Class-Based N-gram Language Difference Models for Data Selection

Amittai Axelrod — University of Maryland & Johns Hopkins

Yogarshi Vyas
Marianna Martindale
Marine Carpuat
— University of Maryland

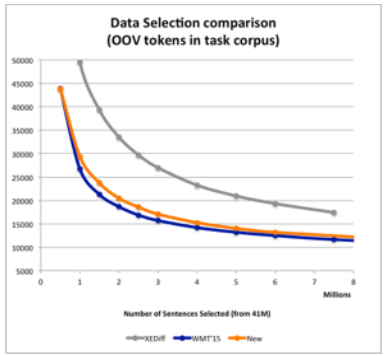
Class-Based N-gram Language Difference Models for Data Selection

Amittai Axelrod — University of Maryland & Johns Hopkins

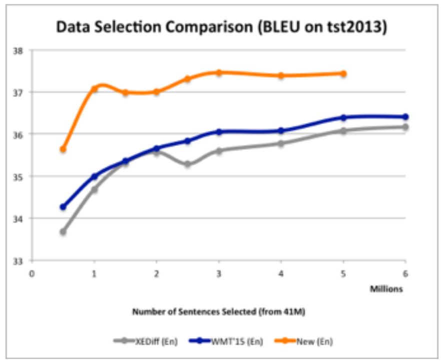
Yogarshi Vyas
Marianna Martindale
Marine Carpuat
} University of Maryland

Outline

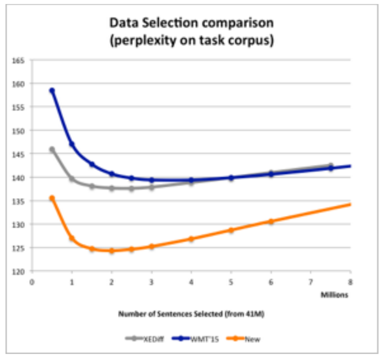
- Context: Domain adaptation + data selection
- Motivation: Inefficiency of cross-entropy difference
- Idea: Move external information into the model
- Implementation: Language difference models
- Results:



-35% OOV



+1.5 BLEU



-10% ppl

12.000 gb LM

↓

0.126 gb

-99% size

Domain* Adaptation

- Ideally:
  "Domain" = defined by some notion of language similarity: topic, lexical choice, style, genre, register, intent, *etc.*

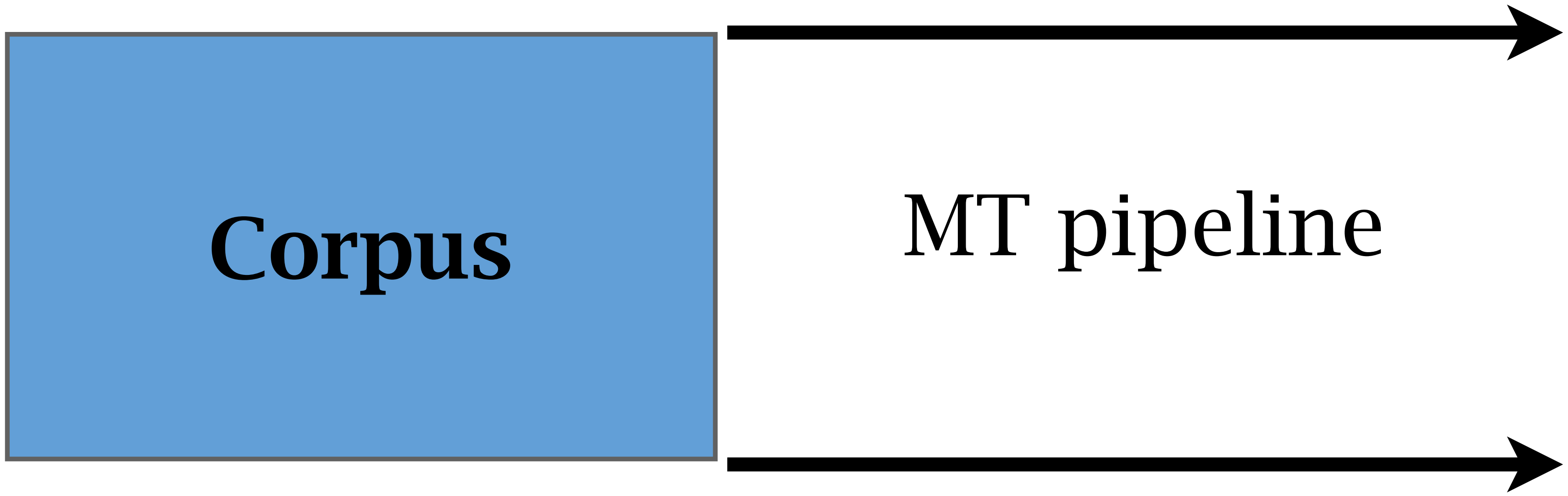
- In practice:
  "Domain" = "particular contextual setting", defined empirically. "in corpus" = "in domain"

