Grammatical Error Correction for ESL-Learners by SMT

Getting it right

Marcin Junczys-Dowmunt Roman Grundkiewicz

Information System Laboratory
Faculty for Math and Computer Science
Adam Mickiewicz University in Poznań, Poland

January 5th, 2015

Part I

What's been going on in with SMT in Grammatical Error Correction (GEC)?

The CoNLL-2014 Shared Task

- Focus: English as a second language (ESL);
- Preceeded by the HOO 2011 and 2012 (Helping Our Own) shared tasks: detection and correction of determiner and preposition errors in scientific papers;
- Preceeded by the CoNLL-2013 shared task on Grammatical Error Correction: correct five error types, determiners, prepositions, noun number, subject-verb-agreement, verb forms. 16 teams.
- This year, 28 error categories.
- 13 teams participated.

NUCLE: NUS Corpus of Learner English

Dahlmeier et. al 2013

- National University of Singapore (NUS) Natural Language Processing Group led by Prof. Hwee Tou Ng and the NUS Centre for English Language Communication led by Prof. Siew Mei Wu;
- 1,400 essays, written by NUS students, ca. 1 million words, ca. 51,000 sentences;
- Topics: environmental pollution, healthcare, etc.;
- Annotated with 28 error categories by NUS English instructors.

NUCLE: NUS Corpus of Learner English

Example

```
<DOC nid="840">
<TEXT>
<P>Engineering design process can be defined ... </P>
<P>Firstly, engineering design ... </P>
...
</TEXT>
<ANNOTATION teacher_id="173">
<MISTAKE start_par="0" start_off="0" end_par="0" end_off="26">
<TYPE>ArtOrDet</TYPE>
<CORRECTION>The engineering design process</CORRECTION>
</MISTAKE>
...
</ANNOTATION>
</DOC>
```

NUCLE: NUS Corpus of Learner English

NUCLE Error Types I

*	6647	ArtOrDet	Article or Determiner
	5300	Wci	Wrong collocation/idiom
	4668	Rloc-	Local redundancy
*	3770	Nn	Noun number
	3200	Vt	Verb tense
	3054	Mec	Punctuation, spelling
*	2412	Prep	Preposition
	2160	Wform	Word form
*	1527	SVA	Subject-verb-agreement
	1452	Others	Other errors
*	1444	Vform	Verb form
	1349	Trans	Link word/phrases
	1073	Um	Unclear meaning
	925	Pref	Pronoun reference

NUCLE: NUS Corpus of Learner English

NUCLE Error Types II

861	Srun	Runons, comma splice
674	WOinc	Incorrect sentence form
571	Wtone	Tone
544	Cit	Citation
515	Spar	Parallelism
431	Vm	Verb modal
411	V0	Missing verb
353	Ssub	Subordinate clause
345	WOadv	Adverb/adjective position
242	Npos	Noun possesive
186	Pform	Pronoun form
174	Sfrag	Fragment
50	Wa	Acronyms
47	Smod	Dangling modifier

Evaluation Metric: MaxMatch (M^2)

Dahlmeier and Ng 2012

- An algorithm for efficiently computing the sequence of phrase-level edits between a source sentence and a hypothesis.
- Finds the maximum overlap with the gold-standard.
- Based on Levenshtein distance matrix for candidate and reference sentence.
- The optimal edit sequence is scored using $F_{1.0}$ (CoNLL 2013) or $F_{0.5}$ measure (CoNLL 2014).

Evaluation Metric: MaxMatch (M^2)

Example calculation

```
S The cat sat at mat .
A 3 4|||Prep|||on|||REQUIRED|||-NONE-|||0
A 4 4|||ArtOrDet|||the||a|||REQUIRED|||-NONE-|||0

S The dog .
A 1 2|||NN|||dogs|||REQUIRED|||-NONE-|||0

S Giant otters is an apex predator .
A 2 3|||SVA|||are|||REQUIRED|||-NONE-|||0
A 3 4|||ArtOrDet|||-NONE-|||REQUIRED|||-NONE-|||0
A 5 6|||NN|||predators|||REQUIRED|||-NONE-|||0
A 1 2|||NN|||otter|||REQUIRED|||-NONE-|||1

A cat sat on the mat .
The dog .
Giant otters are apex predator .
```

The CoNLL-2014 Shared Task

Final Results and Ranking

	Team	P	R	$M^2_{0.5}$
1	CAMB	39.71	30.10	37.33
2	CUUI	41.78	24.88	36.79
3	AMU	41.62	21.40	35.01
4	POST	34.51	21.73	30.88
5	NTHU	35.08	18.85	29.92
6	RAC	33.14	14.99	26.68
7	UMC	31.27	14.46	25.37
8	PKU	32.21	13.65	25.32
9	NARA	21.57	29.38	22.78
10	SJTU	30.11	5.10	15.19
11	UFC	70.00	1.72	7.84
12	IPN	11.28	2.85	7.09
13	IITB	30.77	1.39	5.90

Grammatical Error Correction by Data-Intensive and Feature-Rich Statistical Machine Translation

Marcin Junczys-Dowmunt, Roman Grundkiewicz. CoNLL-2014 Shared Task

- Grammatical error correction seen as translation from erroneous into corrected sentences.
- Baseline is a standard phrase-based Moses set-up.
- Explore the interaction of:
  - Parameter optimization
  - Web-scale language models
  - Larger error-annoted resources as parallel data
  - Task-specific dense and sparse features
- Data used:
  - TM: NUCLE, Lang-8 (scraped)
  - LM: TM data, Wikipedia, CommonCrawl (Buck et. al 2014)

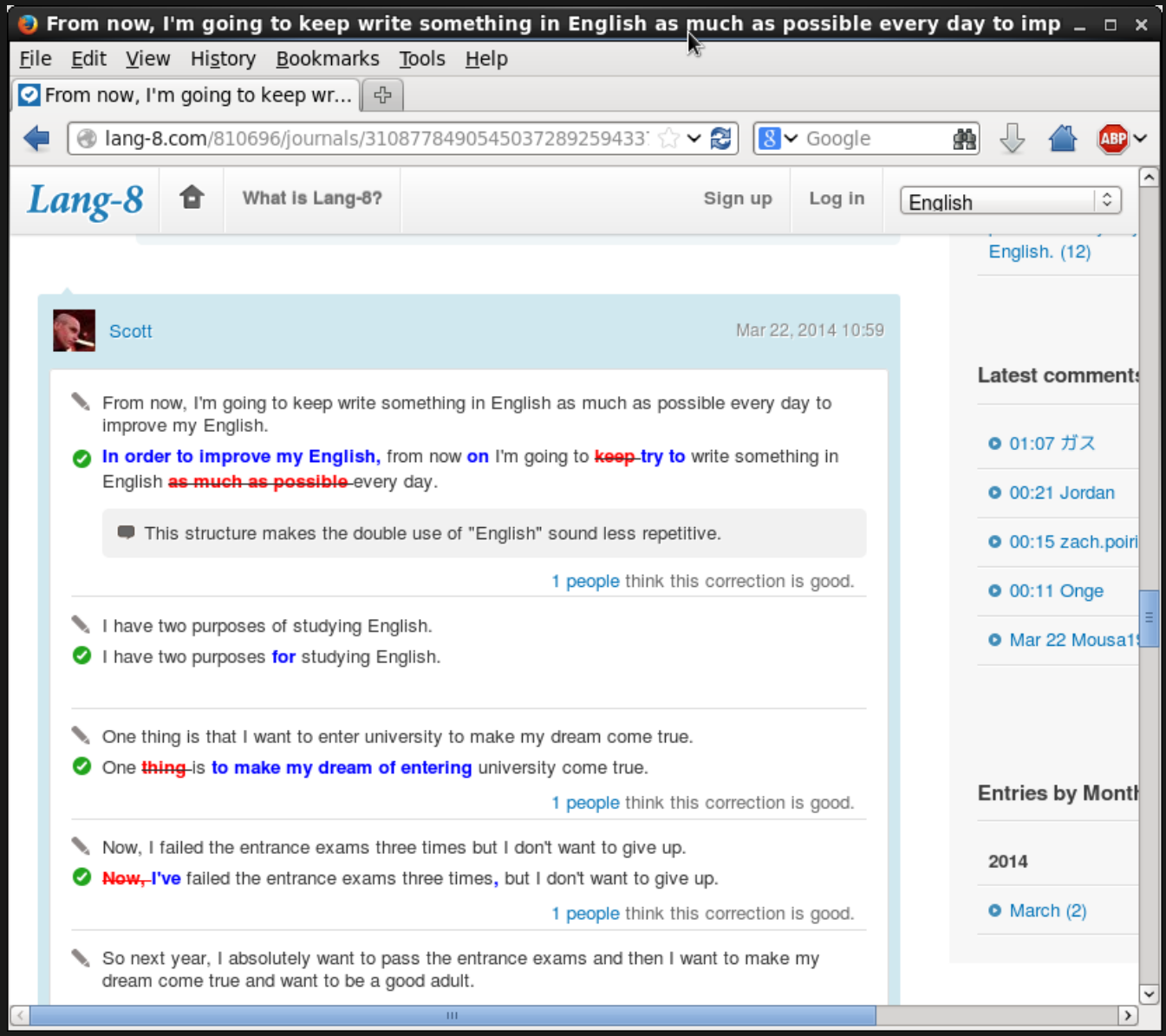
System Combination for Grammatical Error Correction

Raymond Hendy Susanto, Peter Phandi, Hwee Tou Ng. EMNLP 2014

- Uses MEMT to combine several GEC systems:
  - Two classifier-based systems and two SMT-based systems.
  - Various n-top combinations from the CoNLL-2014 shared task.
- Data used:
  - Classifiers: POS-Taggers, Chunkers, etc.
  - SMT-TM: NUCLE, Lang-8 (Mizumoto et al. 2012)
  - SMT-LM: TM data, Wikipedia
  - System Combination: all CoNLL-2014 submissions

Lang-8.com

A social network for language learners



Lang-8.com

Examples

I thought/think that they are/there is a good combination between winter and reading .

Today I have/had a bad began/beginning .

I wanna improve my skill/skills !

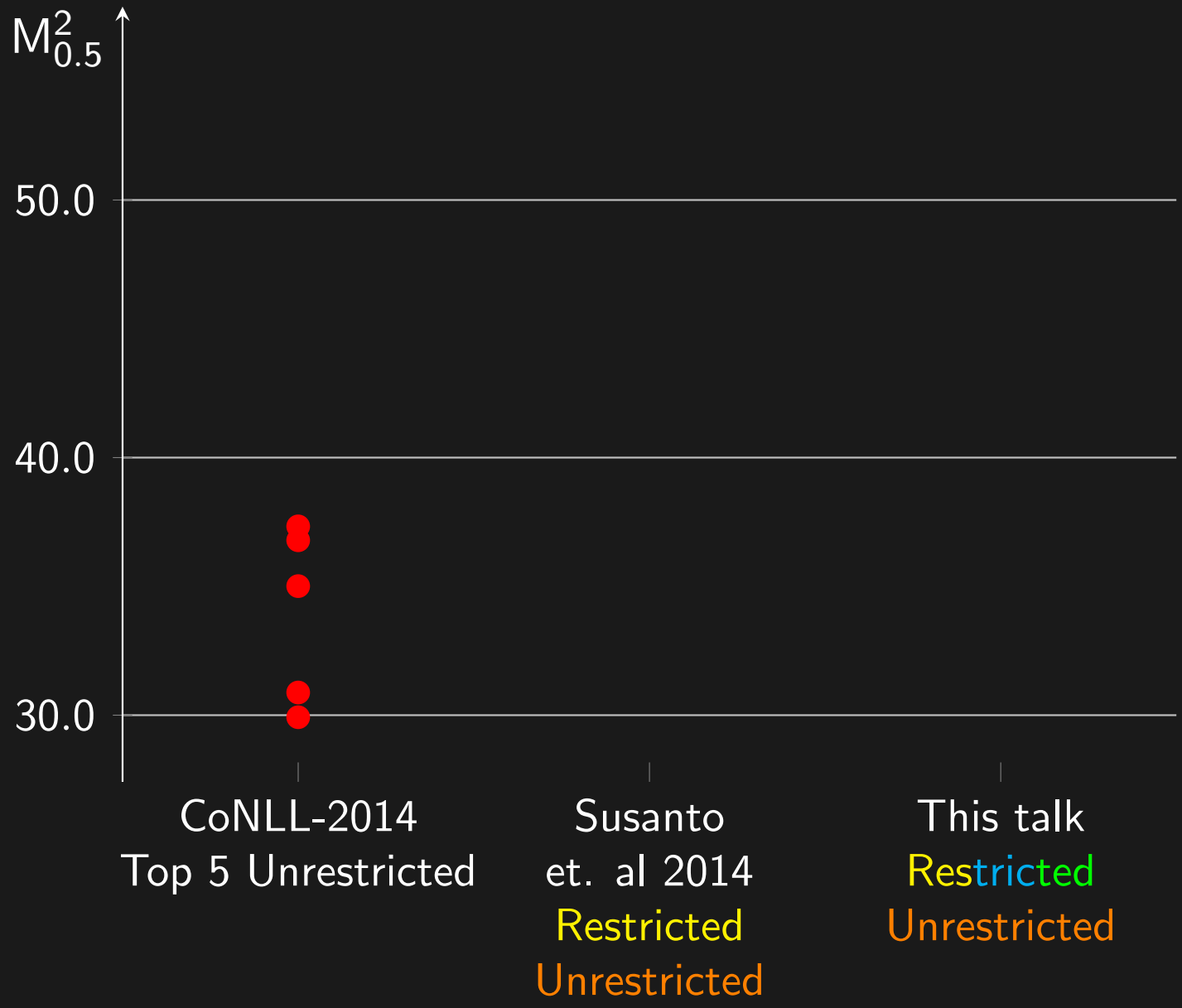
Is there somebody/anybody who wants to be friend/friends with me ?

If you need more information , please don 't hesitate to tell/contact me please .

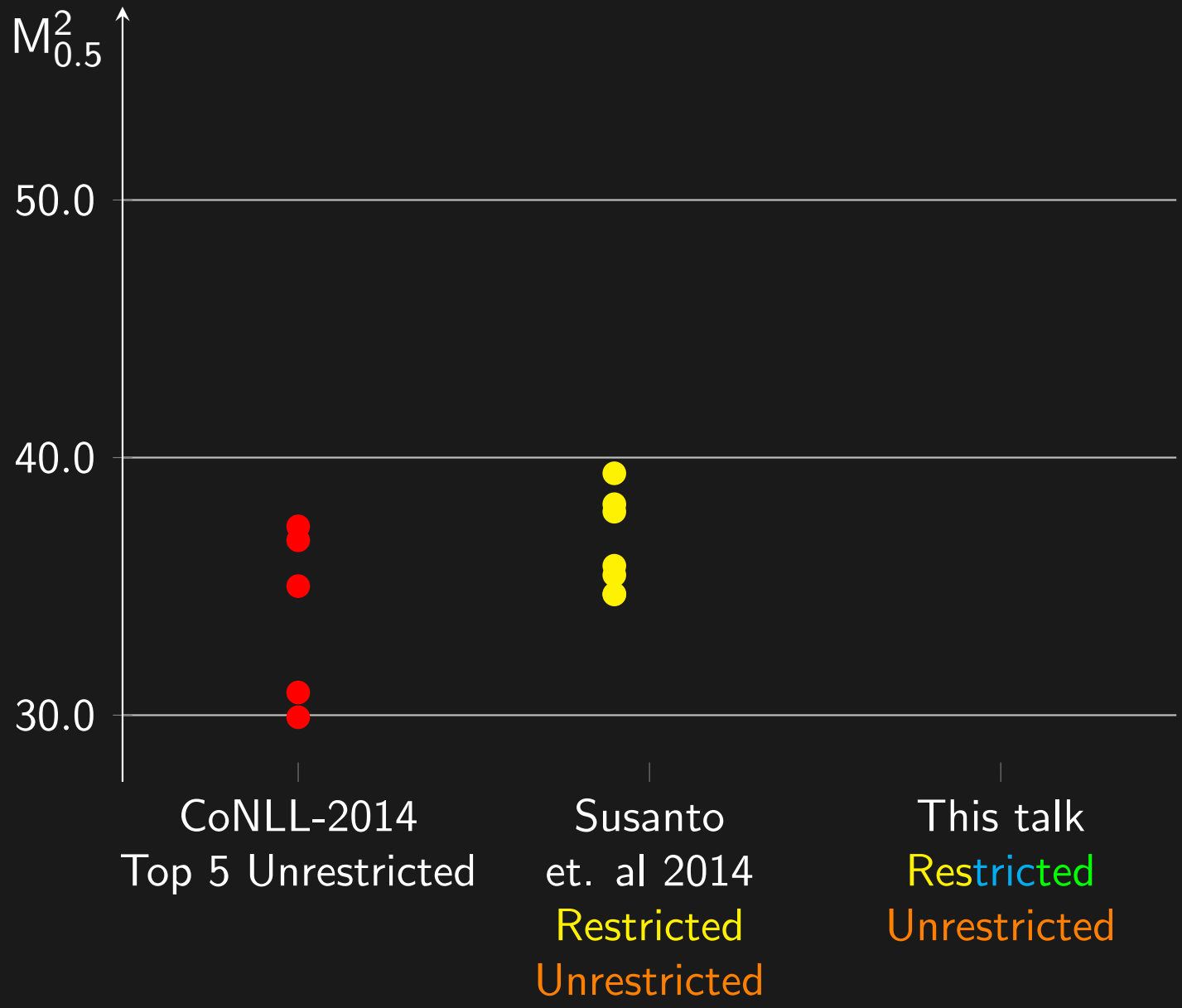
Language model data

Corpus	Sentences	Tokens
NUCLE	57.15 K	1.15 M
Lang-8	2.23 M	30.03 M
Wikipedia	213.08 M	3.37 G
CommonCrawl	59.13 G	975.63 G

