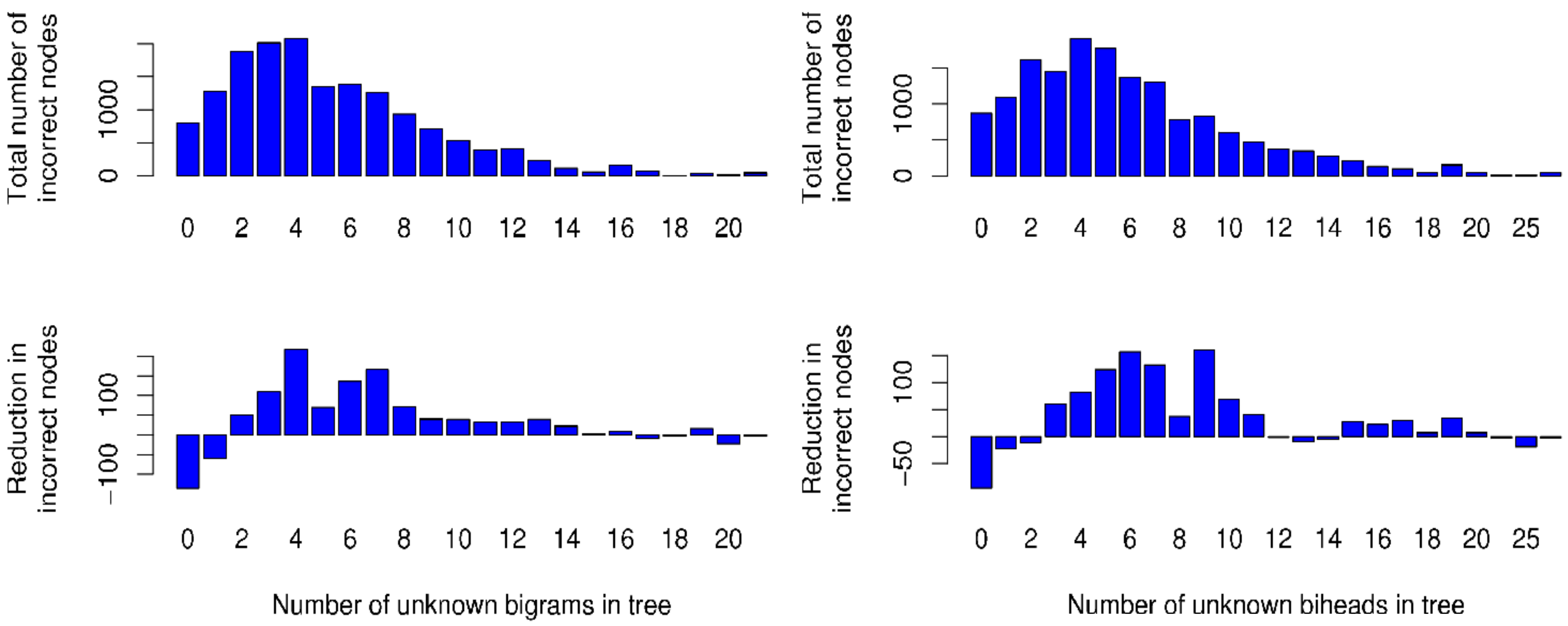
We would like to thank Roi Reichart, Matt Lease and the rest of the BLLIP team, and our anonymous reviewers for their comments.

Questions?

http://bllip.cs.brown.edu/selftraining/

Extra Slides

Bigrams and Biheads



Bigrams and Biheads