Optimizing for Sentence-Level BLEU+1 Yields Short Translations

Preslav Nakov, Francisco Guzman, Stephan Vogel
Qatar Computing Research Institute

INTRODUCTION

Statistical Machine Translation (SMT): Trained on Parallel Texts

SMT systems:
- **learn** from human-generated translations
- extract **useful knowledge** and build **models**
- use the models to **translate** new sentences

SMT: A Log-linear Model

$$E = \arg\max_{\hat{E}} < w, \Phi(\hat{E}, F) >$$

- Phrasal probability
- Lexical probability
- Inverse phrasal probability
- Inverse lexical probability
- Word count
- Language model
- Distortion model

$\phi(\bar{f}_i|\bar{e}_i)=\phi(\text{la casa} \mid \text{the house})$

$\text{lex}(\bar{f}_i|\bar{e}_i) \propto \text{p(la|the)*p(casa|house)}$

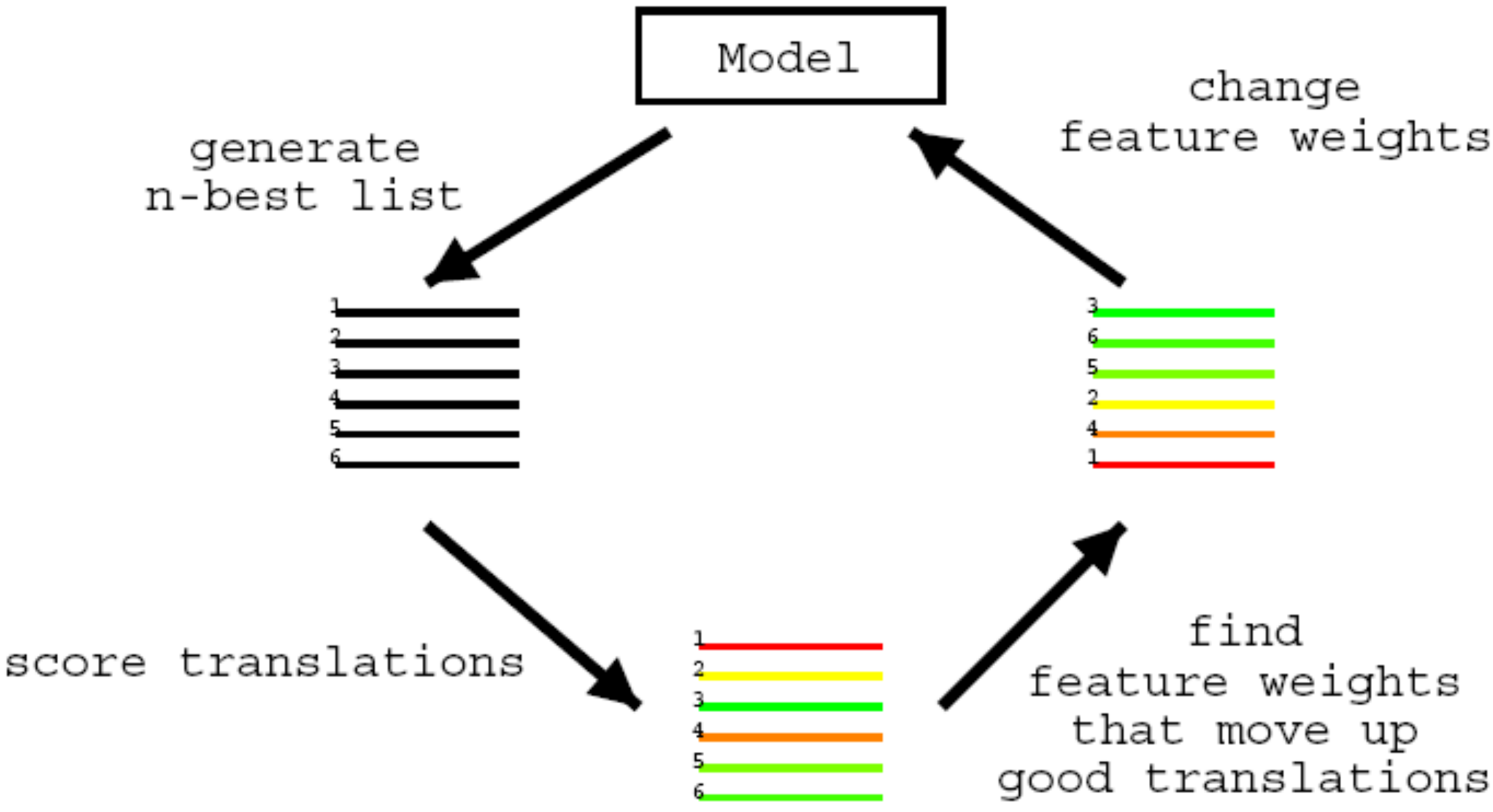
$\phi(\bar{e}_i|\bar{f}_i)=\phi(\text{the house} \mid \text{la casa})$

$\text{lex}(\bar{e}_i|\bar{f}_i) \propto \text{p(the|la)*p(house|casa)}$

$\text{wc} = 1$

$\text{lm}(\bar{e}_i)=\text{lm(the house)}$

Tuning a Log-linear Model



Slide from Philipp Koehn

Some Parameter Optimizers for SMT

	Scales to many parameters?	Fits the typical SMT architecture?	Optimizes
MERT	NO	YES: batch	corpus-level BLEU

Some Parameter Optimizers for SMT

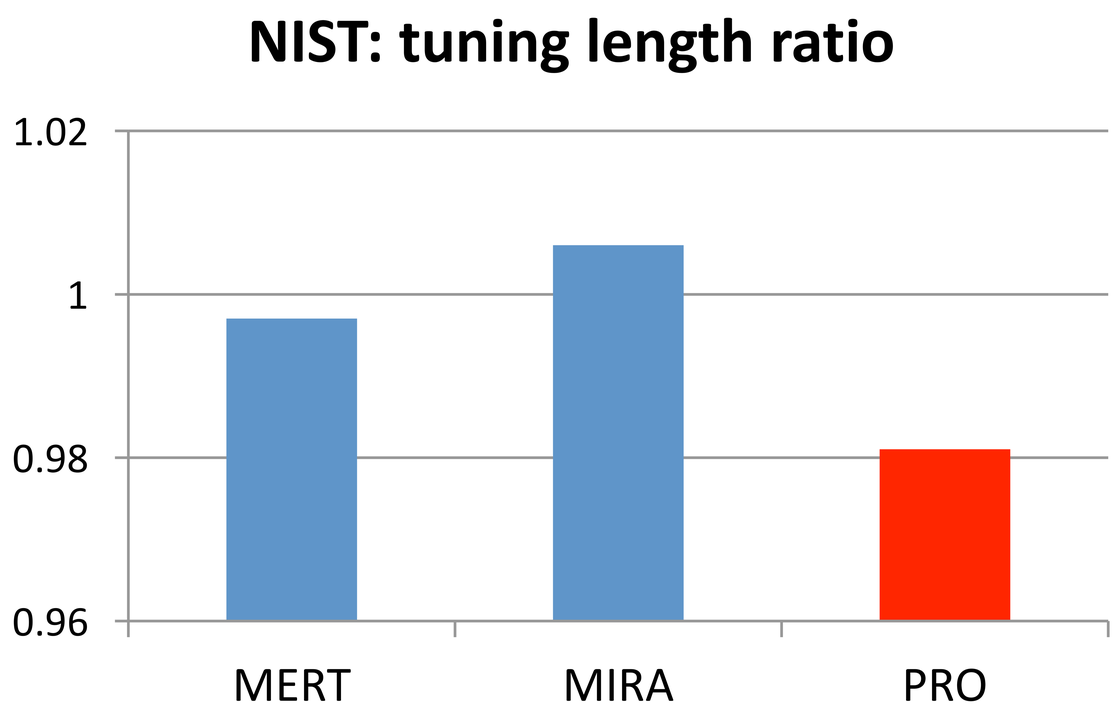
	Scales to many parameters?	**Fits the typical SMT architecture?**	**Optimizes**
MERT	NO	YES: batch	corpus-level BLEU
MIRA	YES	NO: online	corpus-level BLEU (approximation)

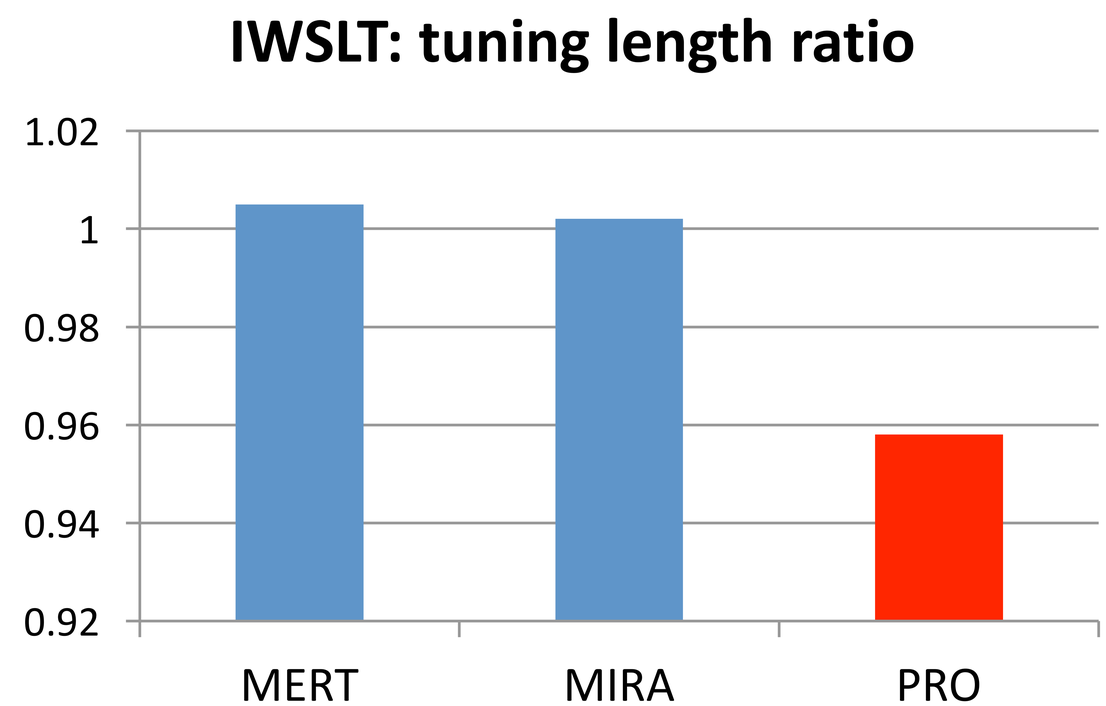
Some Parameter Optimizers for SMT

	Scales to many parameters?	**Fits the typical SMT architecture?**	**Optimizes**
MERT	NO	YES: batch	corpus-level BLEU
MIRA	YES	NO: online	corpus-level BLEU (approximation)
PRO	YES	YES: batch	sentence-level BLEU+1