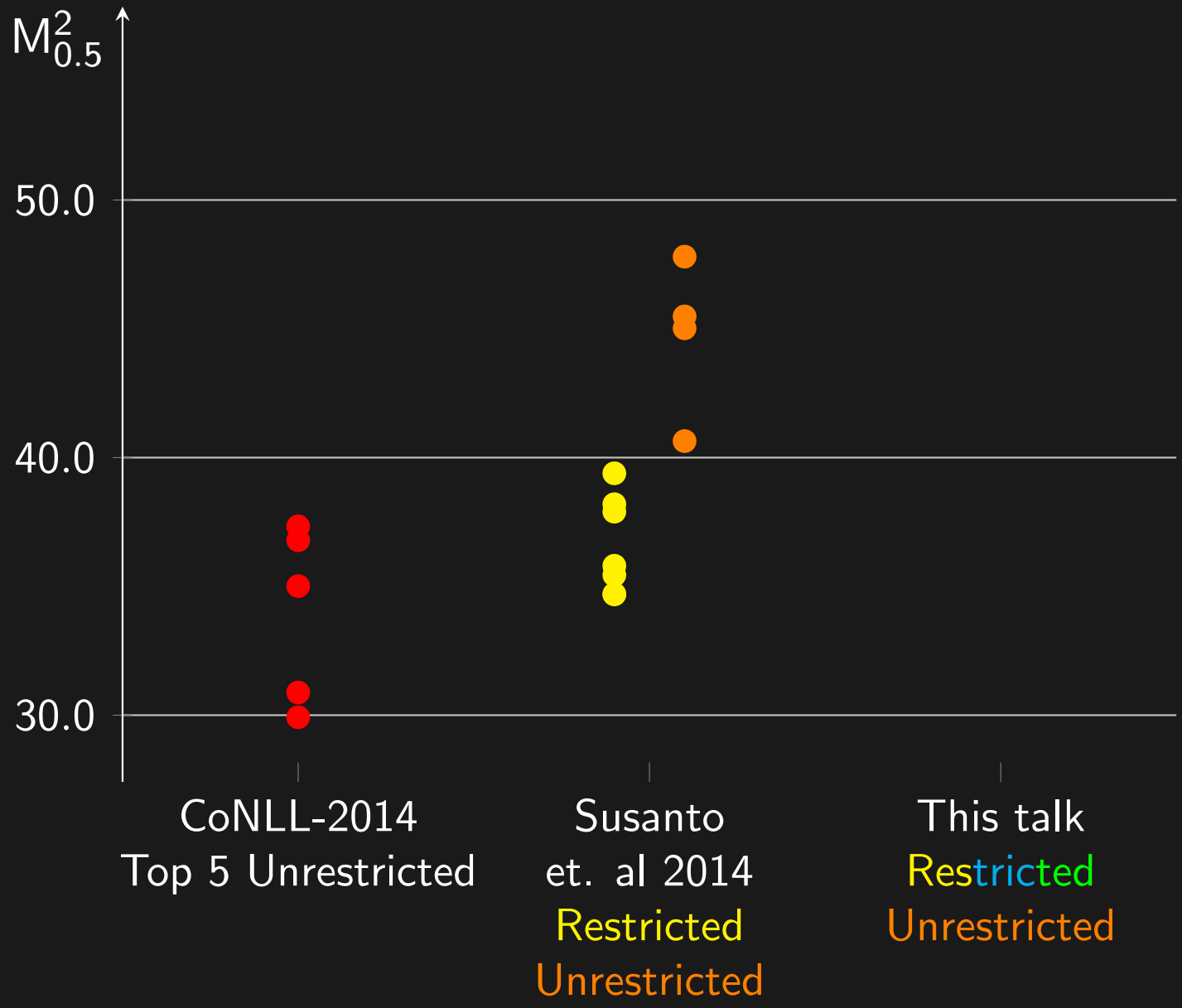
Performance on the CoNLL-2014 Shared Task test set



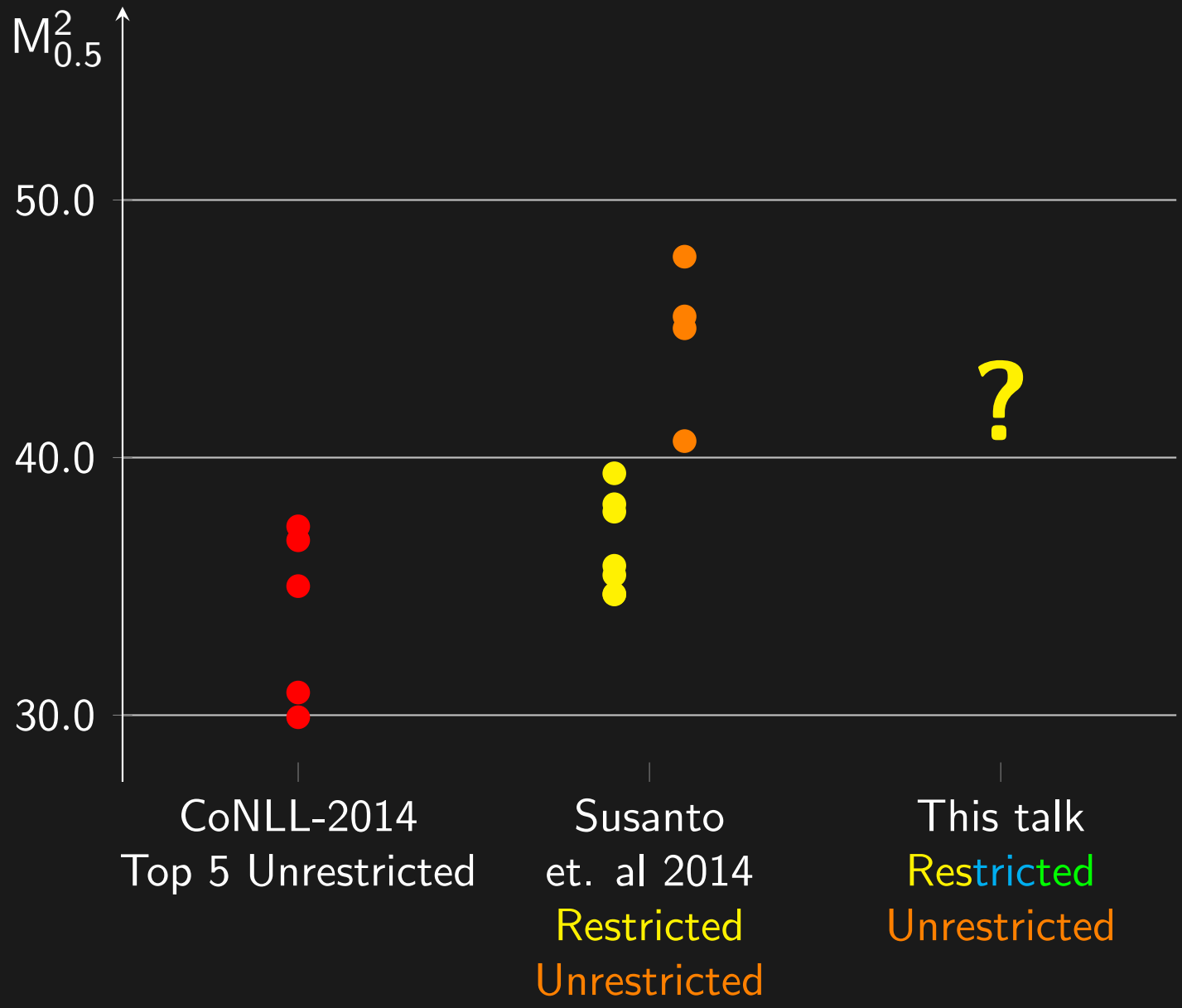
Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Part II

10 Lessons from the CoNLL-2014 Shared Task

Lesson 1: Implement task-specific features

Add features that seem to be relevant for GEC: Levenshtein distance

source phrase (s)	target phrase (t)	LD
a short time .	short term only .	3
a situation	into a situation	1
a supermarket .	a supermarket .	0
a supermarket .	at a supermarket	1
able	unable	1

- $LD(s, t) \equiv$ Levenshtein Distance between source phrase (s) and target phrase (t) in words;
- Feature computes $e^{LD(s,t)}$, sums to total number of edits applied to sentence in log-linear model.

Lesson 1: Implement task-specific features

Add features that seem to be relevant for GEC: Levenshtein distance

source phrase (s)	target phrase (t)	LD	D	I	S
a short time .	short term only .	3	1	1	1
a situation	into a situation	1	0	1	0
a supermarket .	a supermarket .	0	0	0	0
a supermarket .	at a supermarket	1	1	0	0
able	unable	1	0	0	1

- $LD(s, t) \equiv$ Levenshtein Distance between source phrase (s) and target phrase (t) in words;
- Feature computes $e^{LD(s,t)}$, sums to total number of edits applied to sentence in log-linear model.
- From the Levenshtein distance matrix, compute counts for deletions (D), inserts (I), and substitutions (S). e^D, e^I, and e^S are additive in log-linear models.

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_,

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a _TRANS_new_TO_new

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a
_TRANS_new_TO_new _TRANS_problem_TO_problem

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a
_TRANS_new_TO_new _TRANS_problem_TO_problem
_TRANS_surfaces_TO_comes

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a
_TRANS_new_TO_new _TRANS_problem_TO_problem
_TRANS_surfaces_TO_comes _INS_out

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a
_TRANS_new_TO_new _TRANS_problem_TO_problem
_TRANS_surfaces_TO_comes _INS_out _TRANS_._TO_.

Lesson 1: Implement task-specific features

A stateful feature: Operation Sequence Model

Target: Hence , a new problem surfaces .

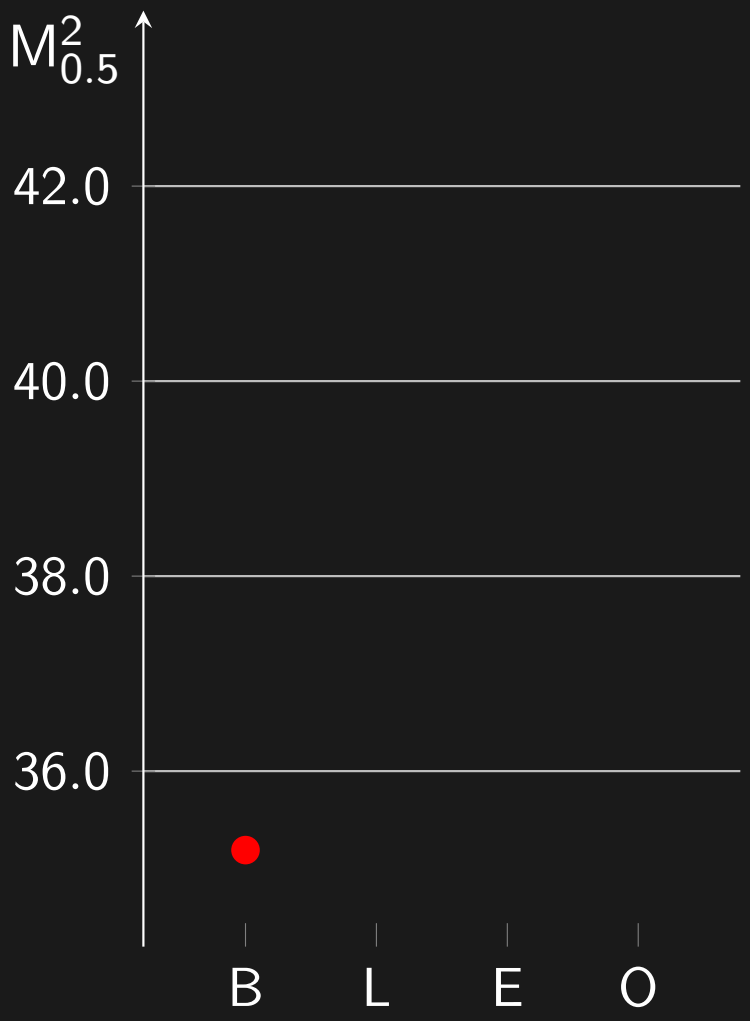
Source: Then a new problem comes out .

TRANS_Hence_TO_Then _DEL_, _TRANS_a_TO_a
_TRANS_new_TO_new _TRANS_problem_TO_problem
_TRANS_surfaces_TO_comes _INS_out _TRANS_._TO_.

- Create sequences for all sentence pairs.
- Compute n-gram language model over operation sequences.
- OSM usually contains reordering operations, here degenerates to edit sequences.

Lesson 1: Implement task-specific features

Results optmized with BLEU

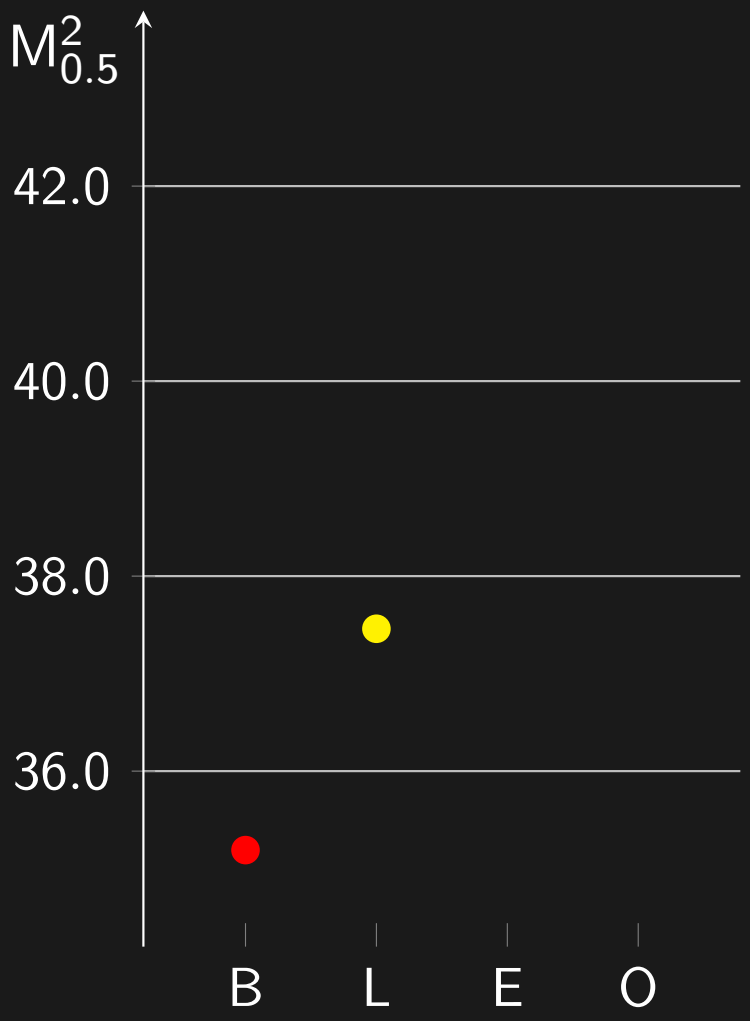


Optimized with BLEU

- (B)aseline: vanilla Moses, disabled reordering.

Lesson 1: Implement task-specific features

Results optmized with BLEU

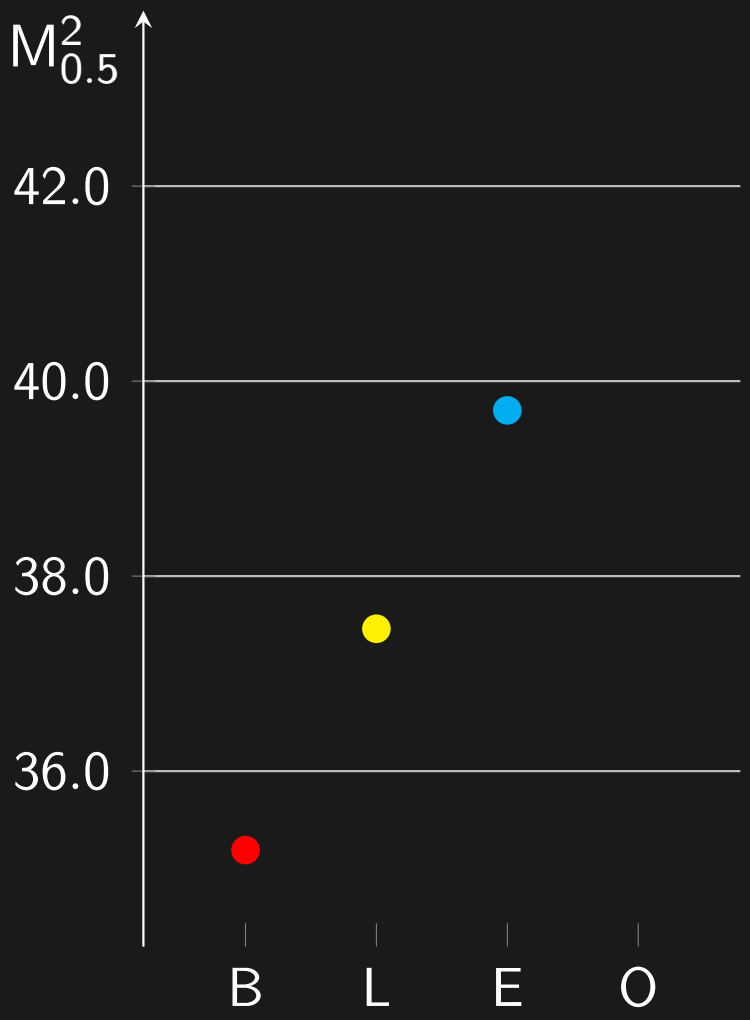


Optimized with BLEU

- (B)aseline: vanilla Moses, disabled reordering.
- (L)evenshtein Distance: word-based as TM score. Sums to number of edits.

Lesson 1: Implement task-specific features

Results optmized with BLEU

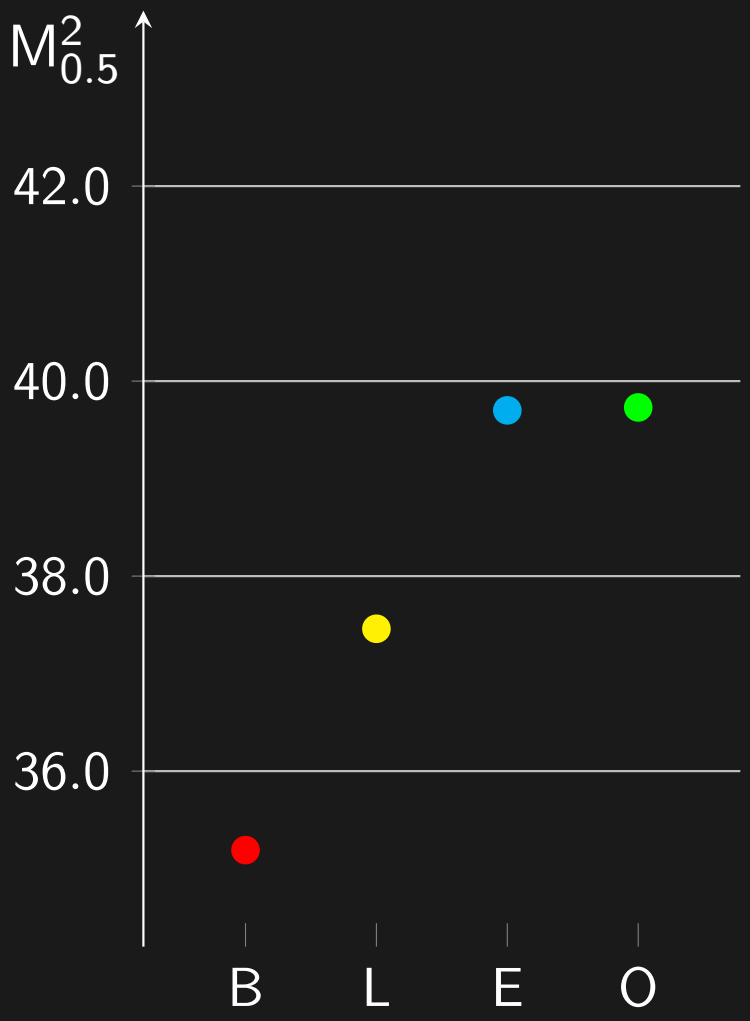


Optimized with BLEU

- (B)aseline: vanilla Moses, disabled reordering.
- (L)evenshtein Distance: word-based as TM score. Sums to number of edits.
- (E)dits: counts inserts, deletions, replacements.

Lesson 1: Implement task-specific features

Results optmized with BLEU

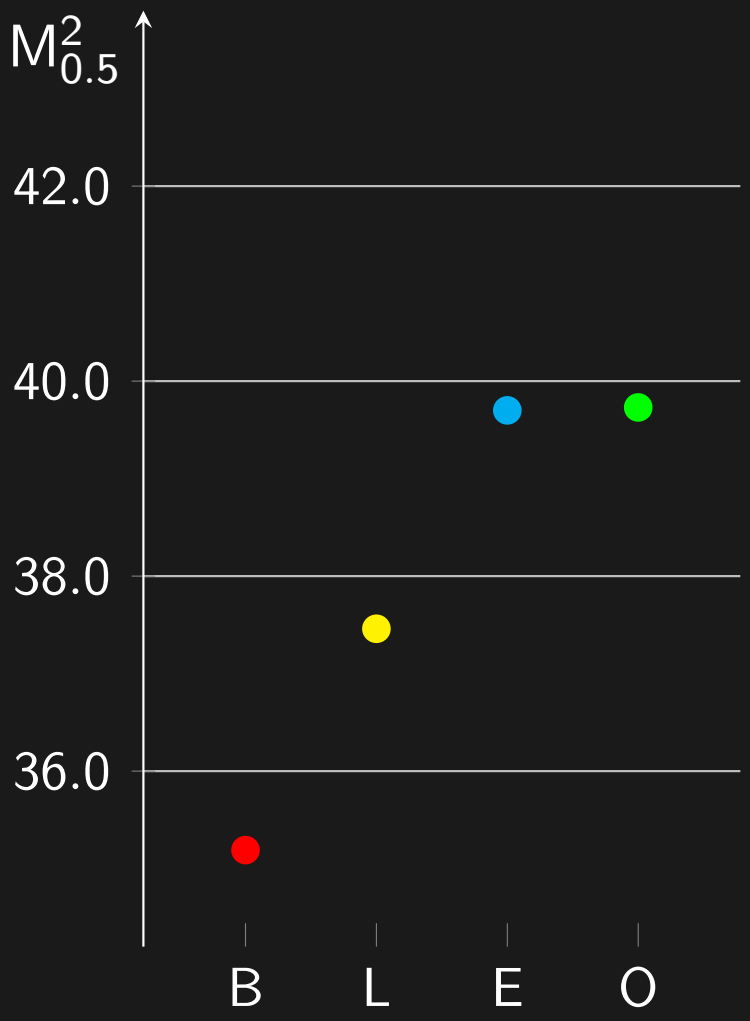


Optimized with BLEU

- (B)aseline: vanilla Moses, disabled reordering.
- (L)evenshtein Distance: word-based as TM score. Sums to number of edits.
- (E)dits: counts inserts, deletions, replacements.
- (O)peration Sequence Model: without reordering degenerates to stateful sequence model of edits.

Lesson 2: Optimize the right metric

Implement an optimizer for the metric that you have to use in the end

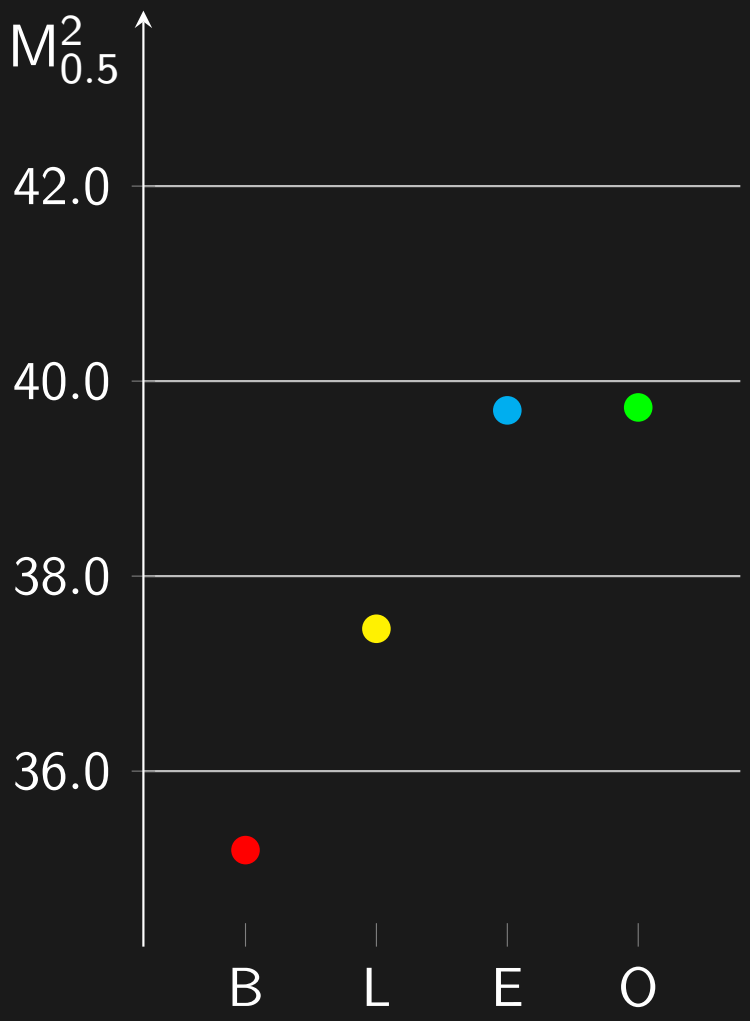


Optimized with BLEU

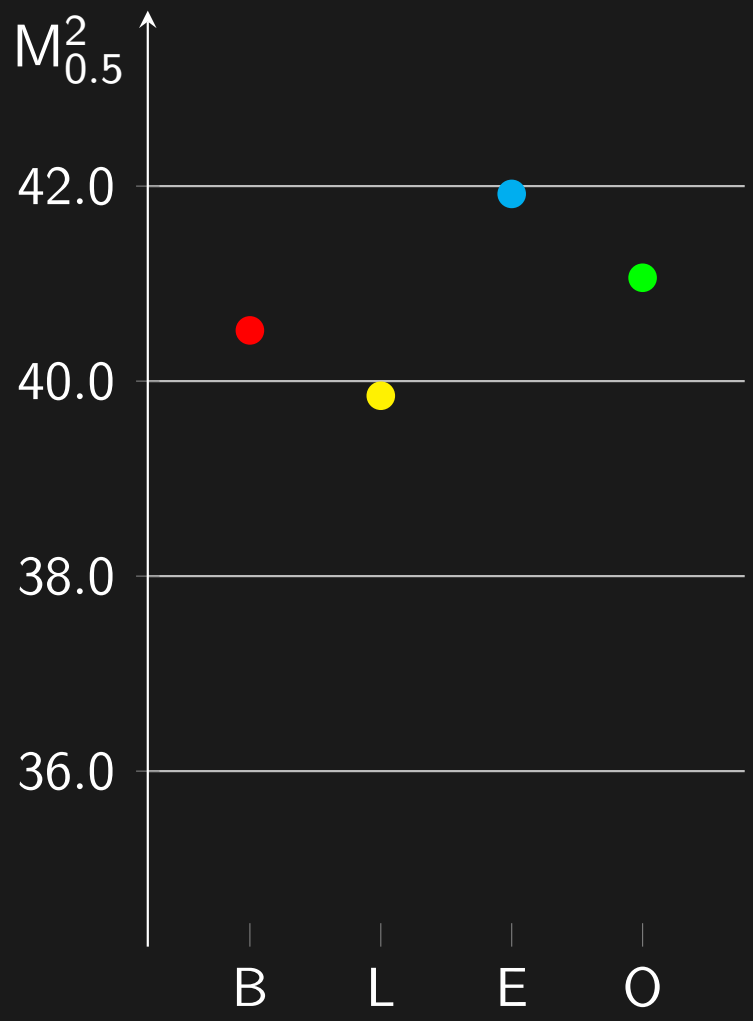
- (B)aseline: vanilla Moses, disabled reordering.
- (L)evenshtein Distance: word-based as TM score. Sums to number of edits.
- (E)dits: counts inserts, deletions, replacements.
- (O)peration Sequence Model: without reordering degenerates to stateful sequence model of edits.

Lesson 2: Optimize the right metric

Implement an optimizer for the metric that you have to use in the end



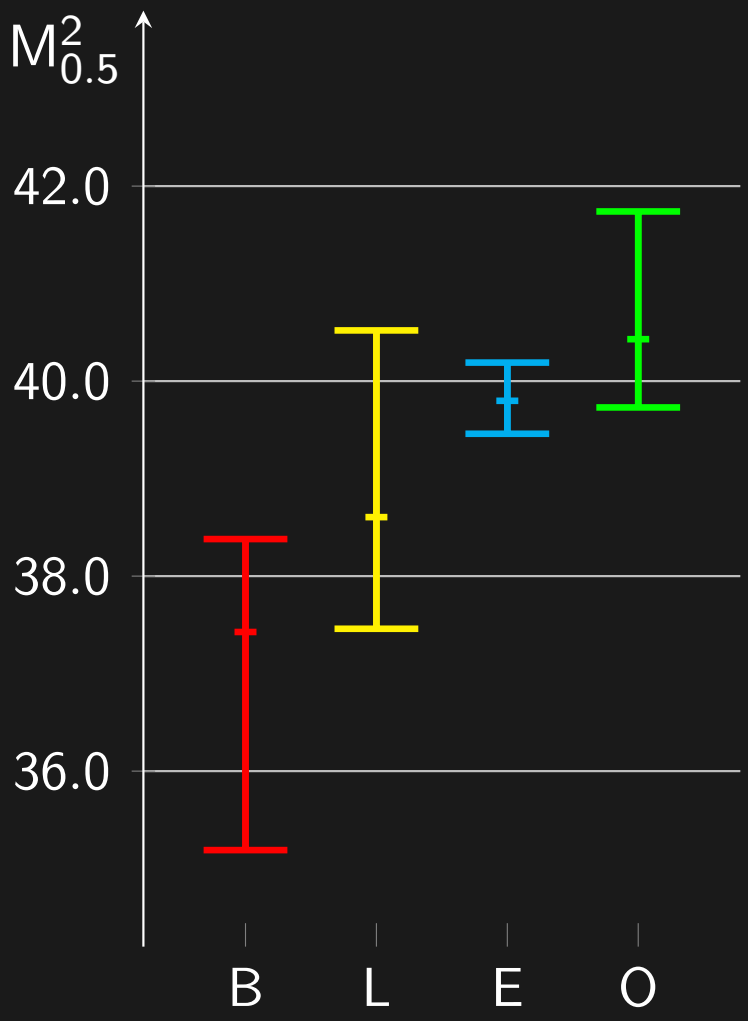
Optimized with BLEU



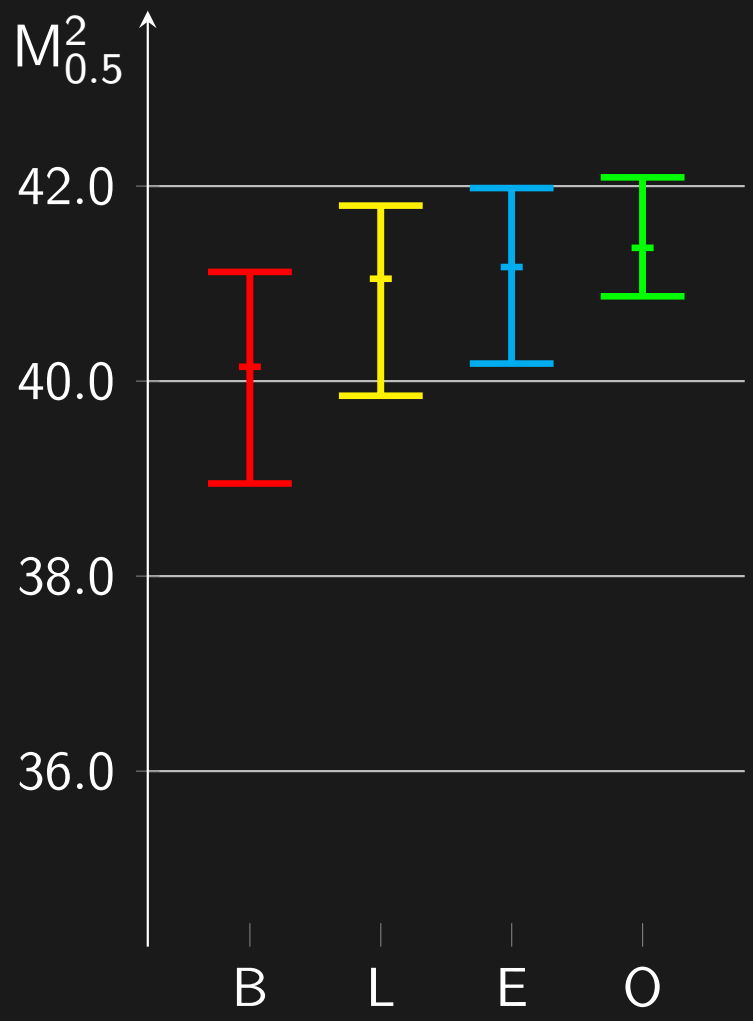
Optimized with M^2

Lesson 3: Account for optimizer instability

Or don't count on good luck (Clark et al. 2011)