THE LENGTH PROBLEM WITH PRO

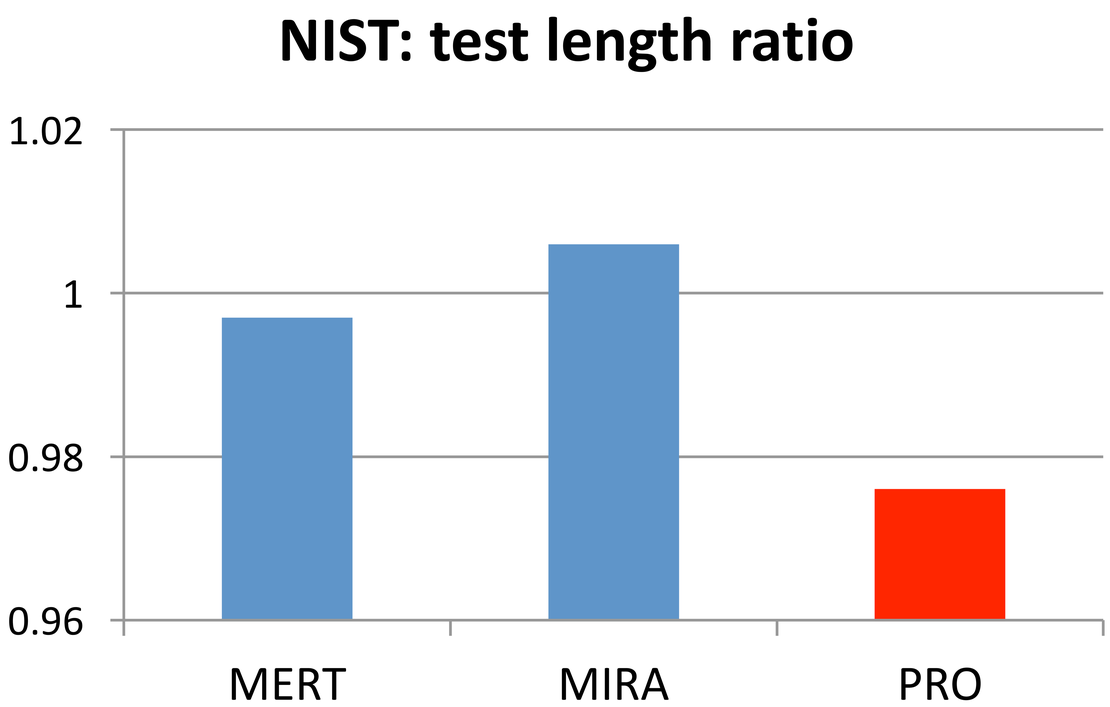
MERT vs. MIRA vs. PRO (on tuning)

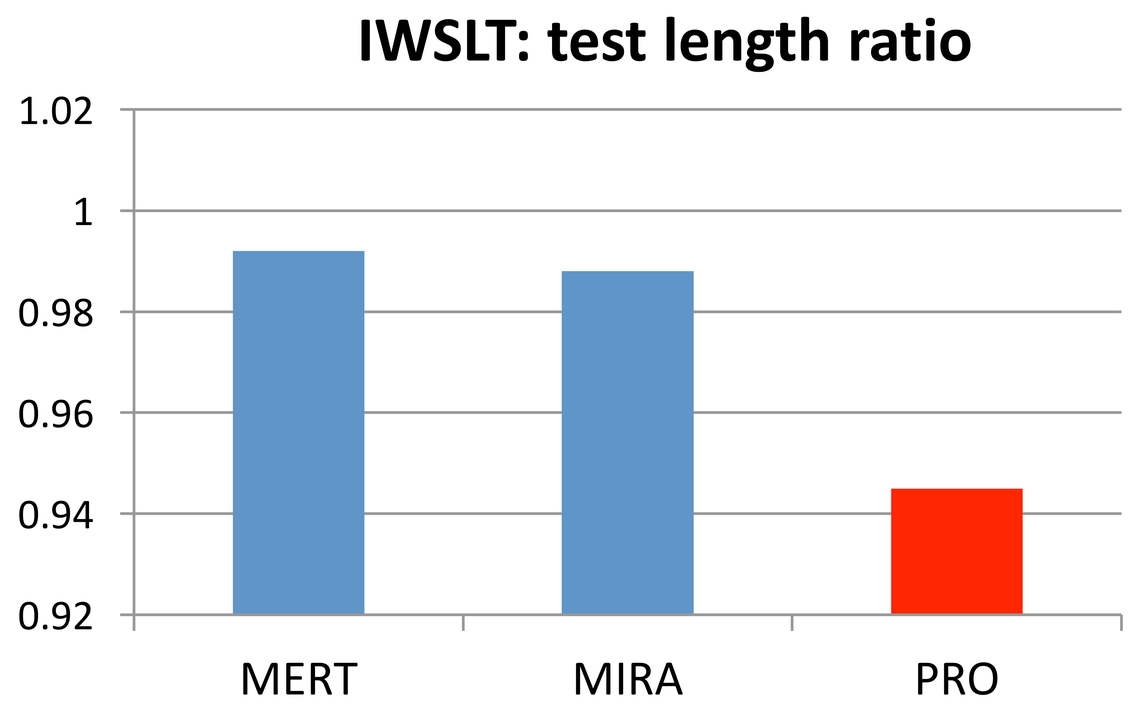




➔ **Issue:** PRO yields too short hypotheses

MERT vs. MIRA vs. PRO (on testing)



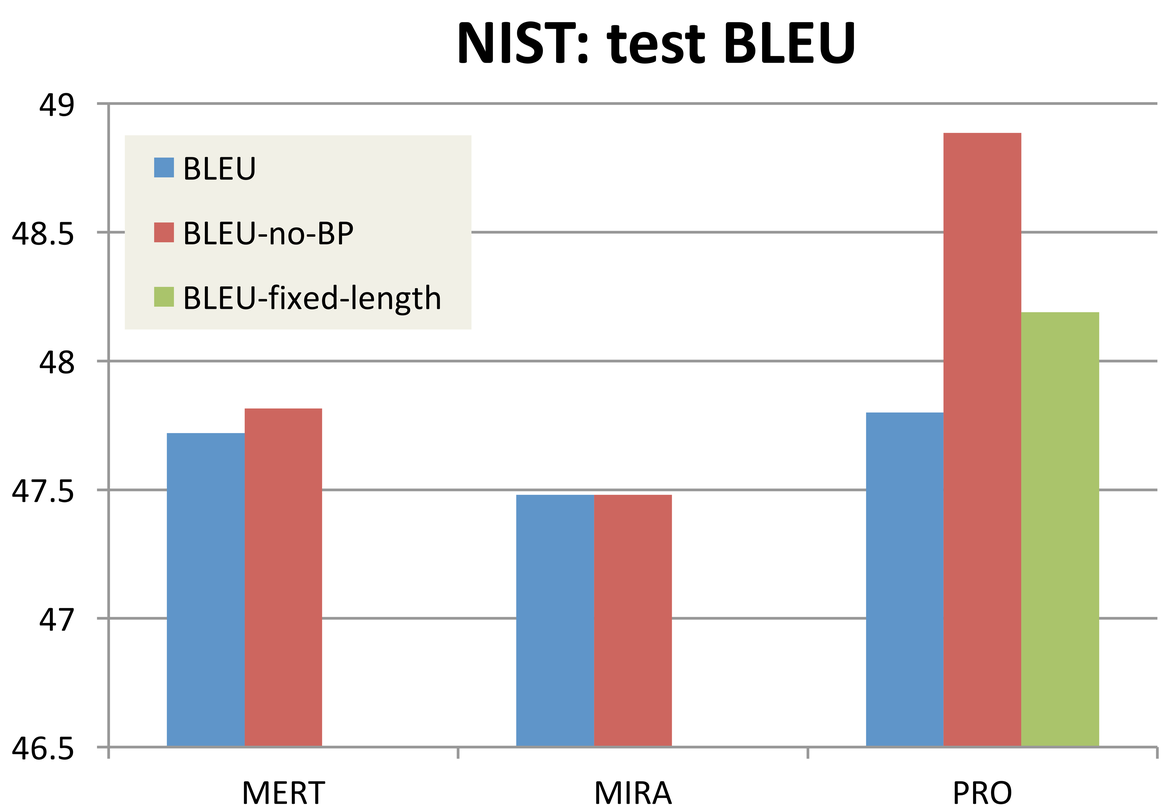


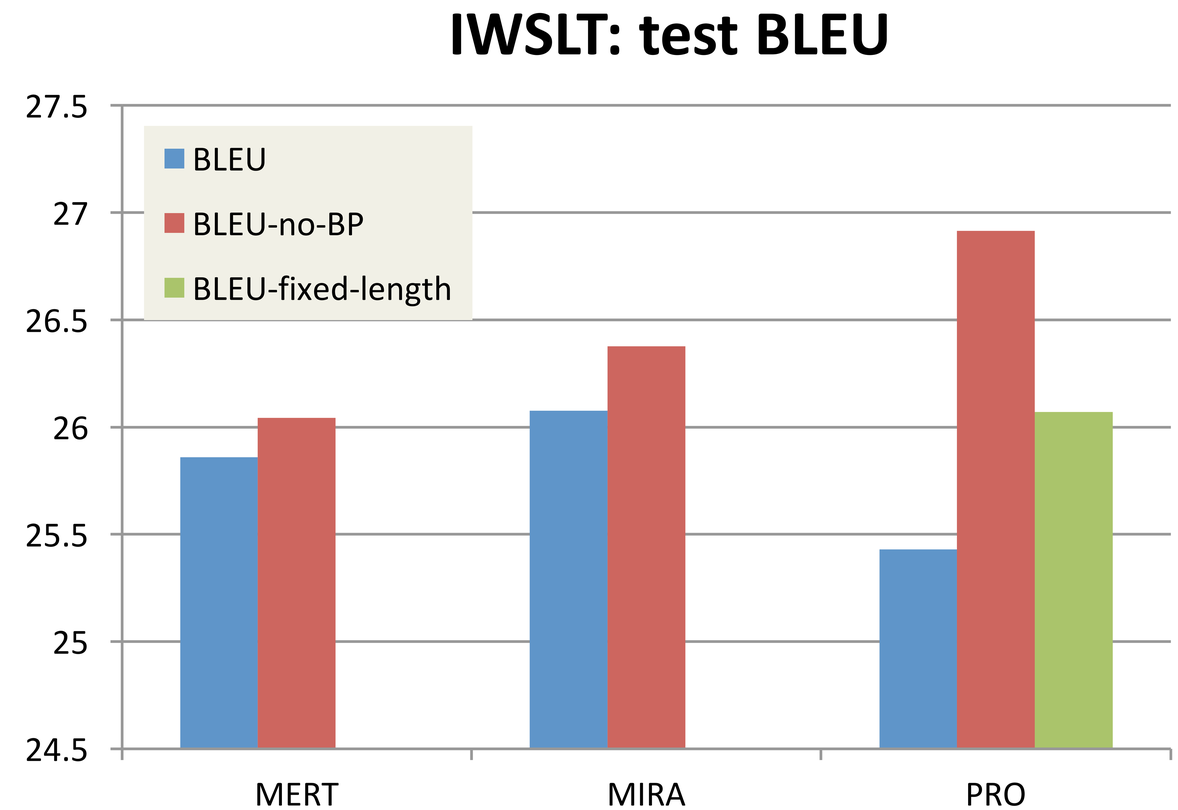
➔ **Issue:** PRO yields too short hypotheses

Length Matters!

- **Short translations**
  - higher precision
  - but heavily penalized by BLEU's BP

- **Goal: increase PRO's length** ➔ reduce BP ➔ improve BLEU





BLEU calculated with multi-bleu (NIST-v13a elsewhere).

Why are PRO's Hypotheses too Short?