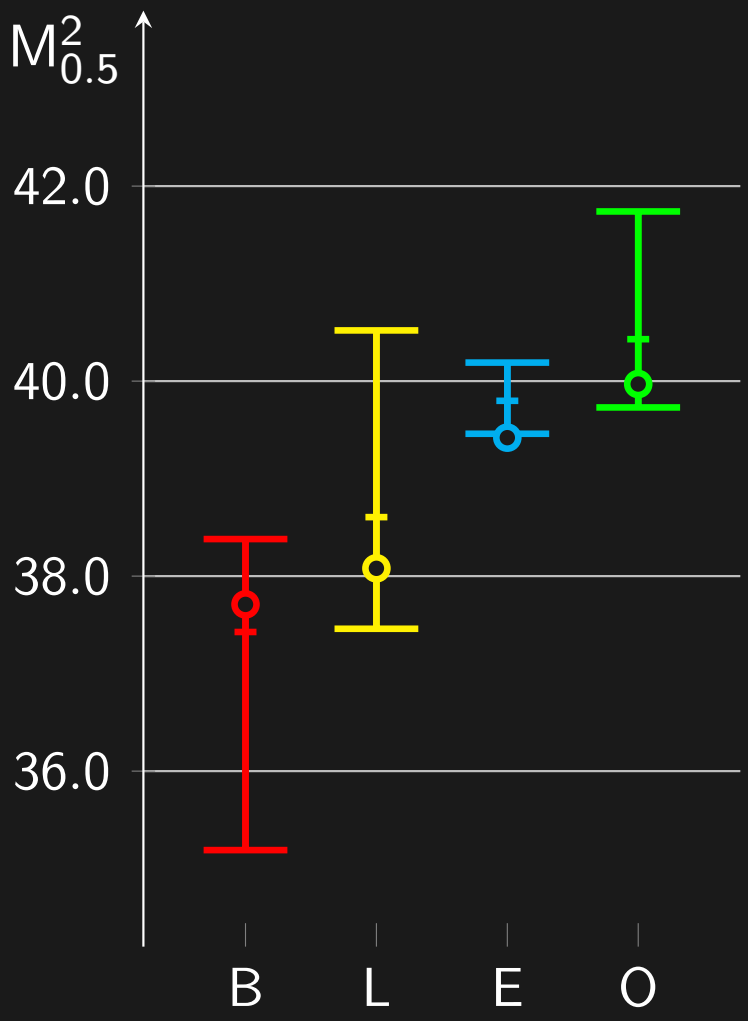
Optimized with BLEU



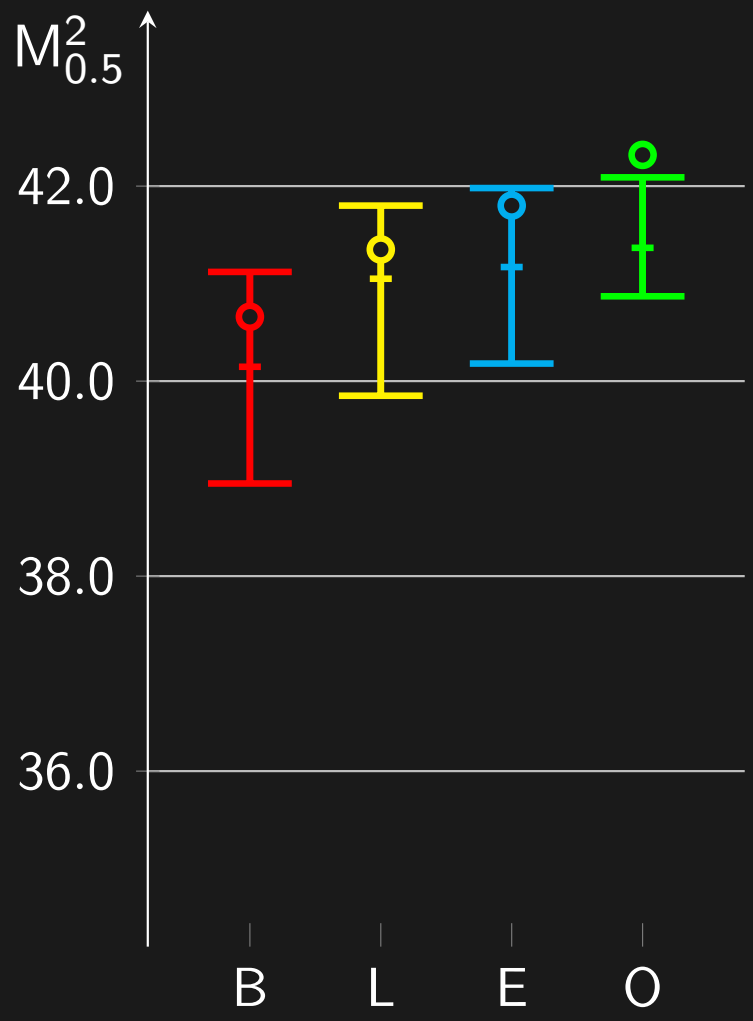
Optimized with M^2

Lesson 3: Account for optimizer instability

Averaged weights beat mean scores (Cettolo et al. 2011)

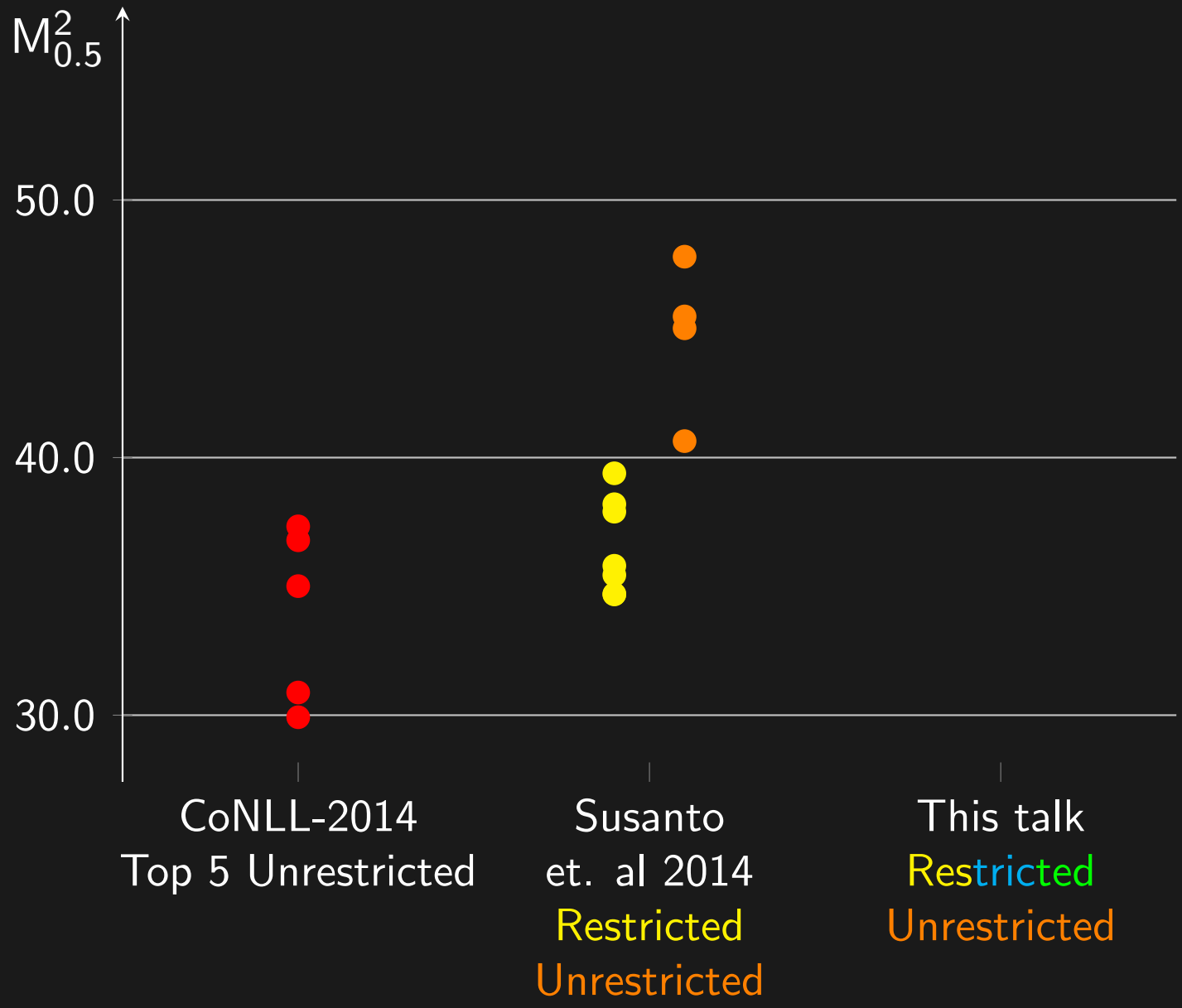


Optimized with BLEU

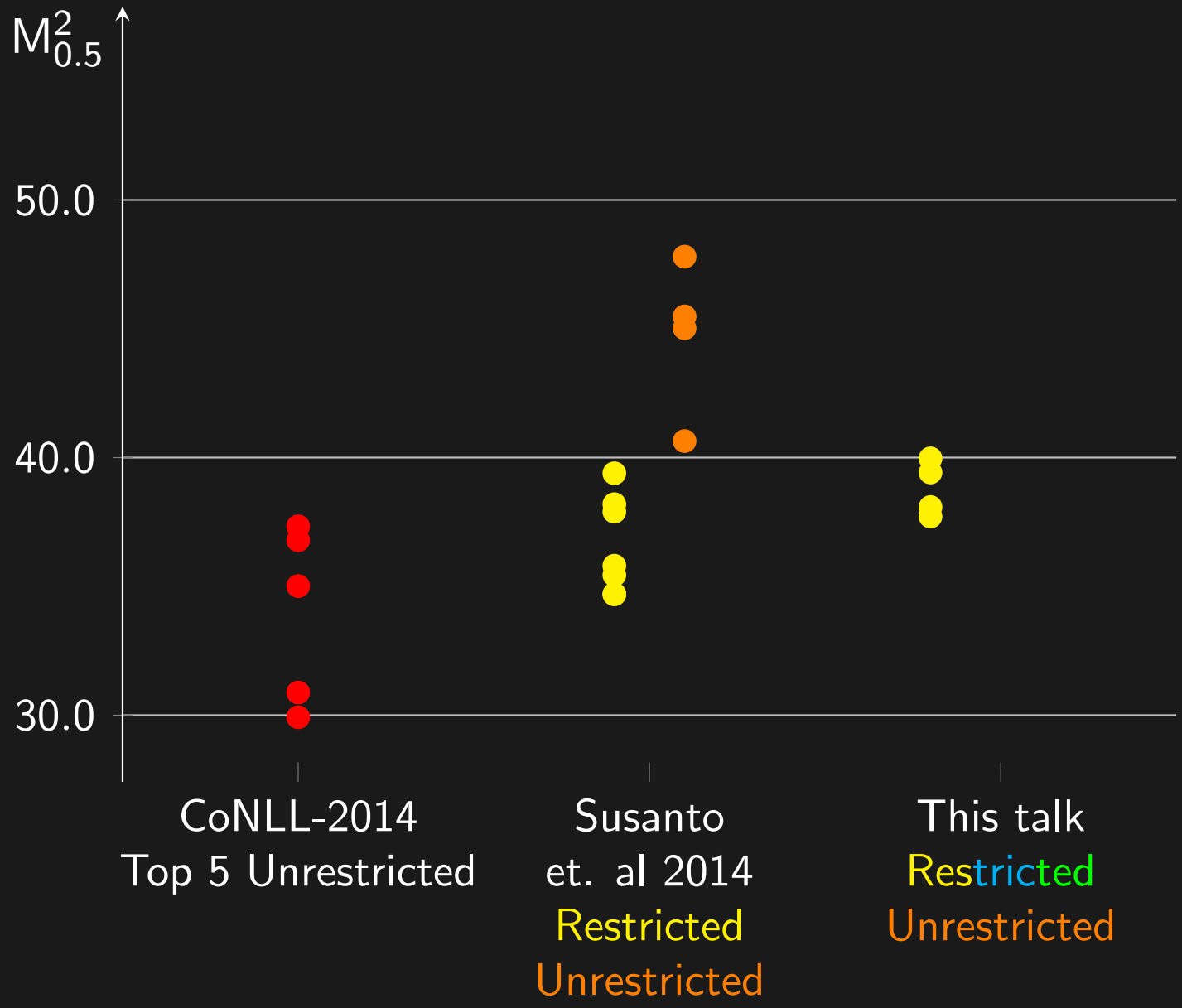


Optimized with M^2

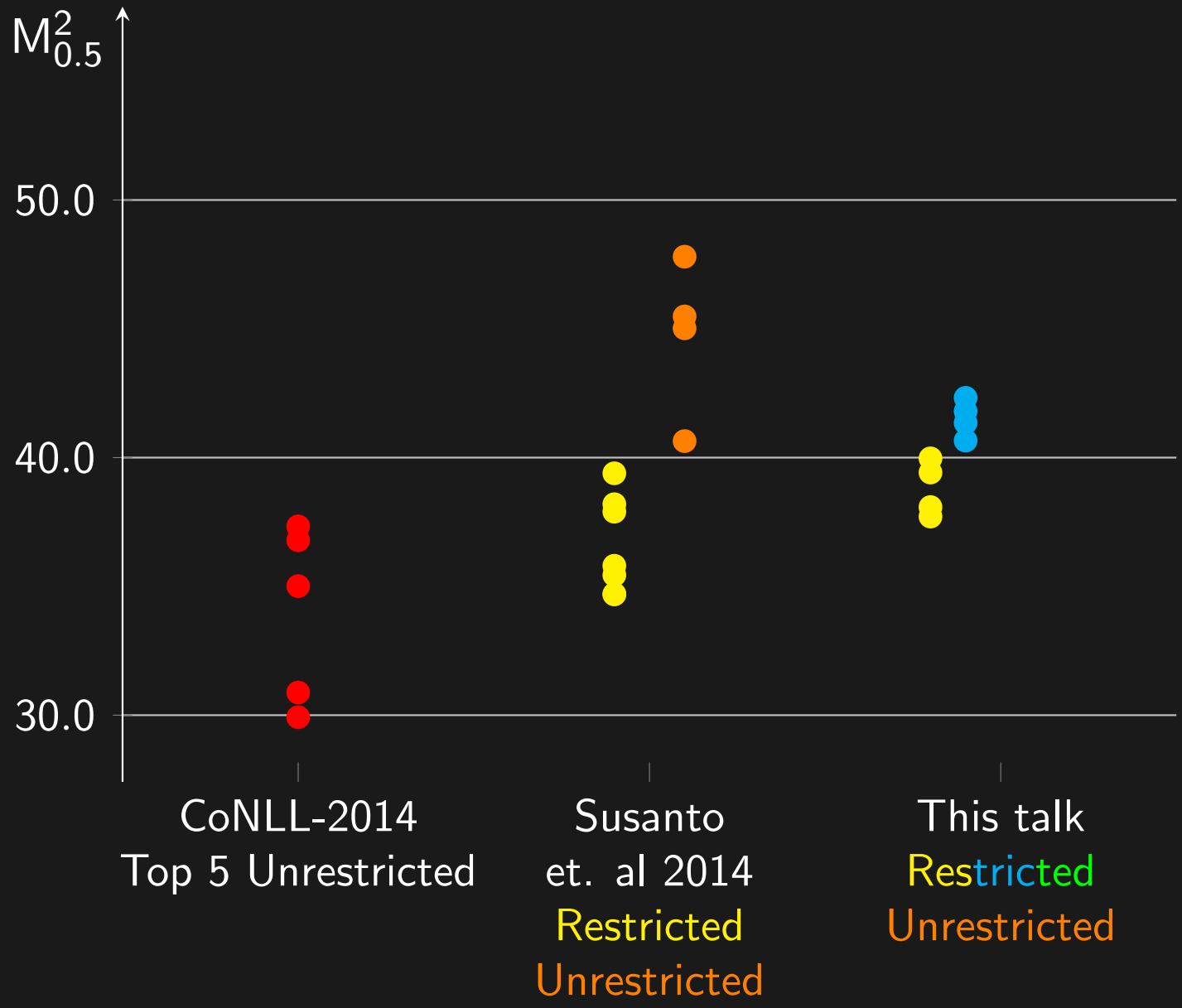
Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Lesson 4: Bigger development sets are better

Maybe a little too big

- Until now, CoNLL-2013 test set used as dev set:
  - Consists of only 1381 sentences with 3461 error annotations.
  - Rate of erroneous tokens: 14.97%

Lesson 4: Bigger development sets are better

Maybe a little too big

- Until now, CoNLL-2013 test set used as dev set:
  - Consists of only 1381 sentences with 3461 error annotations.
  - Rate of erroneous tokens: 14.97%
- But there is also NUCLE, currently used as training data:
  - Consists of only 57151 sentences with 44385 error annotations.
  - Rate of erroneous tokens: 6.23%

Lesson 4: Bigger development sets are better

Maybe a little too big

- Until now, CoNLL-2013 test set used as dev set:
  - Consists of only 1381 sentences with 3461 error annotations.
  - Rate of erroneous tokens: 14.97%
- But there is also NUCLE, currently used as training data:
  - Consists of only 57151 sentences with 44385 error annotations.
  - Rate of erroneous tokens: 6.23%
- We are not supposed to look at the error rate of the CoNLL-2014 test set (ca. 10% for Annotator 1, ca. 12% for Annotator 2)

Lesson 4: Bigger development sets are better

Turning NUCLE into a development set via cross validation

1. Greedily remove sentences until error rate 0.15 is reached. Leaves: 23381 sentences, 39707 annotations.
2. Divide into 4 equal-sized subsets.
3. Tune on 1 subset, add remaining 3 subsets to training data, repeat for each subset.
4. Average weights for all subsets into single weight vector.
5. Repeat steps 3-4 a couple of times (here: 5) to average weights accross iterations.

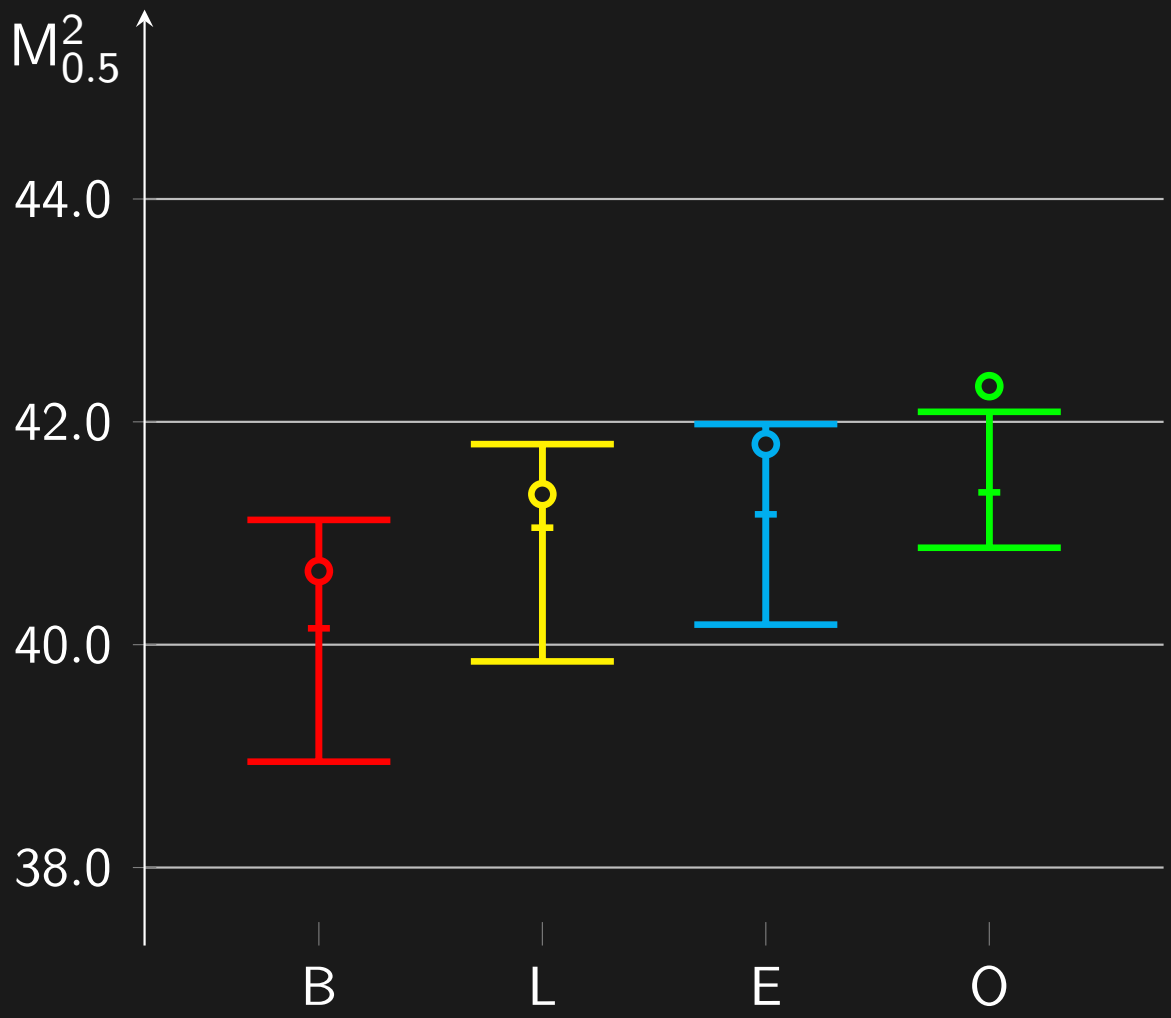
Lesson 4: Bigger development sets are better

Turning NUCLE into a development set via cross validation

- We are not using the previous dev set, though it could just be added to all subsets.
- It is nice to have a second test set (although slighlty cheating due to matching error rate).
- Effect of error rate on tuning:
  - Low error rate: high precision.
  - High error rate: high recall.
- Future work: Can the error rate be used to adjust training data, too?

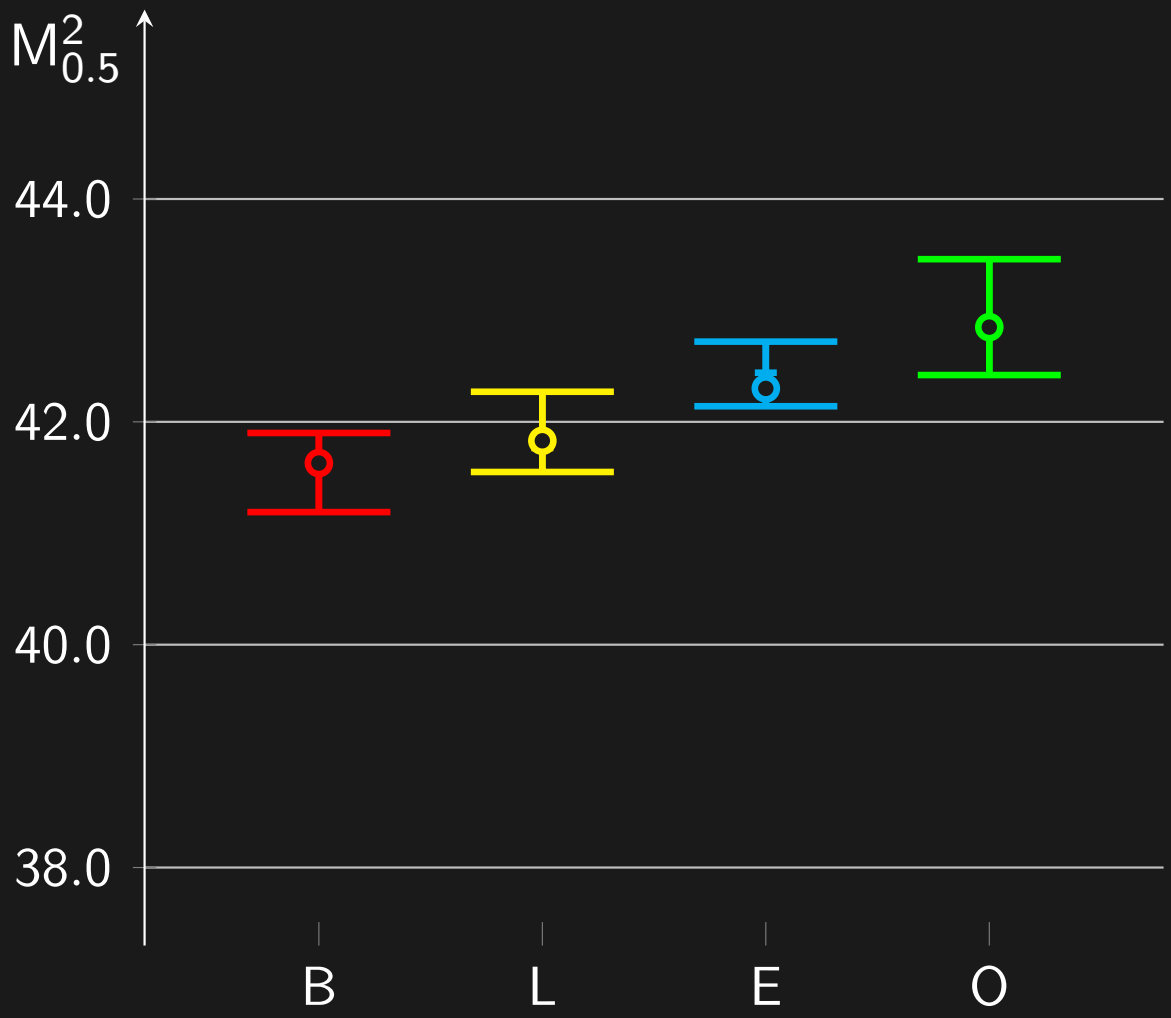
Lesson 4: Bigger development sets are better