- **Possible reasons we explore:**
  1. PRO (its optimization algorithm)
  2. BLEU+1 (vs. BLEU)
  3. Sentence-level optimization (vs. dataset-level)

Experimental Setup

Parameter	NIST 2012	IWSLT 2011
training	ALL but UN (8.6M words)	TED (1.75M words)
tuning	MT06	dev2010
testing	MT09	tst2010
preprocessing	ATB, MADA+TOKAN	ATB, Stanford segmenter
phrase table	grow-diag-final, maximum phrase length of seven tokens, Kneser-Ney (KN) smoothing	
language model	GigaWord v.5, 5-gram, KN-smoothed	(1) English TED, 5-gram, KN-smoothed (2) Europarl + WMT11 News, 5-gram, KN-smoothed
reordering	msd-bidirectional-fe	
pruning	cube pruning pop = 5000	
decoding	monotone-at-punctuation; drop-unknown	
tuning	nbest = 1000, max-iterations = 25	

POSSIBLE REASON 1:
PRO’S OPTIMIZATION ALGORITHM

Possible Reason 1: PRO Itself

PRO's steps *(1-3 for each sentence separately; 4 – combine all)*

1. **Sampling**
   - Randomly sample 5000 pairs
2. **Selection**
   - Choose those whose BLEU+1 diff > 0.05
3. **Acceptance**
   - Accept the top 50 sentence pairs (with max differences)
4. **Learning**
   - Use the pairs for all sentences to train a ranker

1: Remove this threshold
(maybe it selects short-long pairs, where shorter is better)

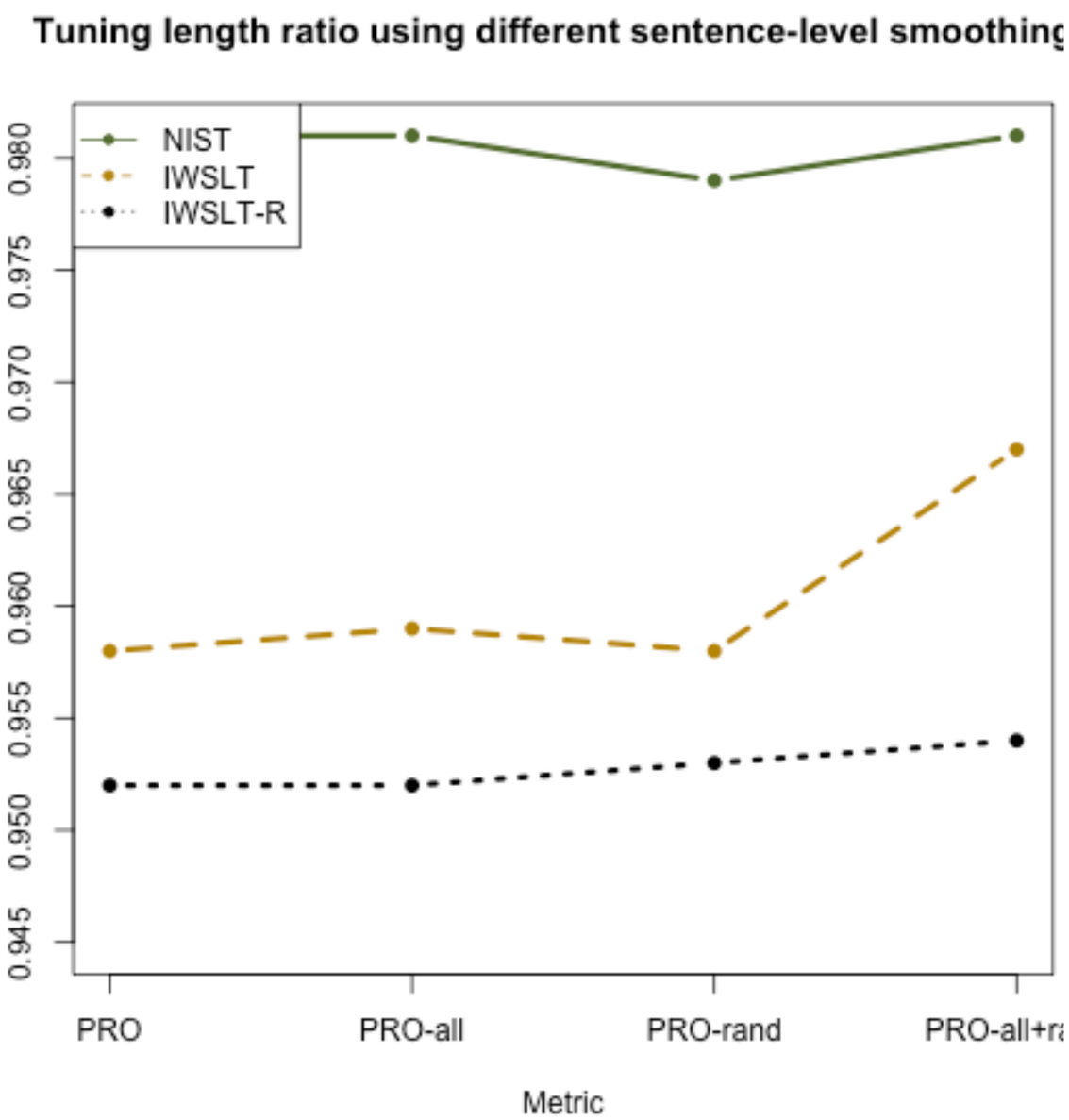
2: Accept a random subset (maybe max diff are short-long pairs?)

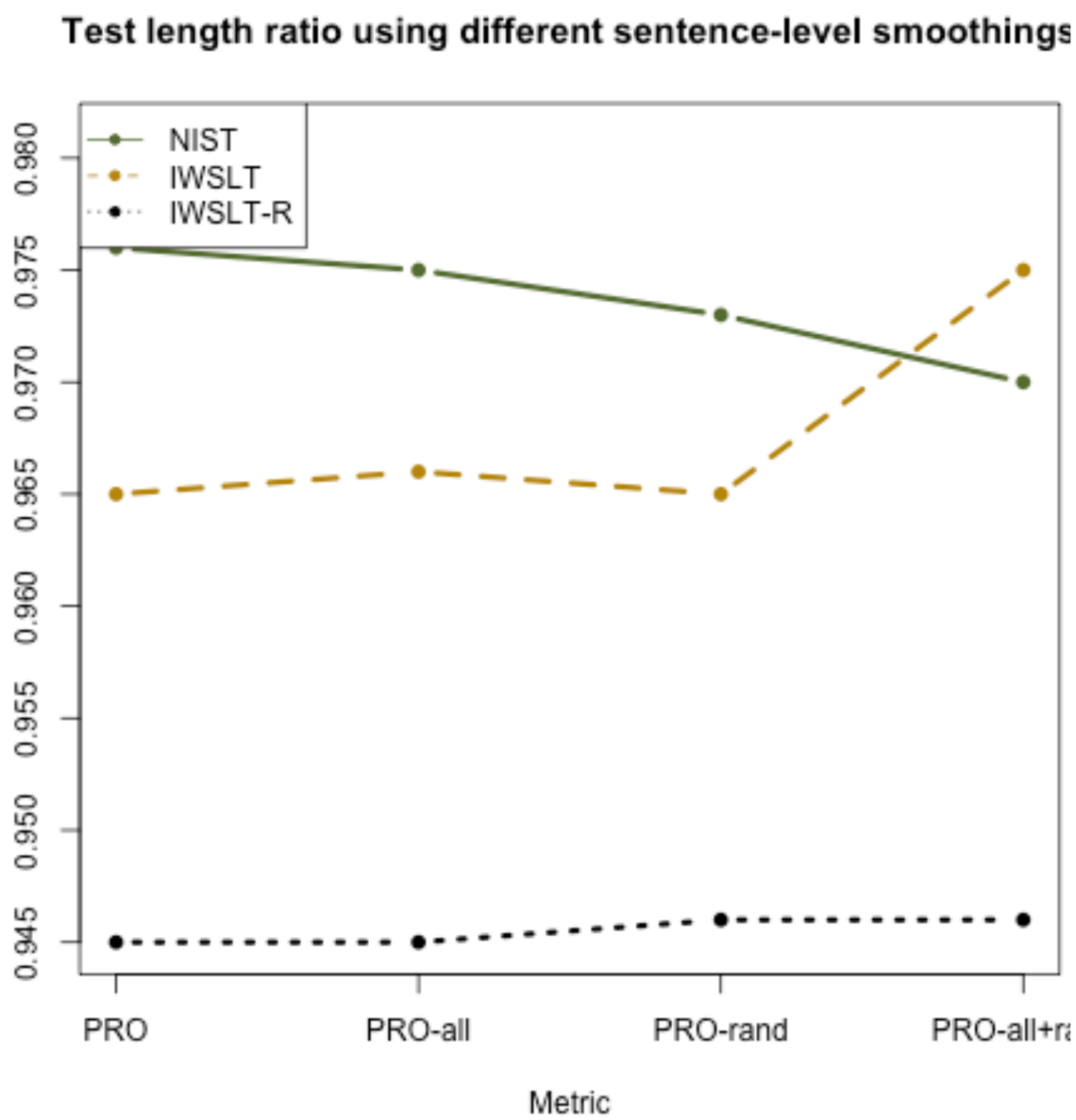
Fixing PRO's Optimization

We experiment with

1. Select all
2. Accept at random
3. Combination of the two

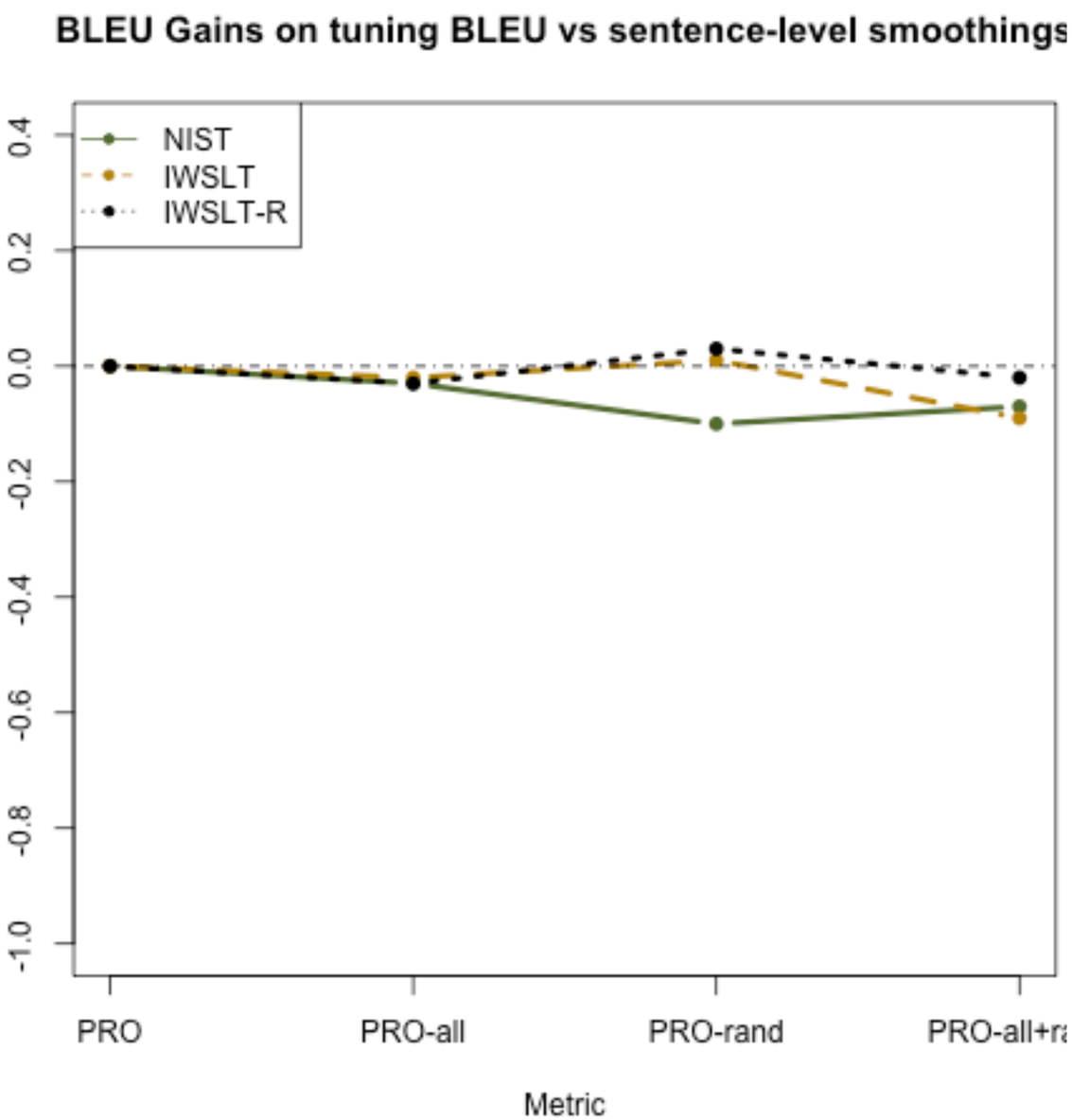
Fixing PRO's Optimization: Length Ratios