Performance jump from switching dev sets

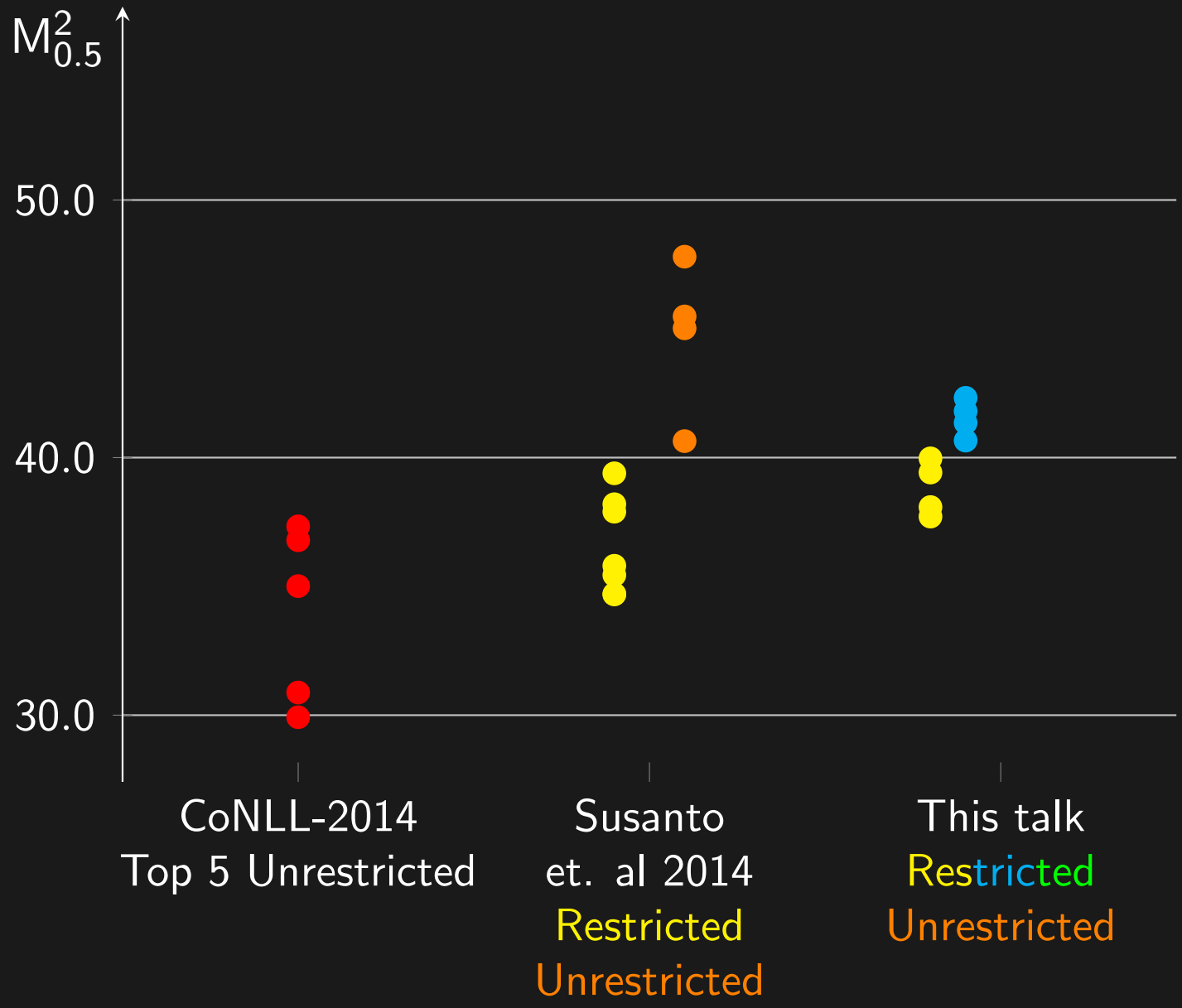


Lesson 4: Bigger development sets are better

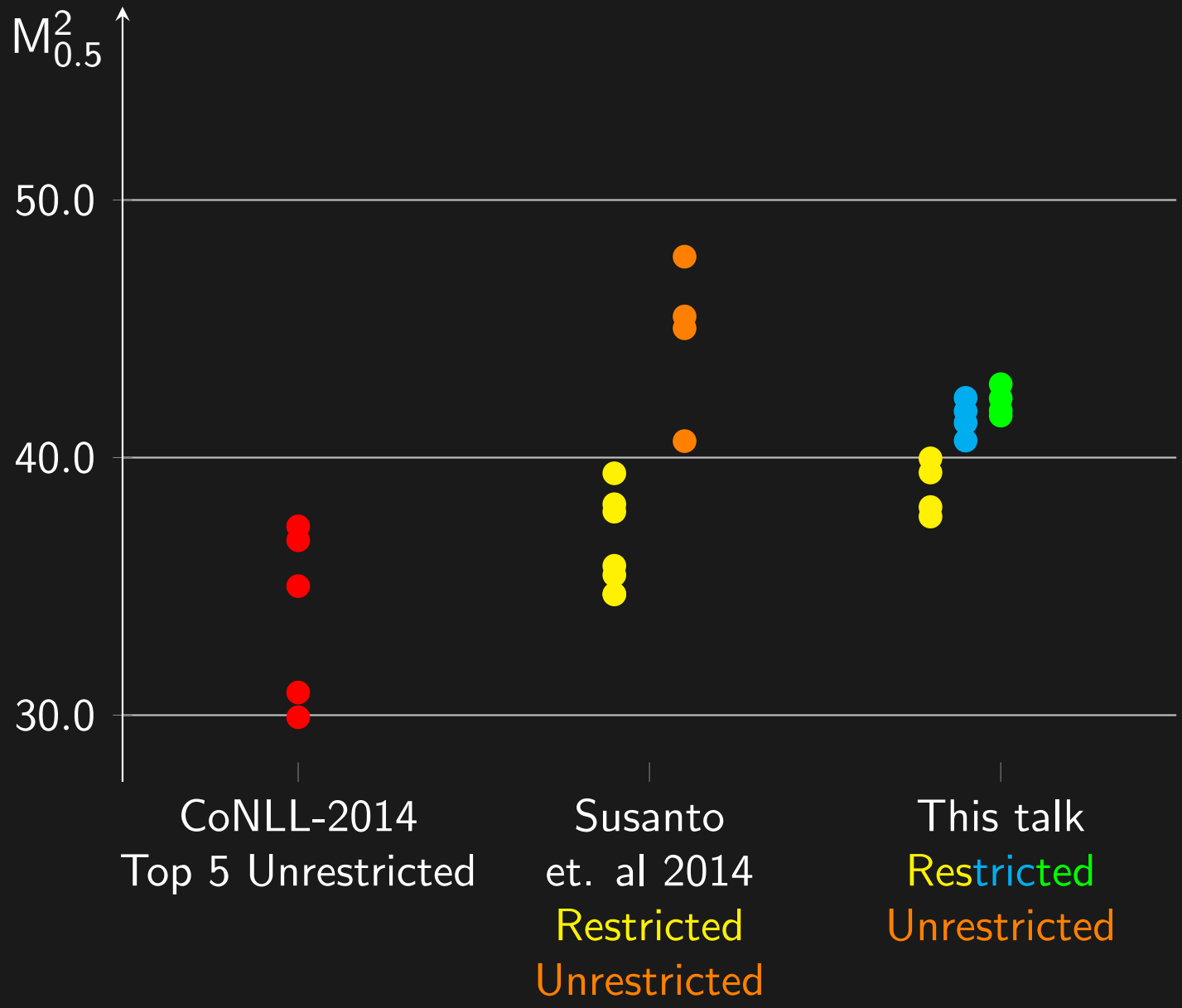
Performance jump from switching dev sets



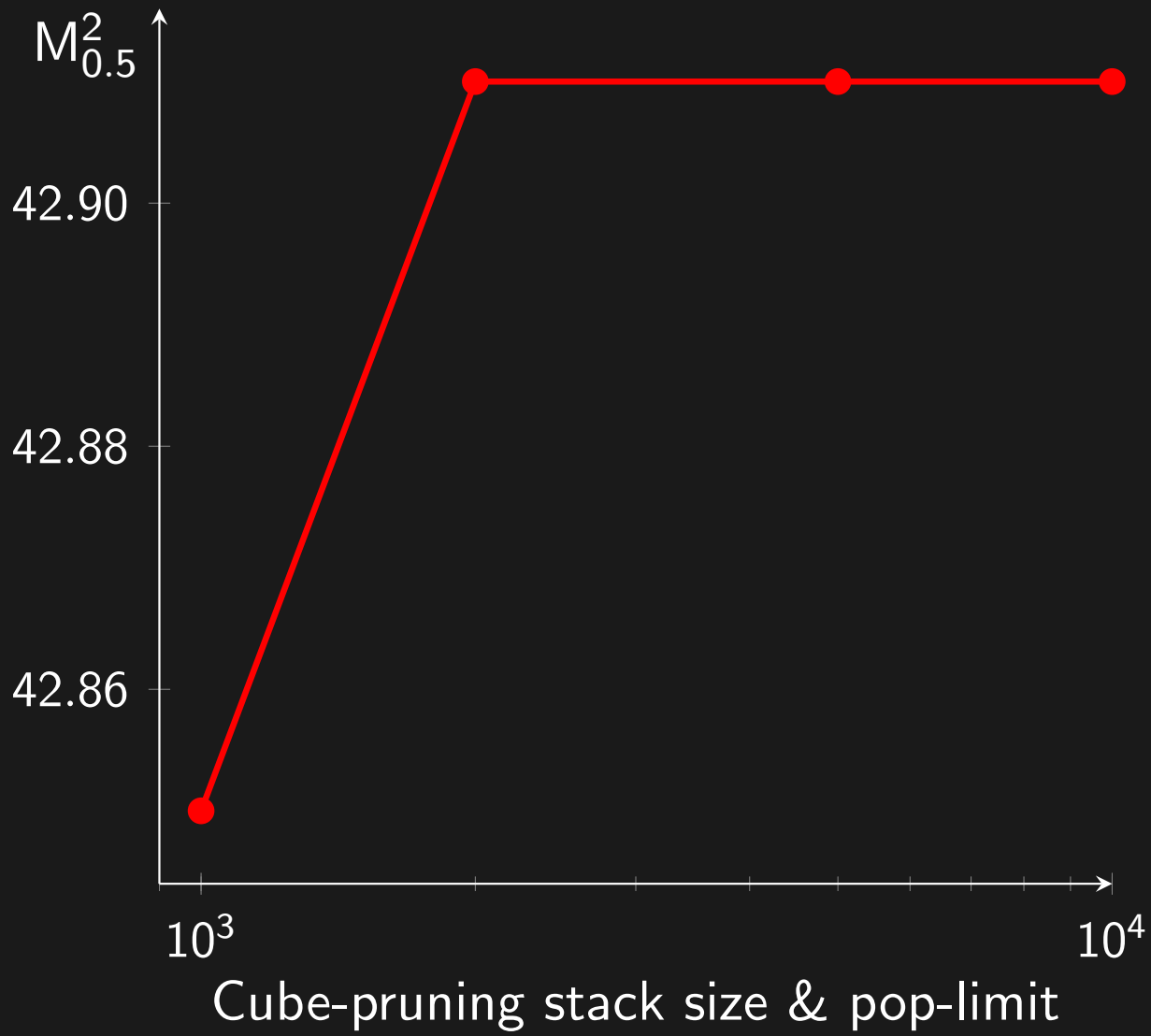
Performance on the CoNLL-2014 Shared Task test set



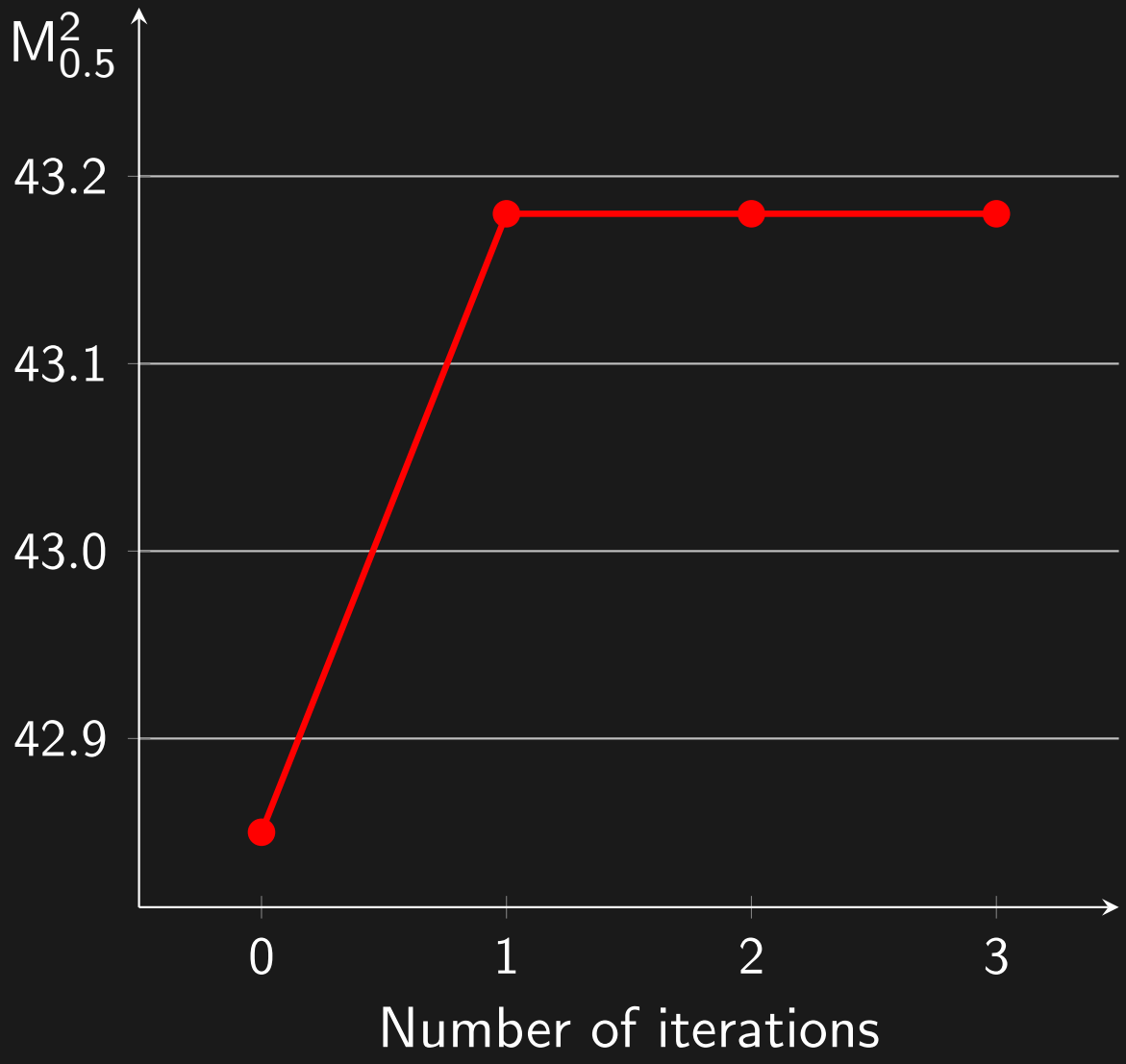
Performance on the CoNLL-2014 Shared Task test set



Lesson 5: Expanding the search space barely helps



Lesson 6: Self-composition helps (once)



Lesson 7: There's something about sparse-features

But tuning them correctly is hard

- In our CoNLL-2014 shared task submission, we reported that sparse features help.
- Correlation without causation: what actually helped was the tuning scheme.
- Still, we strongly believe that for GEC sparse features are something worth exploring.

Lesson 7: There's something about sparse-features

But tuning them correctly is hard

- In our CoNLL-2014 shared task submission, we reported that sparse features help.
- Correlation without causation: what actually helped was the tuning scheme.
- Still, we strongly believe that for GEC sparse features are something worth exploring.
- Problems:
  - kbMIRA does not seem to work well with M^2.
  - Both PRO and kbMIRA give worse results than MERT for M^2 even without sparse features.
  - PRO seems to even out with sparse features.

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1
subst(comes, surfaces)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1
subst(comes, surfaces)=1
del(out)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1
insert(,)=1
subst(comes, surfaces)=1
del(out)=1

Lesson 7: There's something about sparse-features

Sparse feature examples

Source: Then a new problem comes out .

Target: Hence , a new problem surfaces .

subst(Then,Hence)=1 <s>_subst(Then,Hence)_a=1
insert(,)=1 Hence_insert(,)_a=1
subst(comes, surfaces)=1 problem_subst(comes, surfaces)_out=1
del(out)=1 comes_del(out)_.=1

Lesson 8: Nothing beats the language model

Which is actually quite frustrating

- Using Kenneth Heafield's CommonCrawl LM is near impossible due to hardware bottleneck.

Lesson 8: Nothing beats the language model

Which is actually quite frustrating

- Using Kenneth Heafield's CommonCrawl LM is near impossible due to hardware bottleneck.
- Still working on way to find out the upper bound.

Lesson 8: Nothing beats the language model

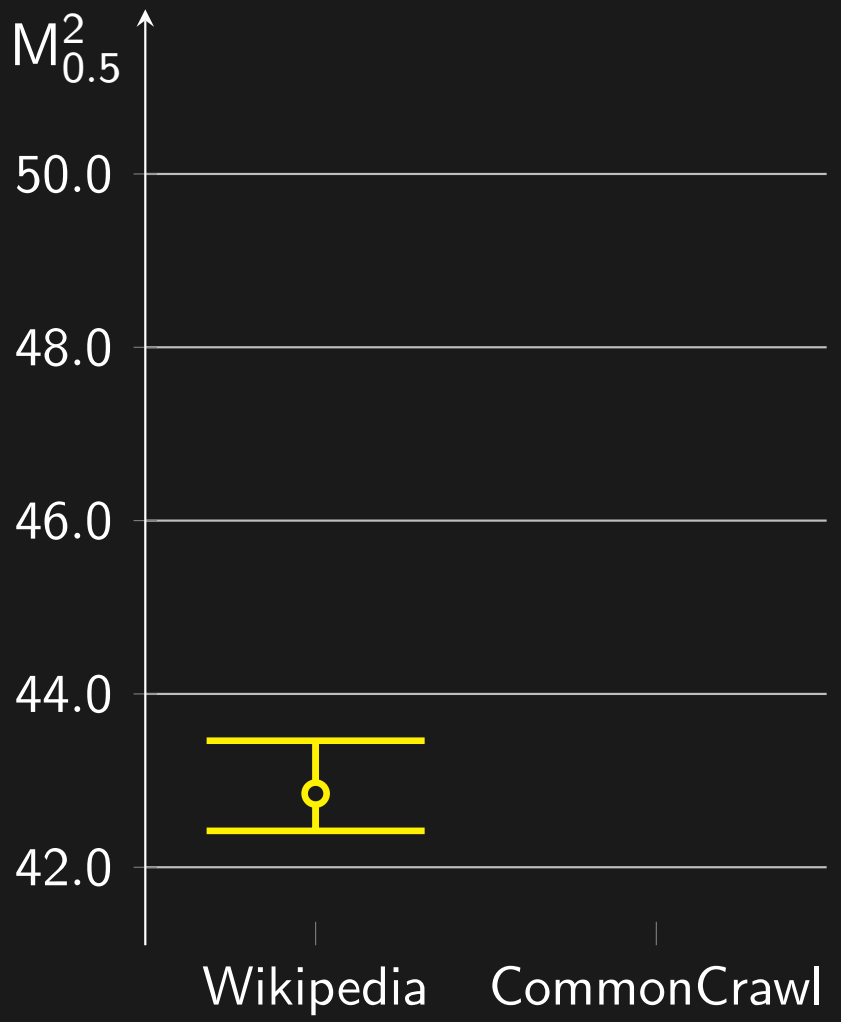
Which is actually quite frustrating

- Using Kenneth Heafield's CommonCrawl LM is near impossible due to hardware bottleneck.
- Still working on way to find out the upper bound.
- Instead: sampled a subset from the the raw text data with cross-entropy difference filtering.