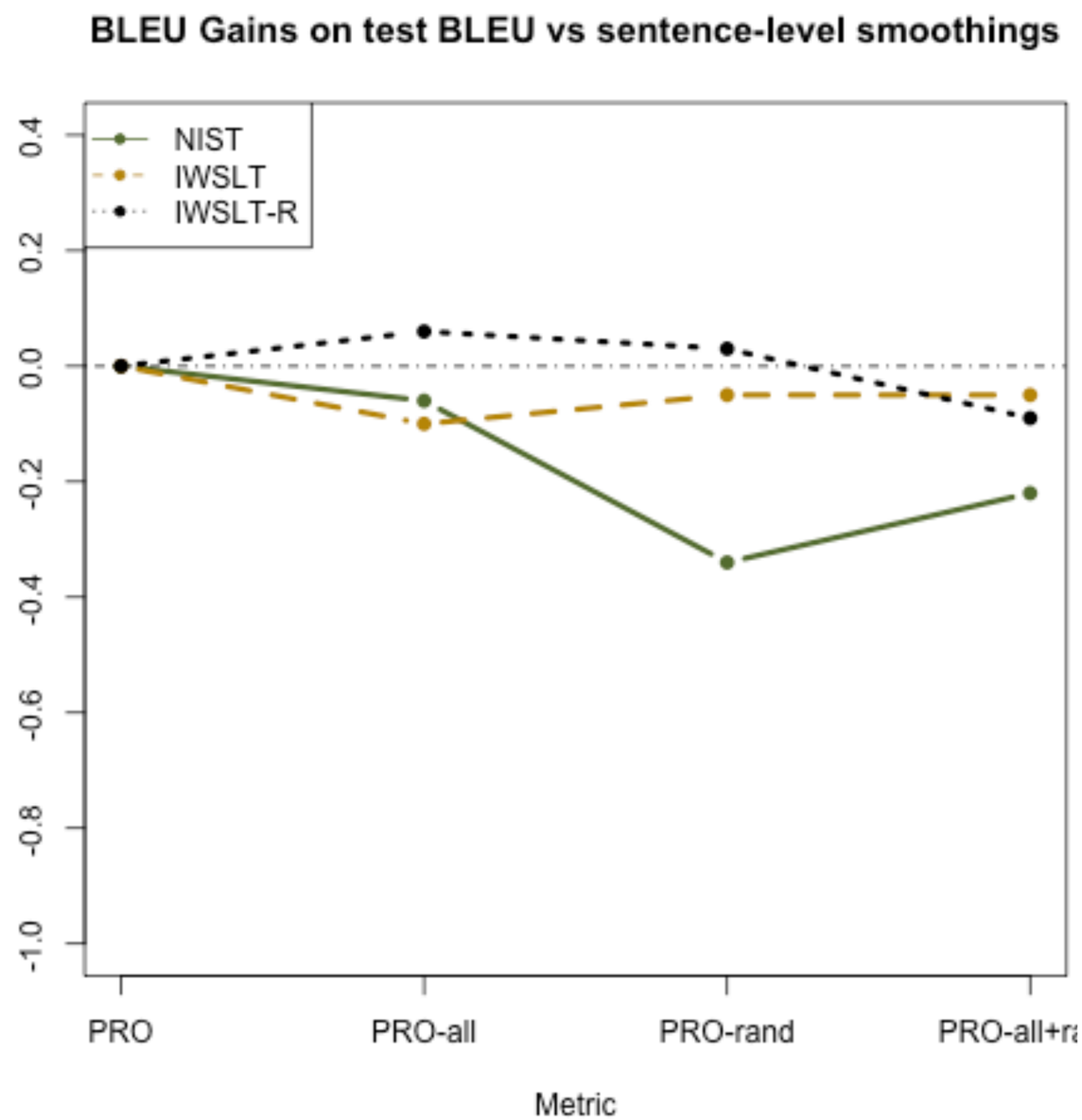




PRO's optimization has generally marginal effect, but large for IWSLT.

Fixing PRO's Optimization: ΔBLEU Scores





Random accept risks focusing on tiny, unimportant distinctions.

POSSIBLE REASON 2: SMOOTHING IN BLEU+1

BLEU

$$\text{BLEU} = BP \cdot \left(\prod_{n=1}^{N} p_n \right)^{\frac{1}{N}}$$

$$p_n = \frac{\sum_i m_{in}}{\sum_i h_{in}}$$

- BP - brevity penalty:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases}$$

$$c = \sum_i c_i \qquad r = \sum_i r_i$$

p_n: n-gram precision
c: candidate's length
r: effective reference corpus length

Sentence-Level BLEU+1 in PRO

For sentence *i*:

$$\text{BLEU} = BP \cdot \left(\prod_{n=1}^{N} p_n \right)^{\frac{1}{N}}$$

$$p_n = \frac{\sum m_n}{\sum h_n}$$

$$p_n = \frac{m_{in}+1}{h_{in}+1}$$

- BP - brevity penalty:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases}$$

$$c = \sum_i c_i \qquad r = \sum_i r_i$$

$$c = c_i \qquad r = r_i$$

p_n: *n*-gram precision
c: candidate's length
r: effective reference corpus length

Problems with Smoothing in BLEU+1

- **BLEU+1 alters precision, but not BP**
  - destroys the balance: precision gets more weight
  - BLEU+1 boosts shorter sentences more

- **BLEU+1 is strictly positive**

Fixing BLEU+1's Smoothing

1. **Use unsmoothed BLEU**

2. **Ground the precision component**

$$PC = \left(\prod_{n=1}^{N} \frac{m_{in}+1}{h_{in}+1}\right)^{\frac{1}{N}} - \left(\prod_{n=1}^{N} \frac{1}{h_{in}+1}\right)^{\frac{1}{N}}$$

- Helps restore the PC-BP balance.
- Allows BLEU+1 to be 0.

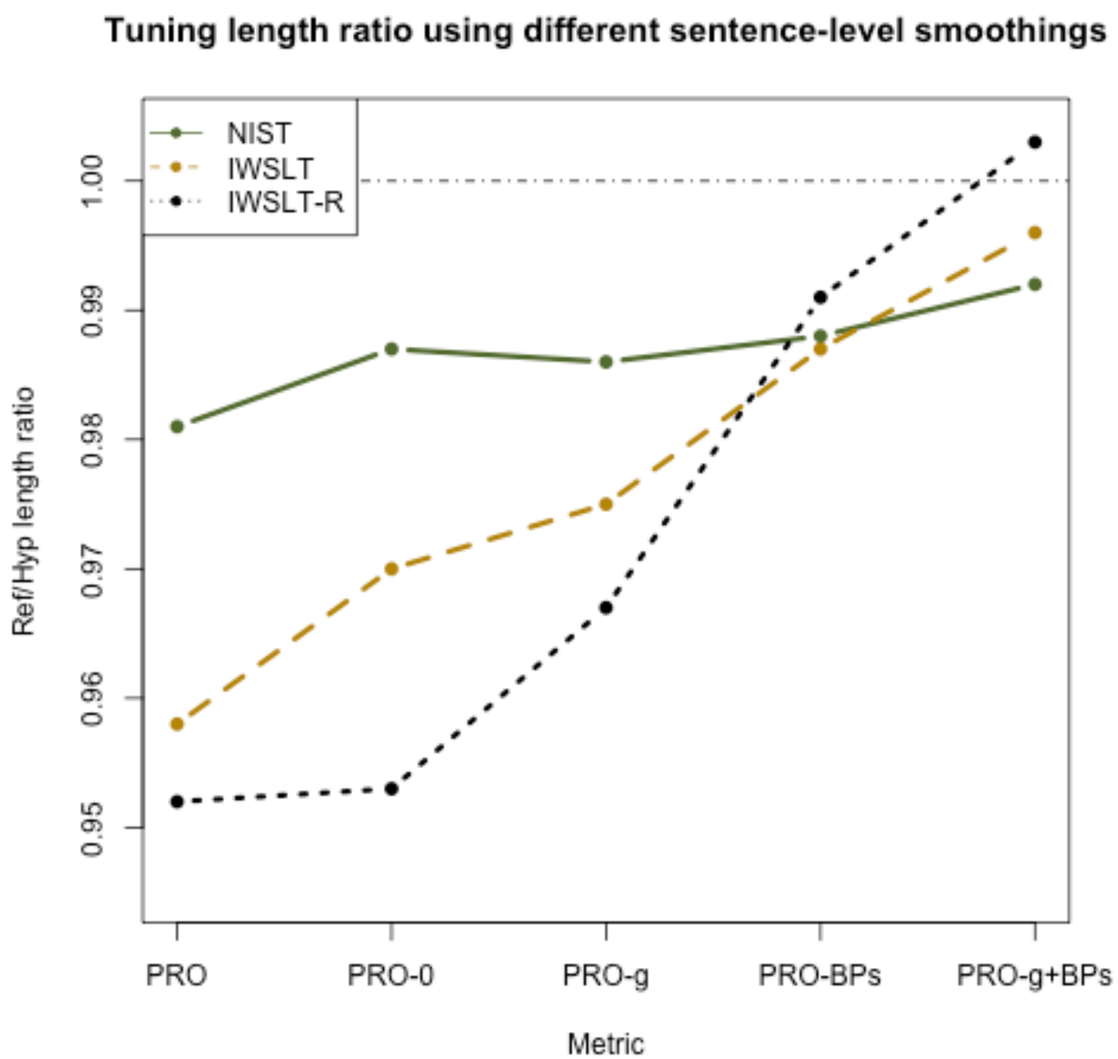
3. **Smooth BP**

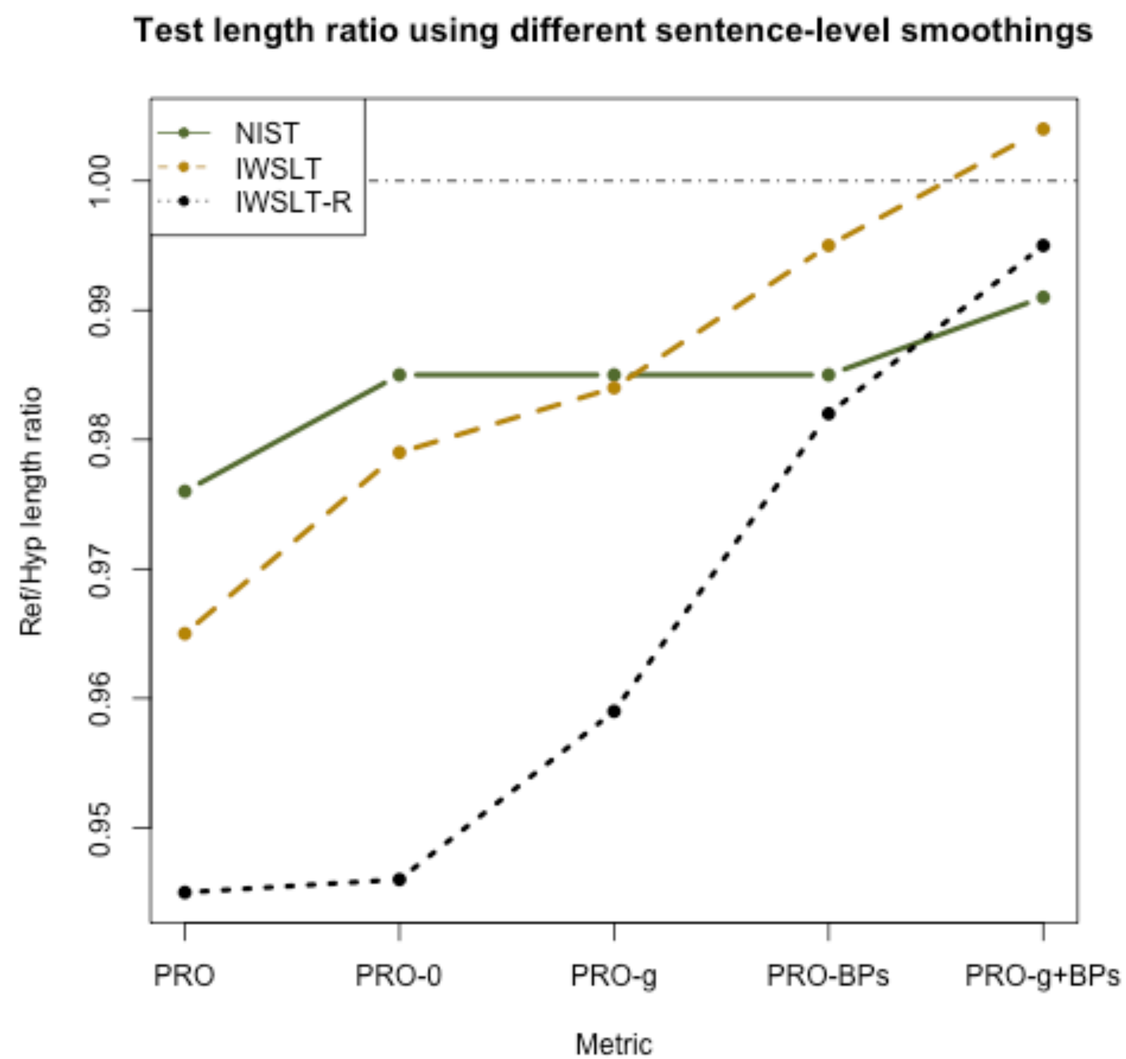
$$r = r_i + 1$$

If we assume an extra match, we should also assume an extra reference word.