Lesson 8: Nothing beats the language model

Which is actually quite frustrating

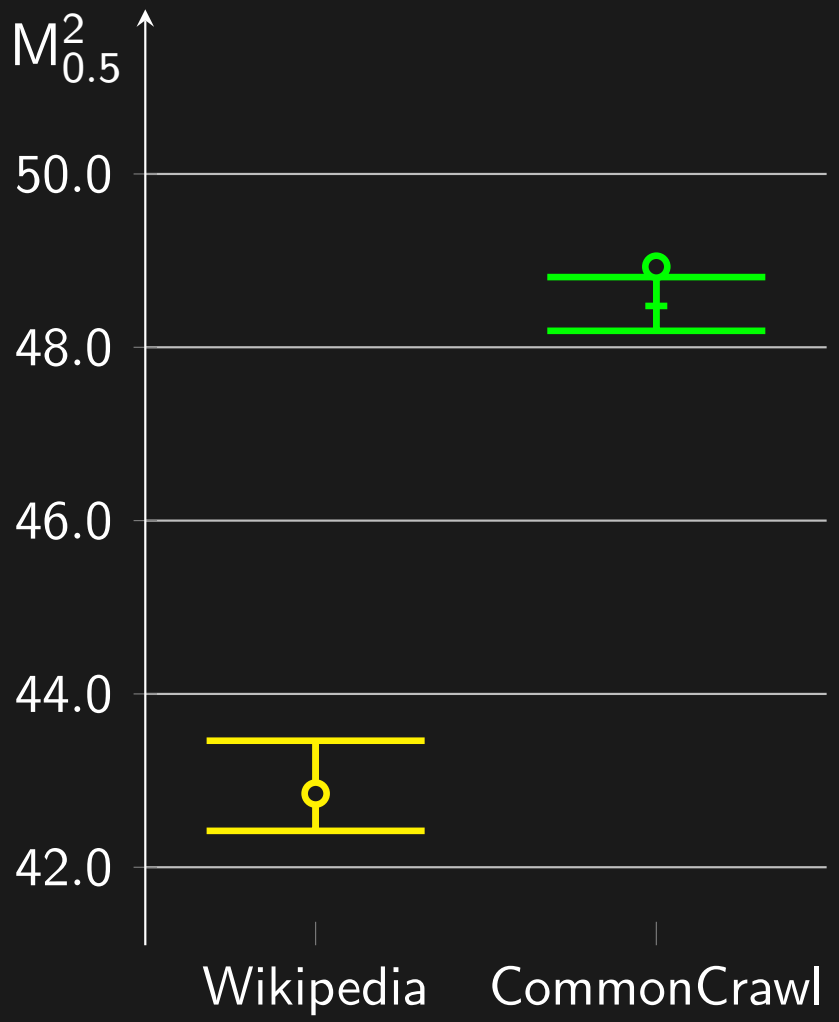
- Using Kenneth Heafield's CommonCrawl LM is near impossible due to hardware bottleneck.
- Still working on way to find out the upper bound.
- Instead: sampled a subset from the the raw text data with cross-entropy difference filtering.



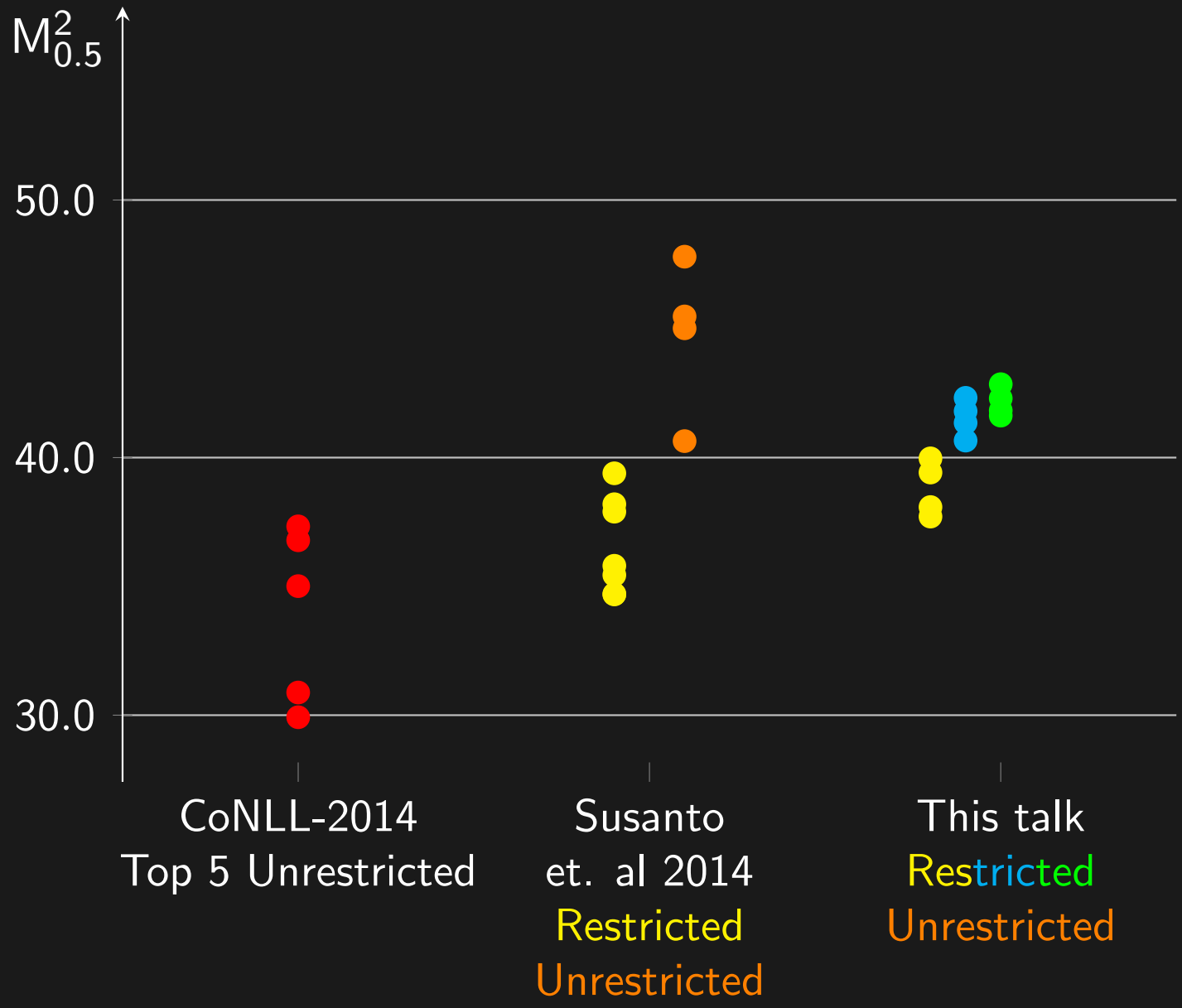
Lesson 8: Nothing beats the language model

Which is actually quite frustrating

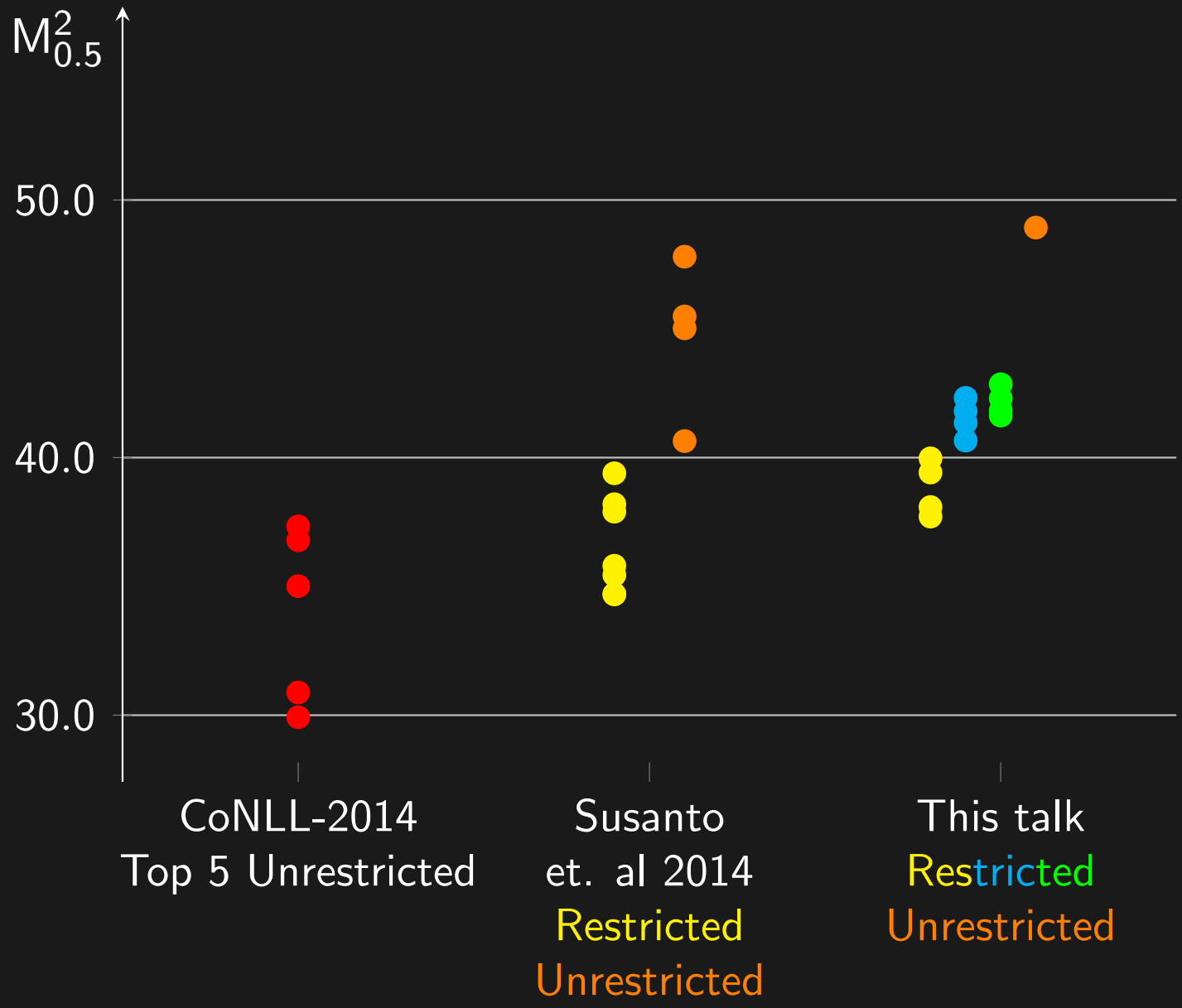
- Using Kenneth Heafield's CommonCrawl LM is near impossible due to hardware bottleneck.
- Still working on way to find out the upper bound.
- Instead: sampled a subset from the the raw text data with cross-entropy difference filtering.



Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Lesson 9: Collecting parallel monolingual data is hard

- For the Shared Task 2014 we scraped more data from Lang-8.

Lesson 9: Collecting parallel monolingual data is hard

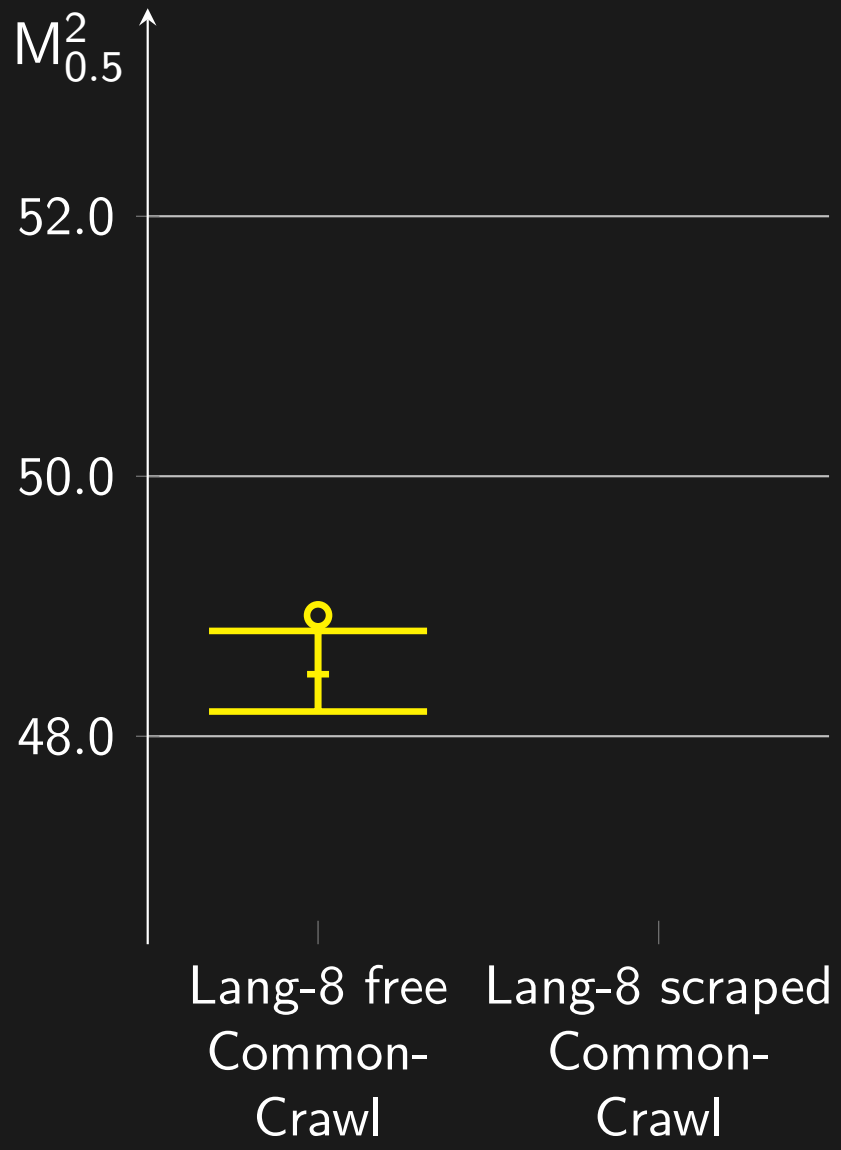
- For the Shared Task 2014 we scraped more data from Lang-8.
- Collected twice as many sentences (ca. 4M) compared to the free release with 2M .

Lesson 9: Collecting parallel monolingual data is hard

- For the Shared Task 2014 we scraped more data from Lang-8.
- Collected twice as many sentences (ca. 4M) compared to the free release with 2M .
- Legally and morally dubious.

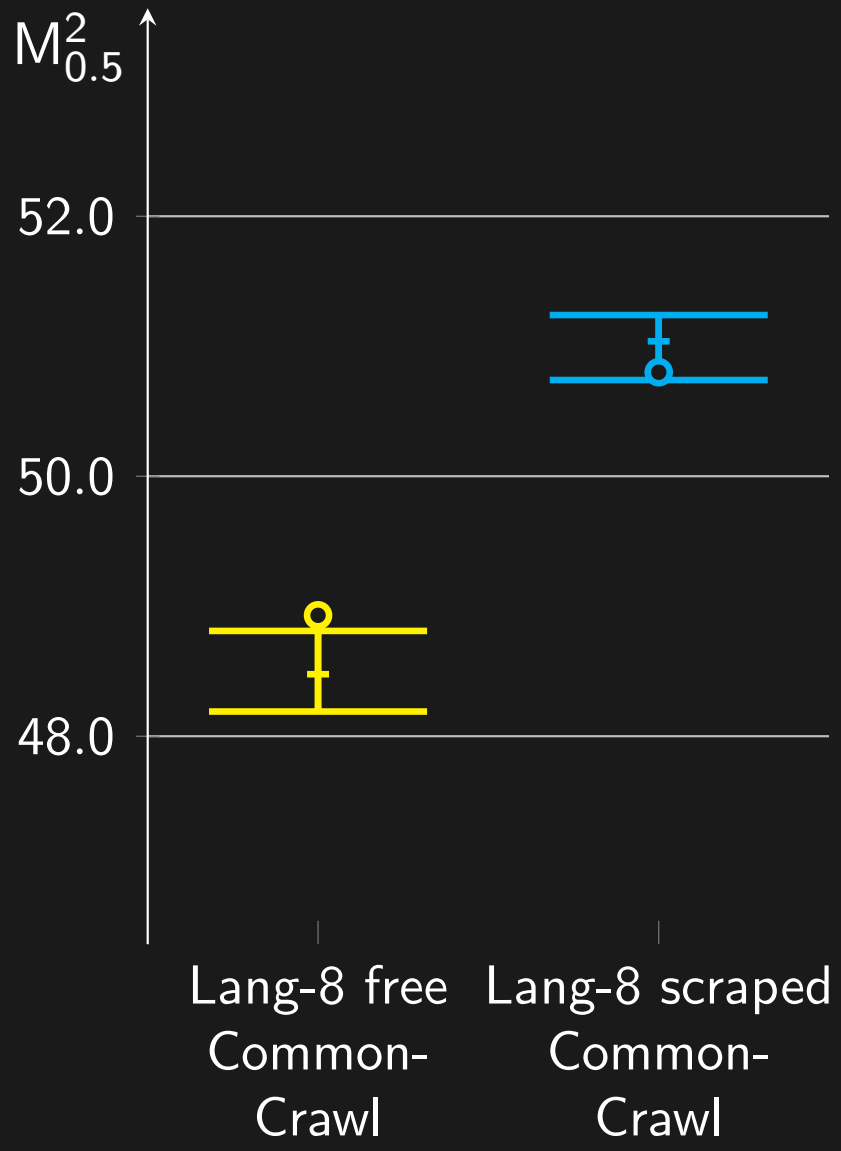
Lesson 9: Collecting parallel monolingual data is hard

- For the Shared Task 2014 we scraped more data from Lang-8.
- Collected twice as many sentences (ca. 4M) compared to the free release with 2M .
- Legally and morally dubious.

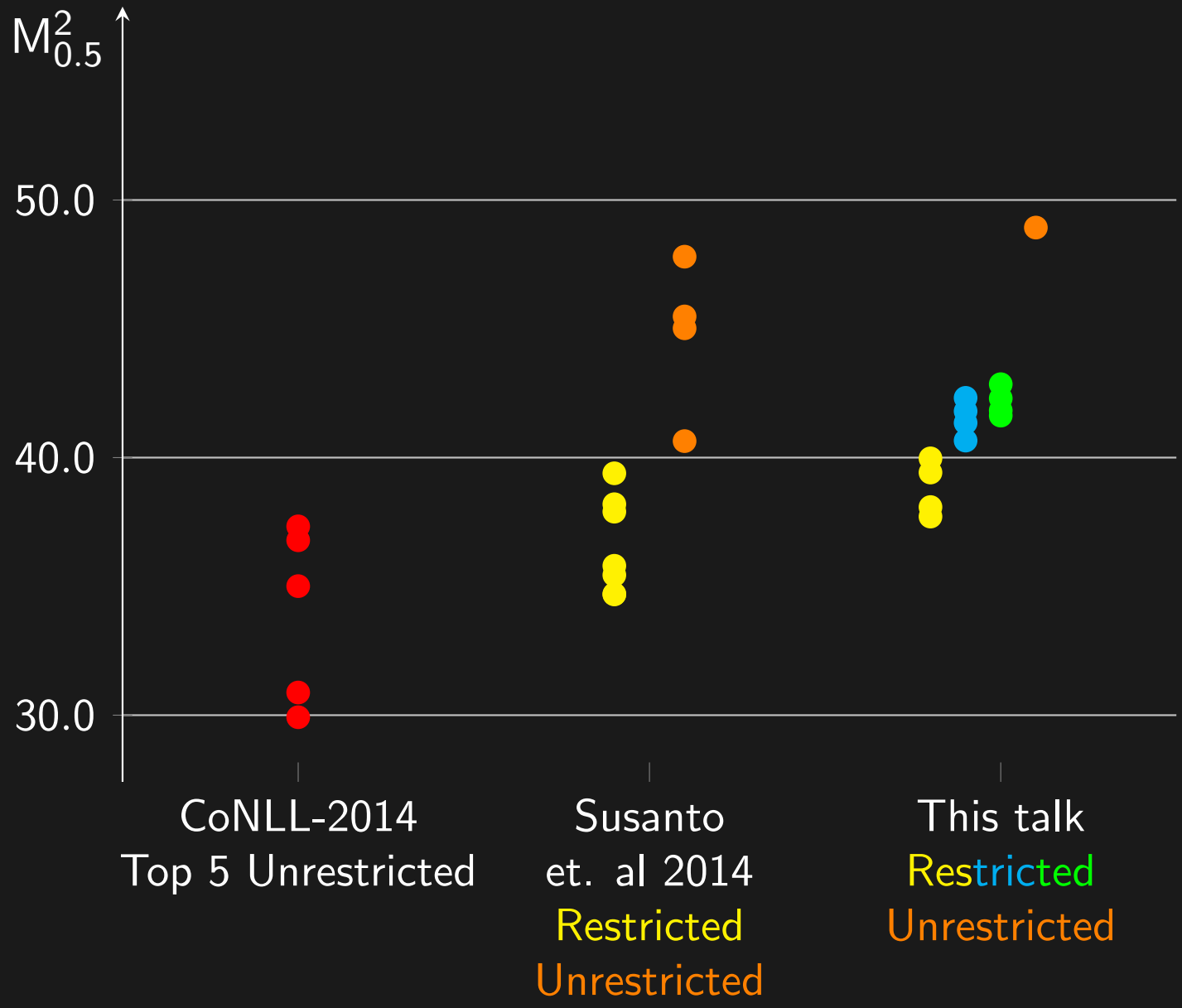


Lesson 9: Collecting parallel monolingual data is hard

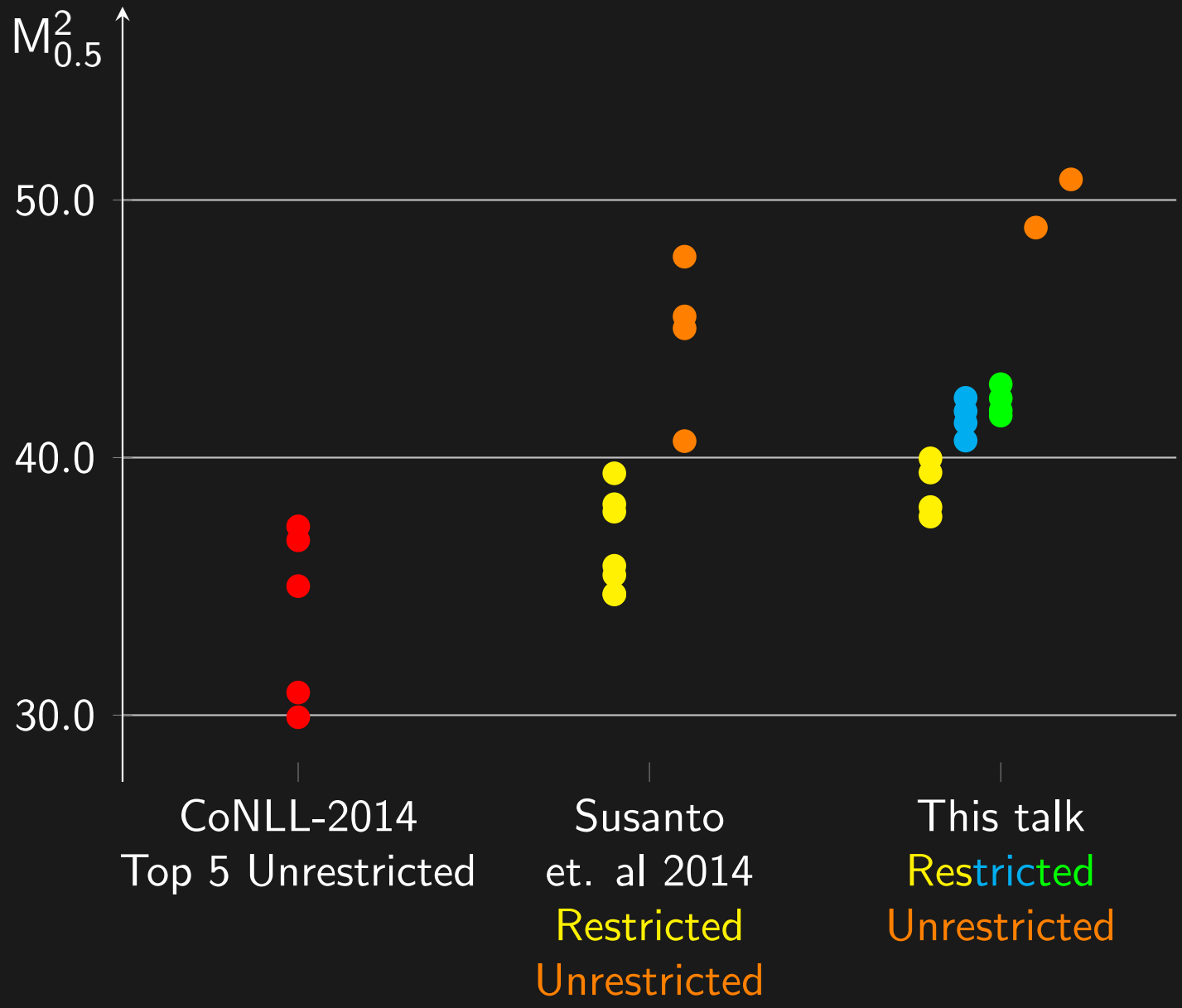
- For the Shared Task 2014 we scraped more data from Lang-8.
- Collected twice as many sentences (ca. 4M) compared to the free release with 2M .
- Legally and morally dubious.



Performance on the CoNLL-2014 Shared Task test set



Performance on the CoNLL-2014 Shared Task test set



Lesson 9: Collecting parallel monolingual data is hard

Other less obvious but legally safe sources

- Wikipedia edit histories (Done that)
  - Over 14 million sentences with small editions.
  - Publically available as the WikEd corpus.
  - Currently thinking what to do with that.

Lesson 9: Collecting parallel monolingual data is hard

Other less obvious but legally safe sources

- Wikipedia edit histories (Done that)
  - Over 14 million sentences with small editions.
  - Publically available as the WikEd corpus.
  - Currently thinking what to do with that.
- Other wikis: over 400,000 Wikia wikis (Working on it)
  - The same format as Wikipedia-Dumps.
  - Our scripts can already work with that.
  - Wikia has a local branch in Poznań!
  - And one of my students is an intern

Lesson 9: Collecting parallel monolingual data is hard

Other less obvious but legally safe sources

- Wikipedia edit histories (Done that)
  - Over 14 million sentences with small editions.
  - Publically available as the WikEd corpus.
  - Currently thinking what to do with that.
- Other wikis: over 400,000 Wikia wikis (Working on it)
  - The same format as Wikipedia-Dumps.
  - Our scripts can already work with that.
  - Wikia has a local branch in Poznań!
  - And one of my students is an intern
- Diff between two Commoncrawl revisions (Dreading it)
  - Take revisions maybe one year apart.
  - Group documents by URL.
  - Sentence align and keep pairs with small differences.

Lesson 9: Collecting parallel monolingual data is hard

Other less obvious but legally safe sources

- Wikipedia edit histories (Done that)
  - Over 14 million sentences with small editions.
  - Publically available as the WikEd corpus.
  - Currently thinking what to do with that.
- Other wikis: over 400,000 Wikia wikis (Working on it)
  - The same format as Wikipedia-Dumps.
  - Our scripts can already work with that.
  - Wikia has a local branch in Poznań!
  - And one of my students is an intern
- Diff between two Commoncrawl revisions (Dreading it)
  - Take revisions maybe one year apart.
  - Group documents by URL.
  - Sentence align and keep pairs with small differences.
- ...

Lesson 10: GEC needs a better metric

Maybe tuning with BLEU is actually smarter

	Team	P	R	$M^2_{0.5}$
1	CAMB	39.71	30.10	37.33
2	CUUI	41.78	24.88	36.79
3	AMU	41.62	21.40	35.01
4	POST	34.51	21.73	30.88
5	NTHU	35.08	18.85	29.92
6	RAC	33.14	14.99	26.68
7	UMC	31.27	14.46	25.37
8	PKU	32.21	13.65	25.32
9	NARA	21.57	29.38	22.78
10	SJTU	30.11	5.10	15.19
11	UFC	70.00	1.72	7.84
12	IPN	11.28	2.85	7.09
13	IITB	30.77	1.39	5.90

Lesson 10: GEC needs a better metric

Maybe tuning with BLEU is actually smarter

	Team	P	R	$M^2_{0.5}$
1	CAMB	39.71	30.10	37.33
2	CUUI	41.78	24.88	36.79
3	AMU	41.62	21.40	35.01
4	POST	34.51	21.73	30.88
5	NTHU	35.08	18.85	29.92
6	RAC	33.14	14.99	26.68
7	UMC	31.27	14.46	25.37
8	PKU	32.21	13.65	25.32
9	NARA	21.57	29.38	22.78
10	SJTU	30.11	5.10	15.19
11	UFC	70.00	1.72	7.84
12	IPN	11.28	2.85	7.09
13	IITB	30.77	1.39	5.90
14	Input	0.00	0.00	0.00

Lesson 10: GEC needs a better metric

Maybe tuning with BLEU is actually smarter

	Team	P	R	$M^2_{0.5}$	BLEU
1	CAMB	39.71	30.10	37.33	81.77
2	CUUI	41.78	24.88	36.79	83.46
3	AMU	41.62	21.40	35.01	83.42
4	POST	34.51	21.73	30.88	81.61
5	NTHU	35.08	18.85	29.92	82.42
6	RAC	33.14	14.99	26.68	81.91
7	UMC	31.27	14.46	25.37	83.66
8	PKU	32.21	13.65	25.32	83.71
9	NARA	21.57	29.38	22.78	–
10	SJTU	30.11	5.10	15.19	85.96
11	UFC	70.00	1.72	7.84	86.82
12	IPN	11.28	2.85	7.09	83.39
13	IITB	30.77	1.39	5.90	86.50
14	Input	0.00	0.00	0.00	86.79

Lesson 10: GEC needs a better metric

Maybe tuning with BLEU is actually smarter

	Team	P	R	$M^2_{0.5}$	BLEU
11	UFC	70.00	1.72	7.84	86.82
14	Input	0.00	0.00	0.00	86.79
13	IITB	30.77	1.39	5.90	86.50
10	SJTU	30.11	5.10	15.19	85.96
8	PKU	32.21	13.65	25.32	83.71
7	UMC	31.27	14.46	25.37	83.66
2	CUUI	41.78	24.88	36.79	83.46
3	AMU	41.62	21.40	35.01	83.42
12	IPN	11.28	2.85	7.09	83.39
5	NTHU	35.08	18.85	29.92	82.42
6	RAC	33.14	14.99	26.68	81.91
1	CAMB	39.71	30.10	37.33	81.77
4	POST	34.51	21.73	30.88	81.61
9	NARA	21.57	29.38	22.78	–

Summing up

- Proper parameter tuning is crucial.

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.
- With proper tuning, vanilla Moses beats published top-results on the CoNLL-2014 shared task (39.39% vs. 41.63%) when using restricted training data.

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.
- With proper tuning, vanilla Moses beats published top-results on the CoNLL-2014 shared task (39.39% vs. 41.63%) when using restricted training data.
- New top-result with restricted data (1-composition): 43.18%

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.
- With proper tuning, vanilla Moses beats published top-results on the CoNLL-2014 shared task (39.39% vs. 41.63%) when using restricted training data.
- New top-result with restricted data (1-composition): 43.18%
- With additonal LM and TM data and task-specific features, Moses beats all unrestricted systems and previously published system combinations (partially tuned on test data).

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.
- With proper tuning, vanilla Moses beats published top-results on the CoNLL-2014 shared task (39.39% vs. 41.63%) when using restricted training data.
- New top-result with restricted data (1-composition): 43.18%
- With additonal LM and TM data and task-specific features, Moses beats all unrestricted systems and previously published system combinations (partially tuned on test data).
- New top-result with unrestricted data: 50.80% (51.00% 1-composition)

Summing up

- Proper parameter tuning is crucial.
- The language model is the strongest feature.
- With proper tuning, vanilla Moses beats published top-results on the CoNLL-2014 shared task (39.39% vs. 41.63%) when using restricted training data.
- New top-result with restricted data (1-composition): 43.18%
- With additonal LM and TM data and task-specific features, Moses beats all unrestricted systems and previously published system combinations (partially tuned on test data).
- New top-result with unrestricted data: 50.80% (51.00% 1-composition)
- But: the M^2 metric is highly dubious.

Future Work

There is still a lot to learn:

- How do I properly tune sparse features with M^2 and PRO or kbMIRA?
- Is M^2 a sensible metric at all?
- Where and how can I collect more data?
- What's the effect of other LMs like neural language models
- Can I fully integrate ML methods (Vowpal Wabbit)?
- ...

References

- Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram Counts and Language Models from the Common Crawl. LREC 2014.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. BEA 2013.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. Better Evaluation for Grammatical Error Correction. NAACL 2012.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. CoNLL 2014.

References

- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. CoNLL 2014.
- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. COLING 2012.
- Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. HLT 2011.
- Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. 2011. Methods for smoothing the optimizer instability in SMT. MT Summit 2011.