4. **Do both 2 & 3**

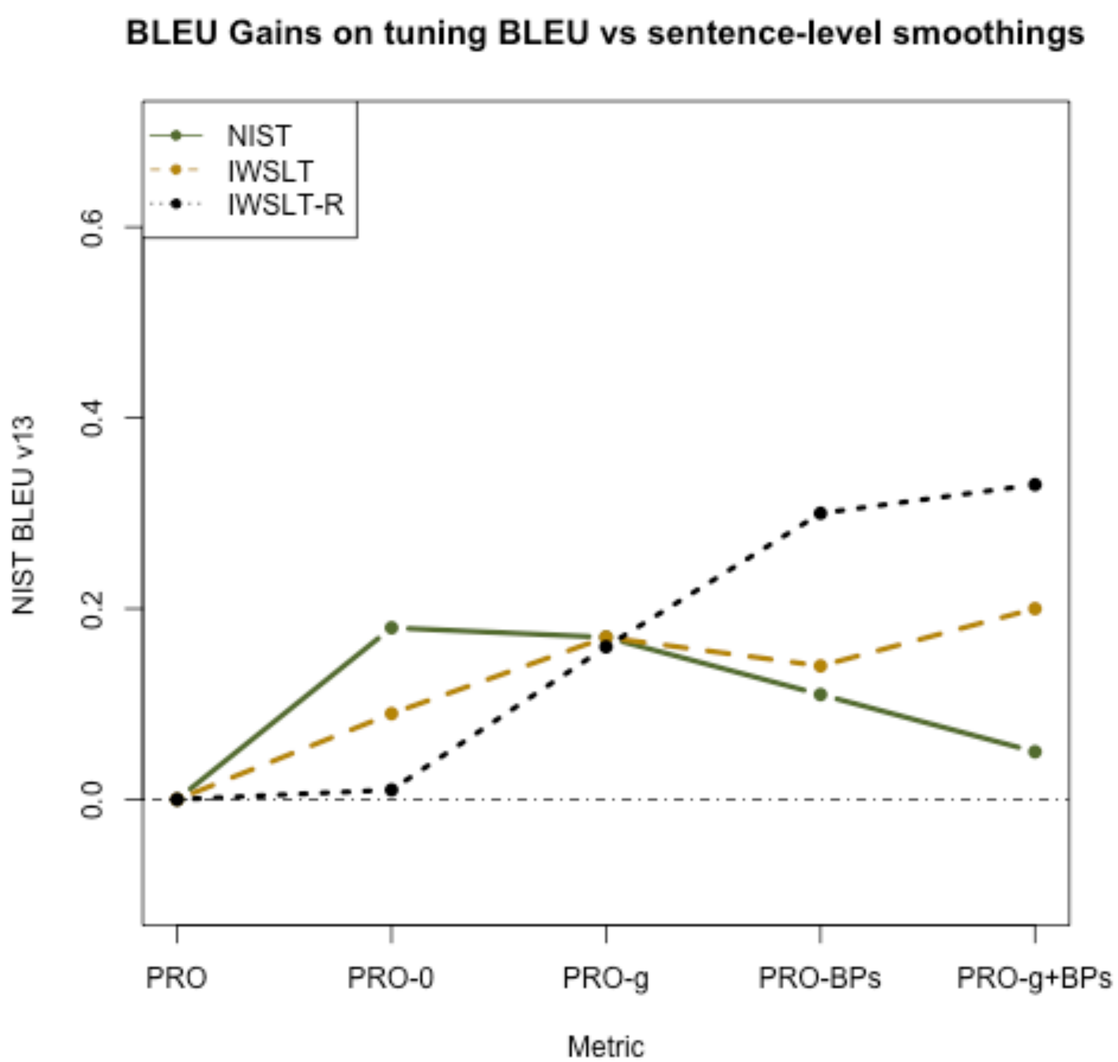
Fixing BLEU+1 Smoothing: Length Ratios

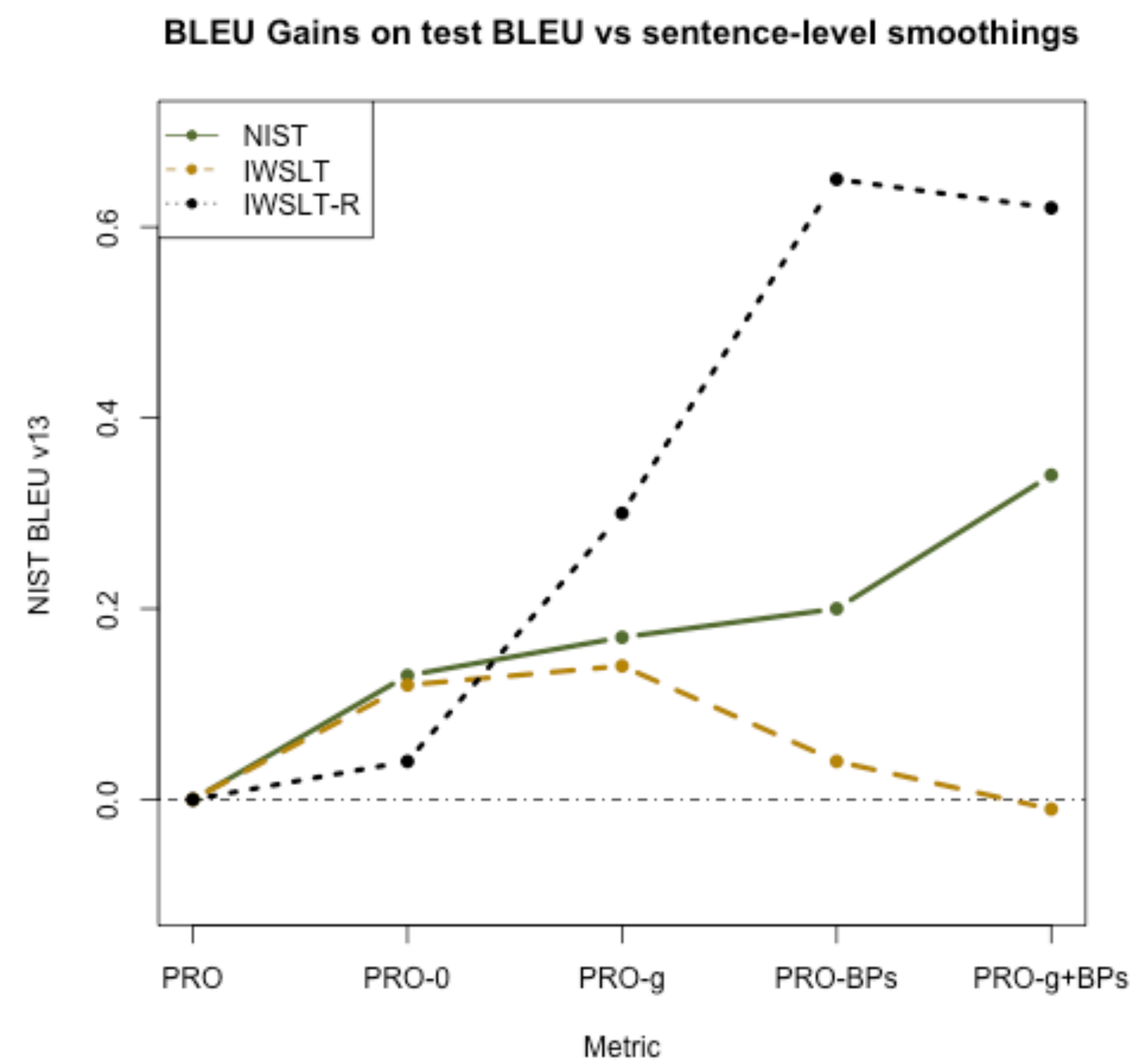




Fixing BLEU+1 smoothing does improve the length ratio for PRO!

Fixing BLEU+1 Smoothing: ΔBLEU Scores





ΔBLEU: up to +0.65 for IWSLT, 0.34 for NIST.

POSSIBLE REASON 3: SENTENCE-LEVEL OPTIMIZATION

Possible Reason 3: Sentence-Level Optimization

- **With BLEU+1, PC weighted higher than BP**

  ➔ encourages staying shorter: PC↑, BP↓

- **No global view of length**
  - Each sentence penalizes BP in isolation

    ➔ Individual sentences are reluctant to getting longer than the reference, even if this can help global BP (and BLEU).

Fixing Sentence-Level Optimization

1. **Scale BP** *(as of the previous iteration of PRO)*

$$r = r_i \quad \rightarrow \quad r = r_i \cdot ILR$$

$$ILR = c^{(-1)} / r^{(-1)} = \sum_j c_j^{(-1)} / \sum_j r_j^{(-1)}$$

2. **Unclip BP**

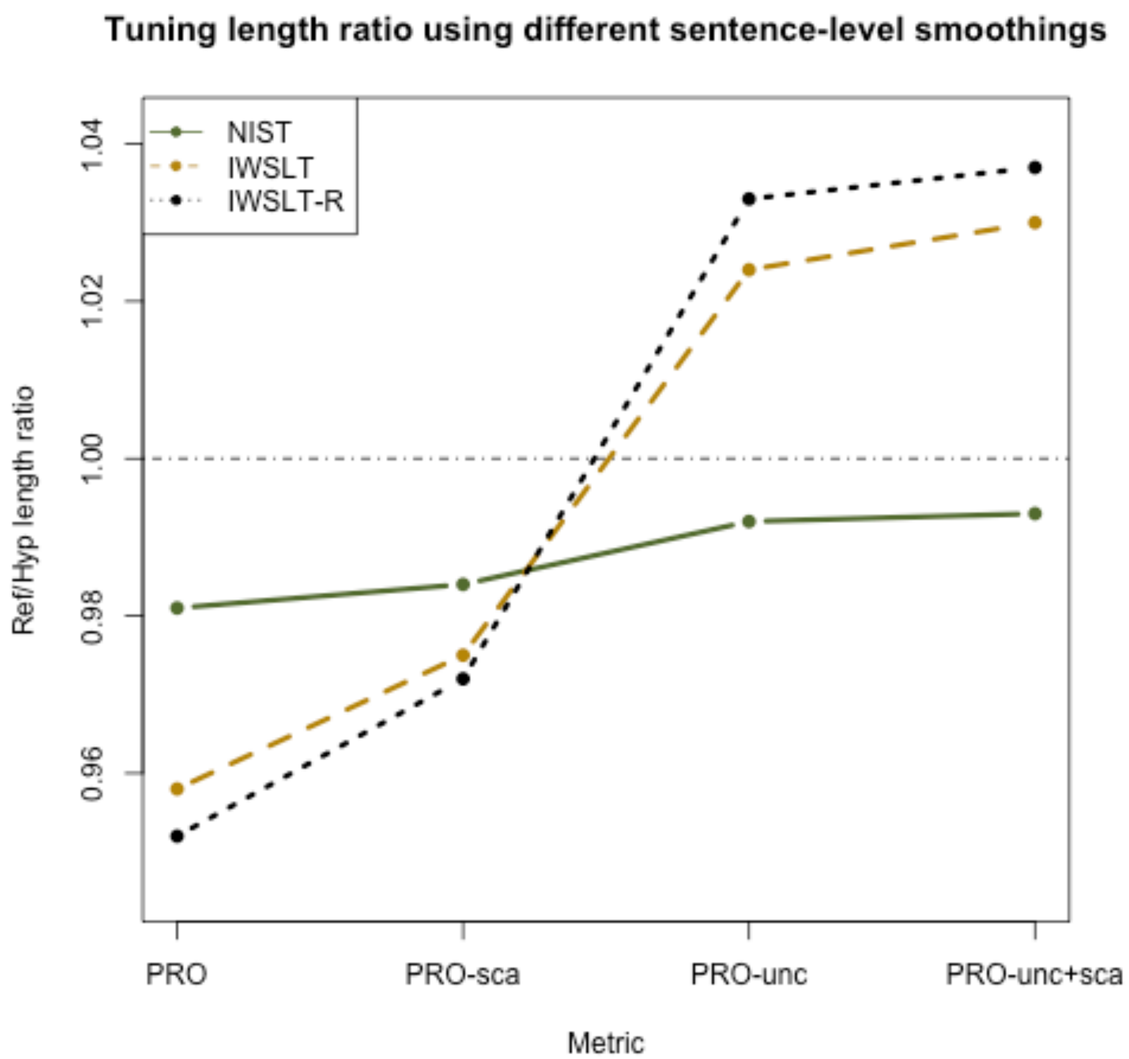
$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \quad \rightarrow \quad BP = \exp\left(1 - \frac{r}{c}\right)$$

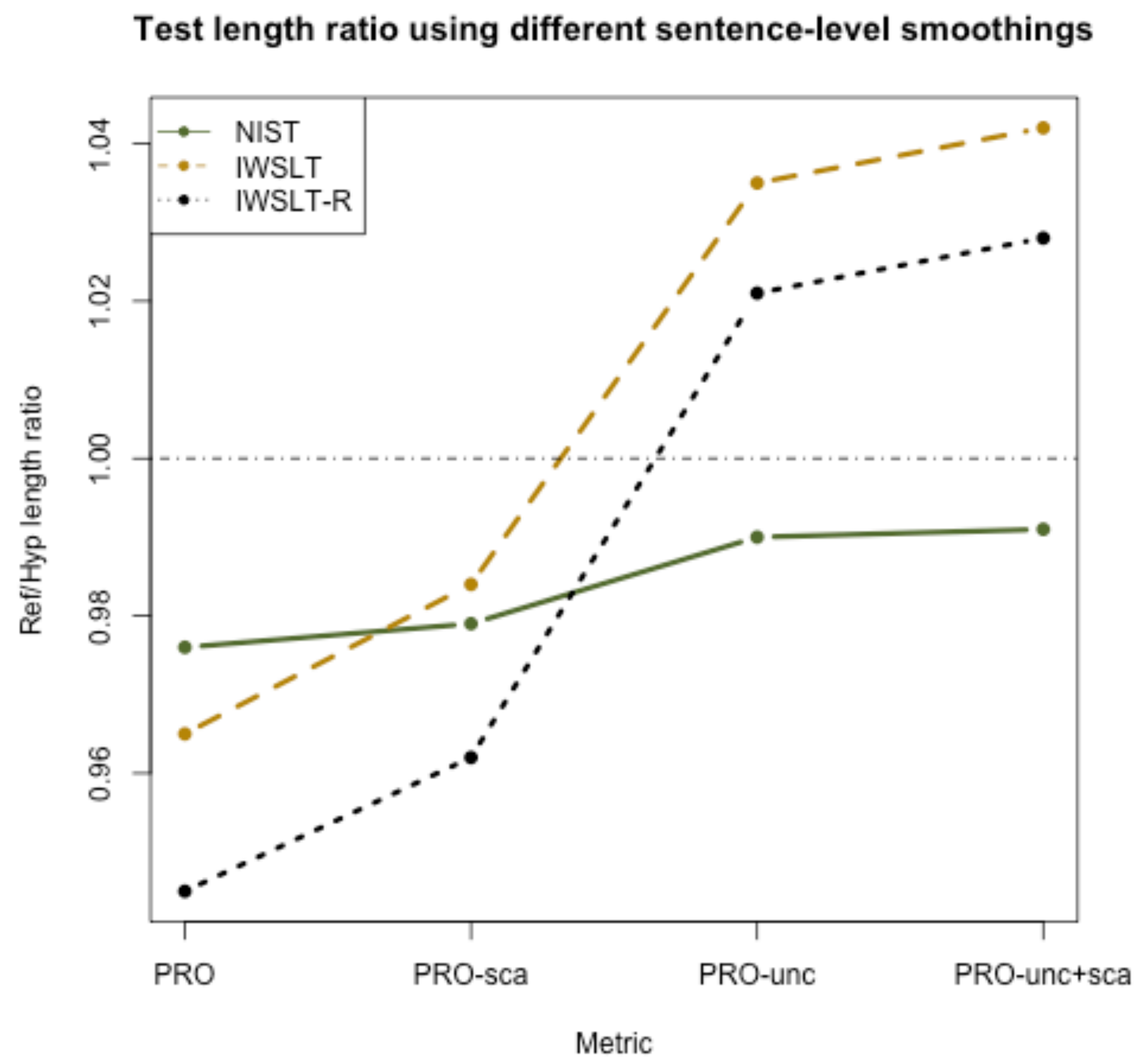
3. **Scale & unclip BP**

Advocated by He & Deng (2012)

4. **Use corpus-level BLEU** (MIRA-style)

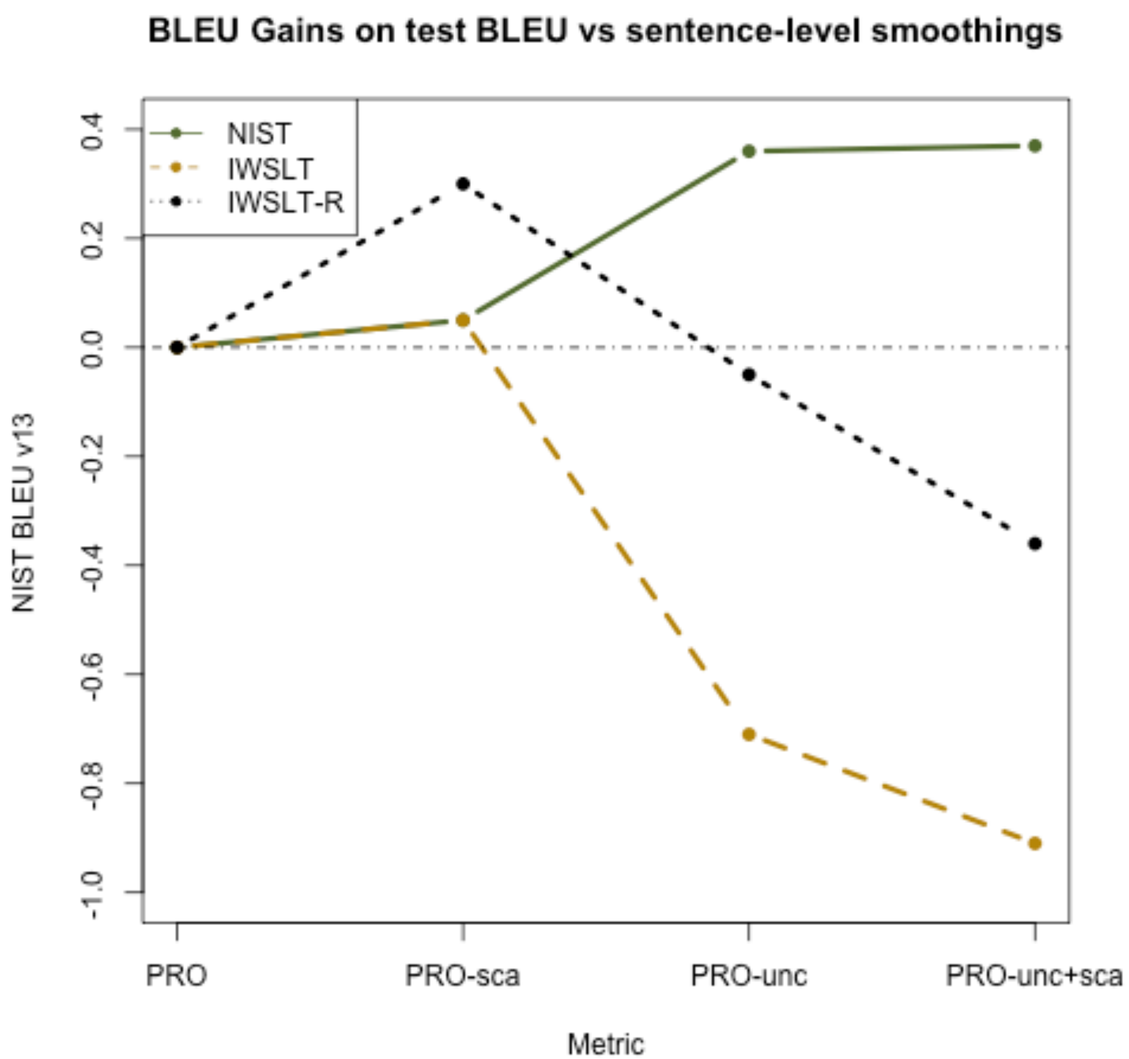
Fixing Sentence-Level Optimization: Length Ratios

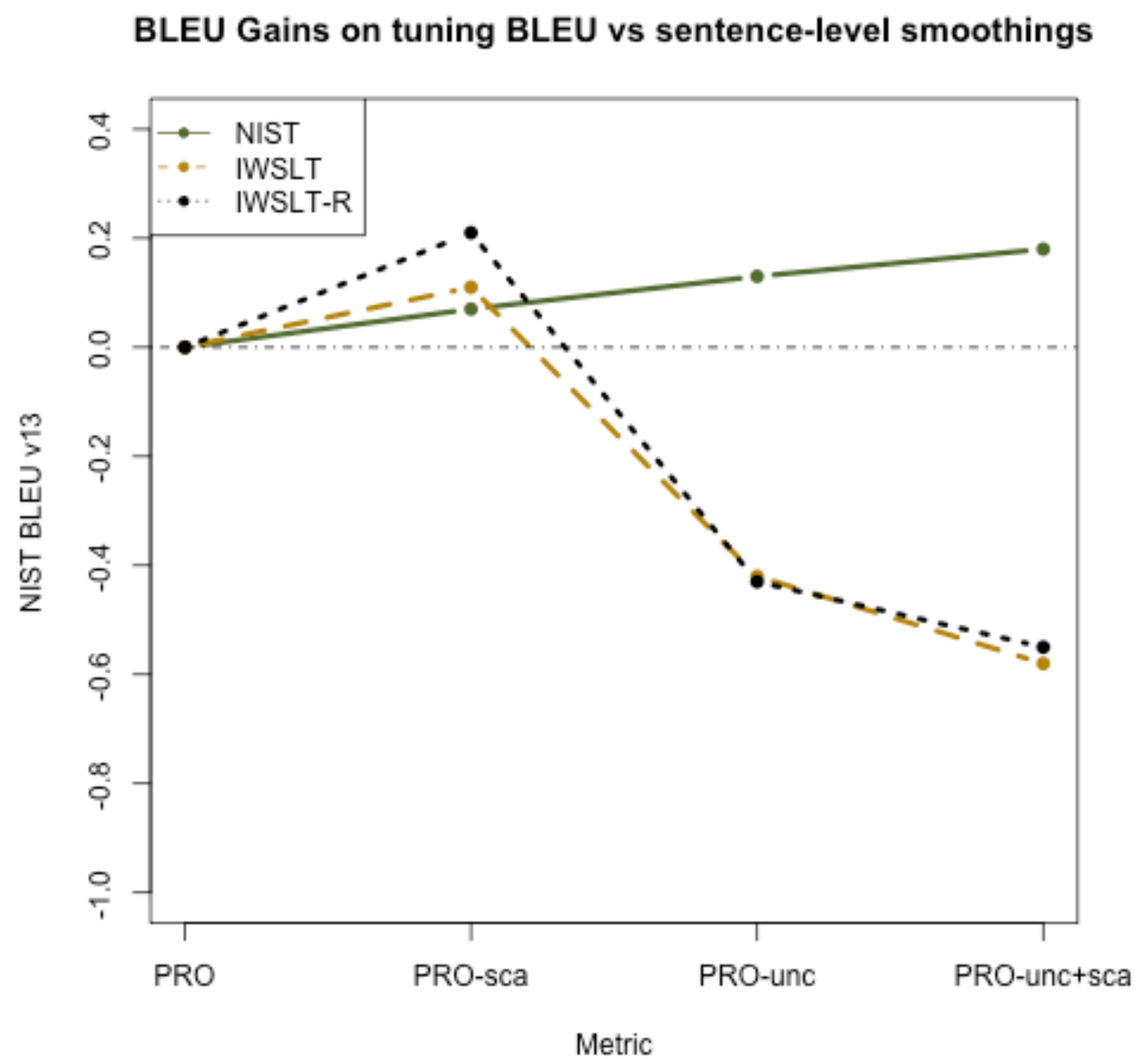




Scaling BP is good for all datasets.
Unclipping BP helps with 4 refs, but overcompensates with 1 ref.

Fixing BLEU+1 Smoothing: ΔBLEU Scores





Scaling BP is good for all datasets.
Unclipping BP helps with 4 refs, but overcompensates with 1 ref.

Sentence-Level vs. Dataset-level Optimization

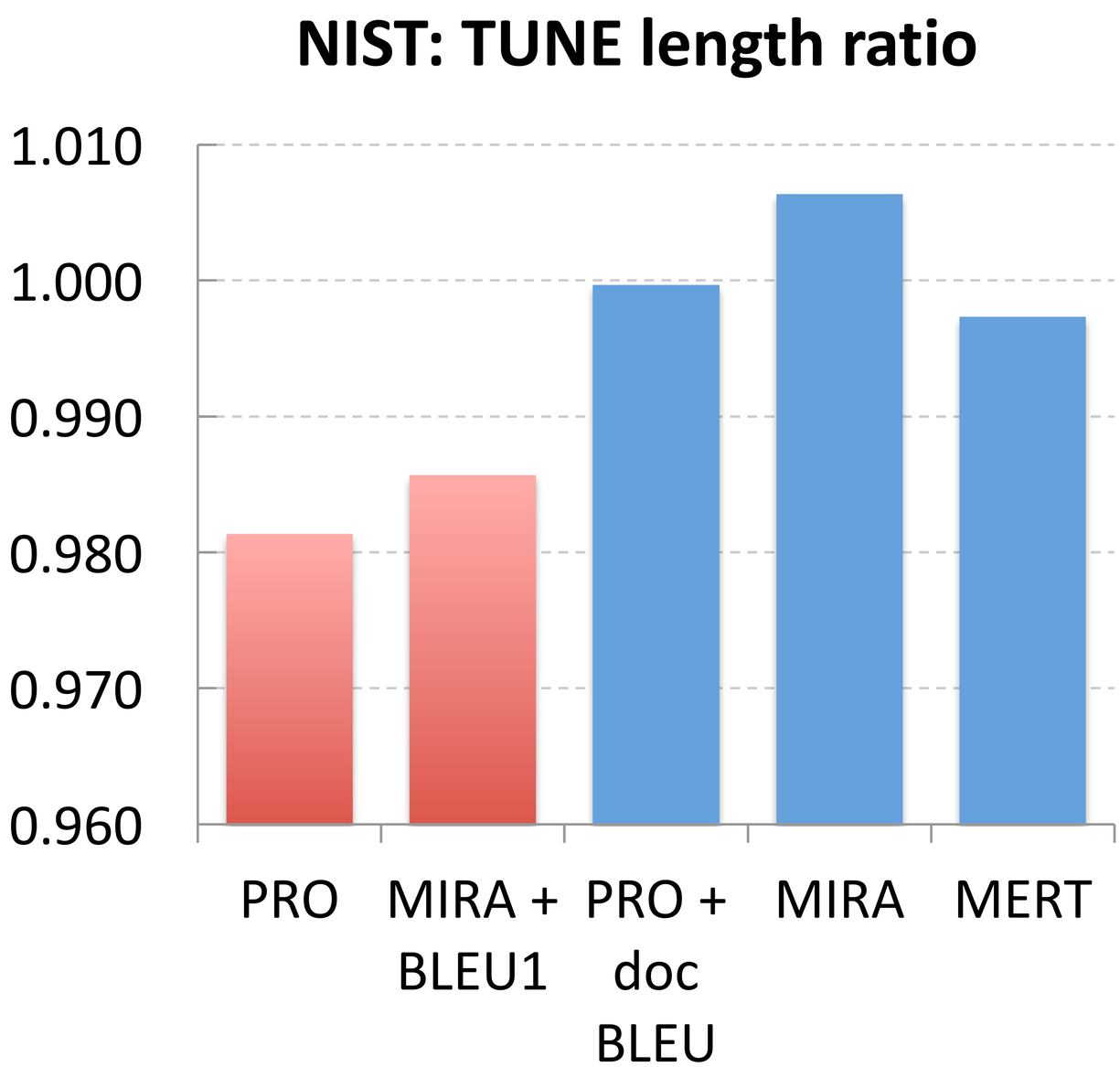
DISCUSSION

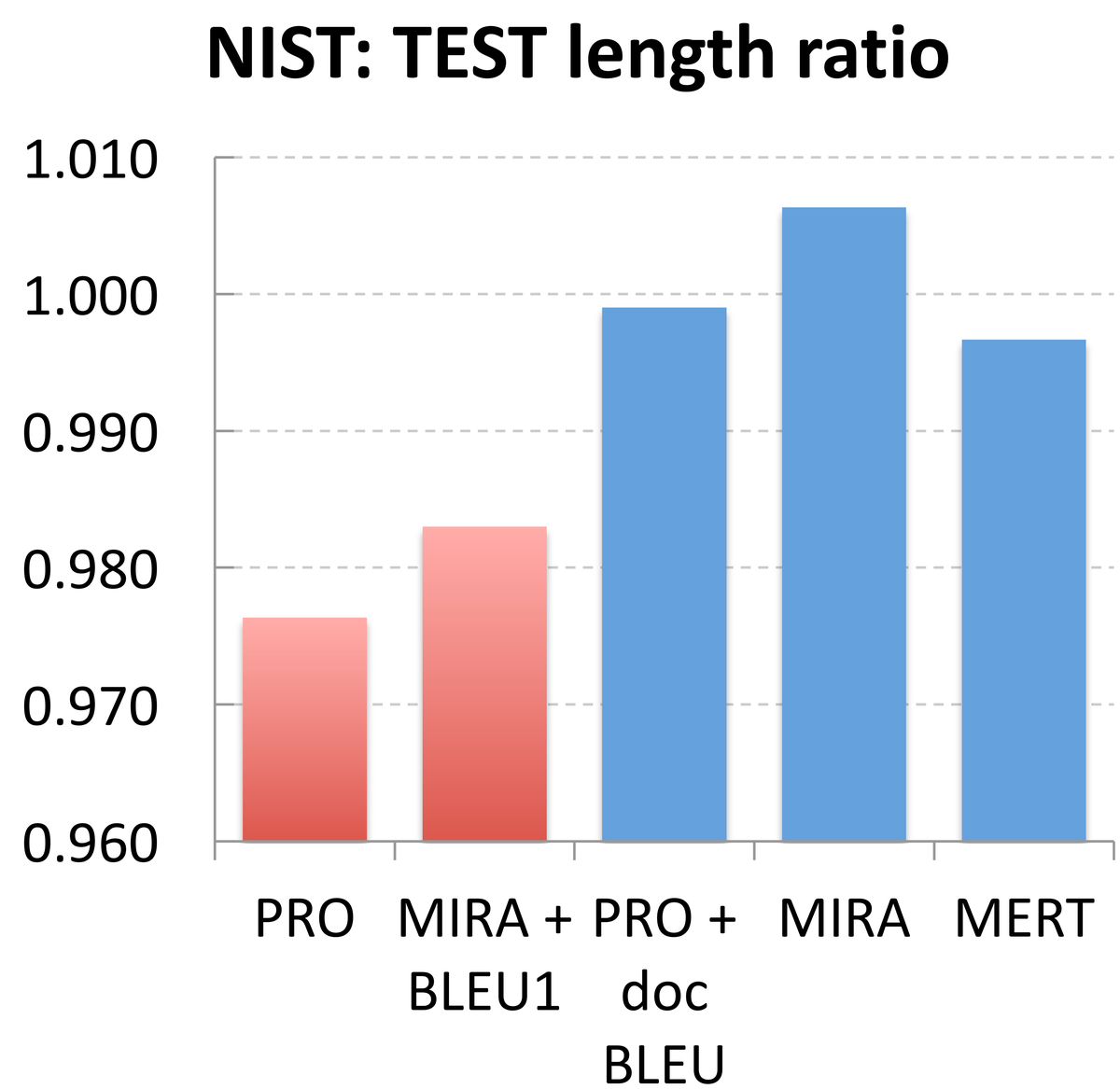
Discussion

- **Objective function (form of BLEU+1) matters**, e.g.
  - grounding PC
  - smoothing BP
- **BUT PRO's optimization does not**

- **<u>So, does the optimizer matter?</u>**
  - Try the same objective function in different optimizers:
    - **BLEU:** MIRA, <u>doc-PRO</u>, *(also MERT)*
    - **sent-BLEU+1:** PRO, <u>sent-MIRA</u>

Two Groups: Objective-Based

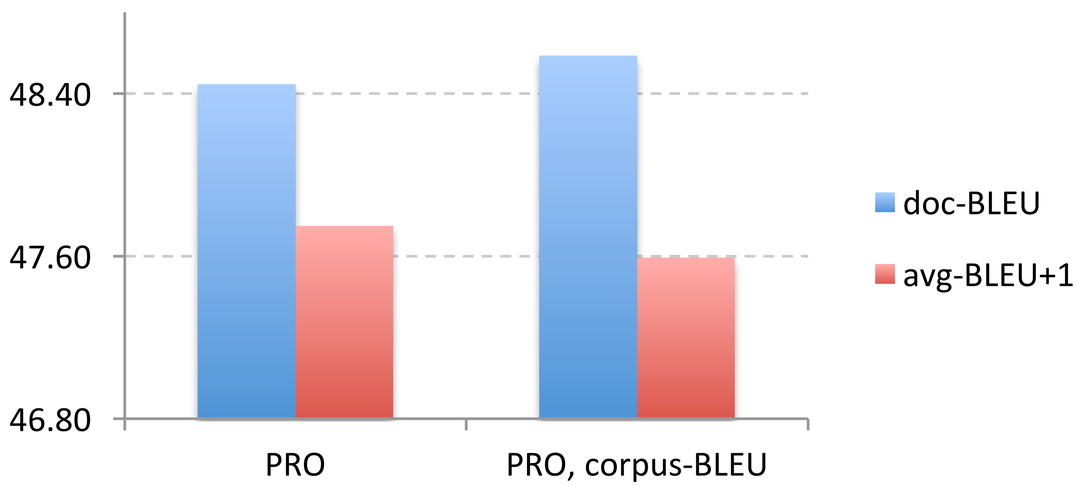




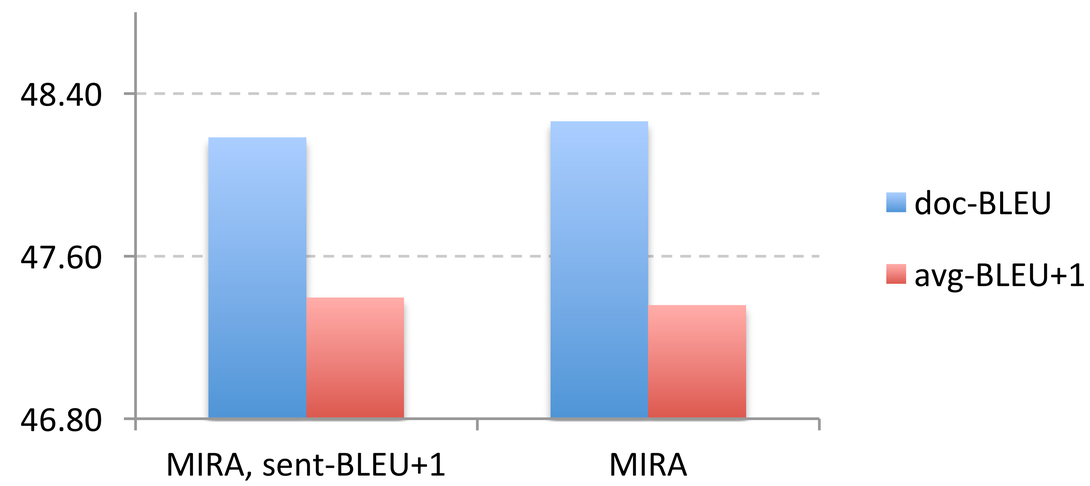
Differences in Objective

PRO

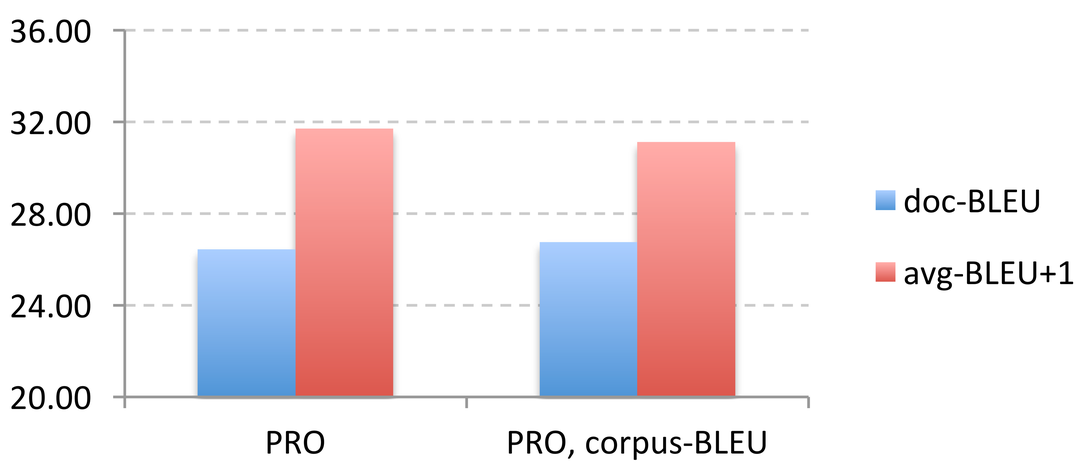
NIST: test

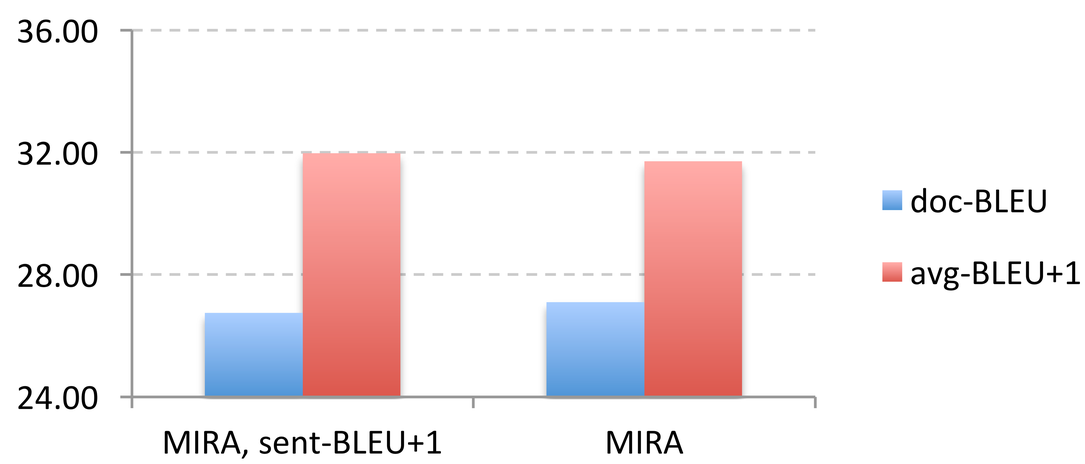


MIRA



IWSLT: test





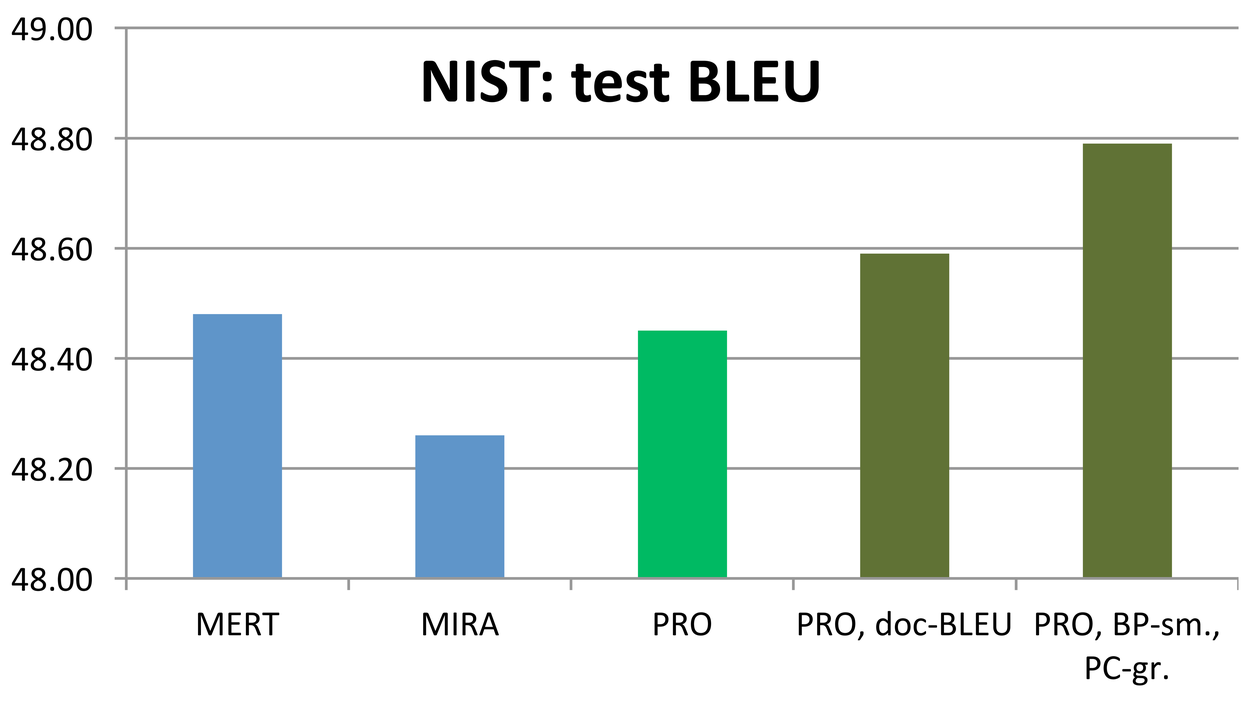
CONCLUSION

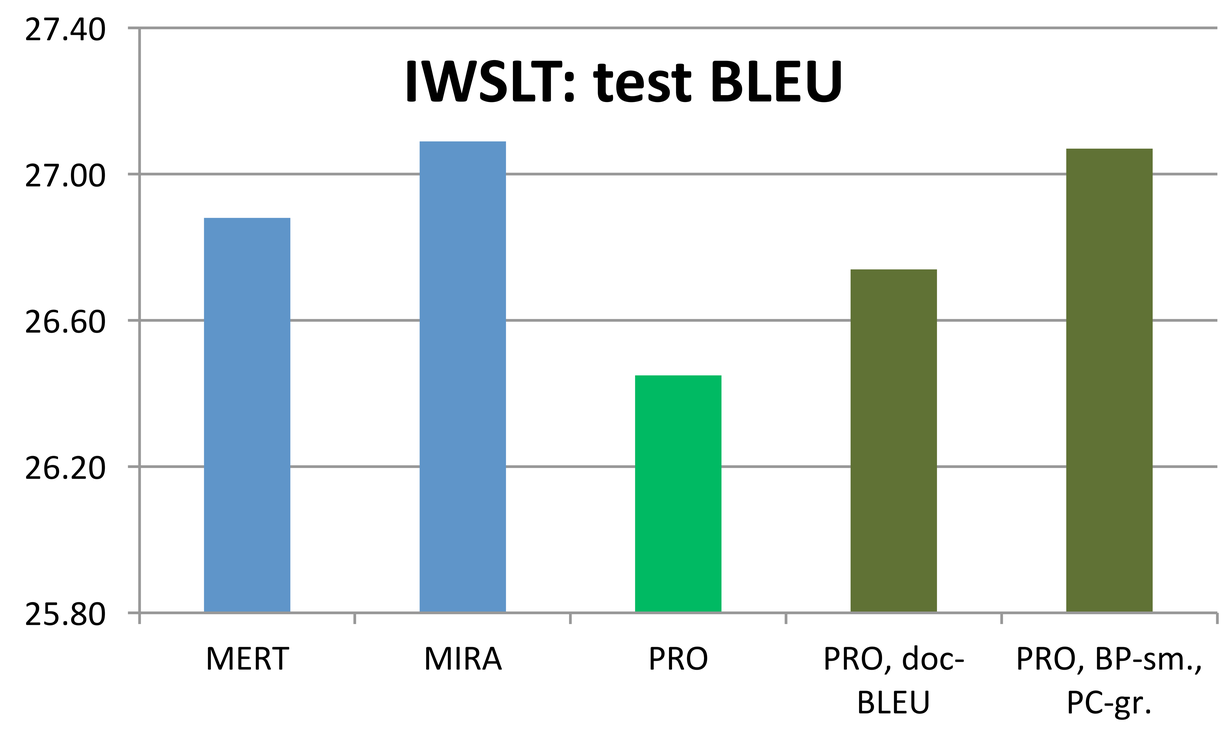
Summary

- **Problem:**
  - PRO yields too short translations (➔ lower BLEU)
- **Possible reasons we explored:**
  - Sampling bias in PRO → **NO**
  - Smoothing in BLEU+1 → **YES**
  - Sentence-level optimization → **YES**
- **Solution ➔** yielded improved BLEU on two datasets
  - better smoothing
  - document-level optimization
- **General finding**
  - Optimizing for sentence-level BLEU+1 yields too short translations
    - demonstrated for variations of MERT, MIRA and PRO

Conclusion

- **Frustration in the community**
  - PRO handles many features, but MERT is still better in many cases.
- **One reason we found**
  - *sentence-level BLEU+1 yields too short hypotheses*
    - ✓ we fixed the length ➔ improved BLEU
    - ✓ preserved the simple sentence-level nature of PRO





Future Work

- **Further study**
  - *optimizers* vs. *objective functions* *(e.g., TER, METEOR)*
  - sentence-level vs. corpus-level optimization *(combine?)*
  - characterize the hypotheses from different optimizers

Thank you!