- For clarity, we use “domain” to mean “corpus”.

Domain Adaptation

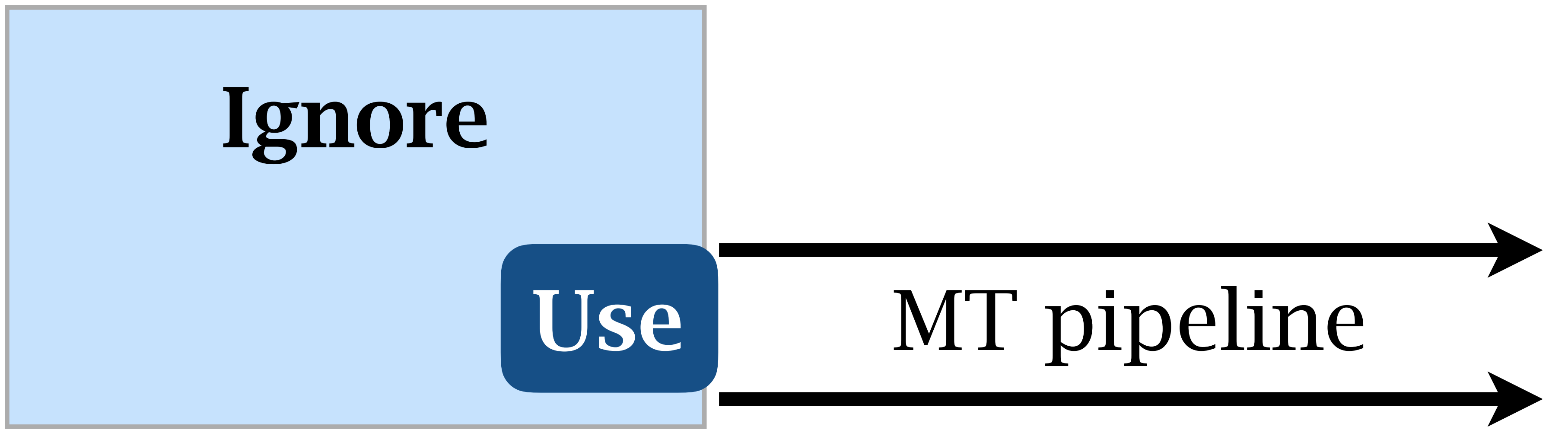
- Training data rarely matches desired tasks.
- Adaptation:
  - Build system on available training data
  - Adjust the system to new task
    [Often: retune parameters on new task's data]
- Drawbacks:
  - Large systems can be expensive.
  - Out-of-domain systems can be very wrong.

Data Selection



- Insight:
  not all training examples are equally valuable.
-
-

Data Selection



- Insight:
  not all training examples are equally valuable.
- Use your regular pipeline,
  but improve the input!
- "filter Big Data down to Relevant Data"

Data Selection

- For a particular translation task:
    - Identify the most relevant training data.
    - Build a model on only this subset.
- Goal:
    - Better task-specific performance
    - Cheaper (computation, size, time)

Data Selection Algorithm

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select top n%
-
-
-

Data Selection Algorithm

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select top n%
- Use n% to build task-specific MT system
- Combine with system trained on task data (optional)
- Apply task-specific system to task.

Cross-Entropy Difference

- Cross-entropy H relates to perplexity by: $ppl = 2^{H}$
- Score and rank by cross-entropy difference:

$$\underset{s \in POOL}{\operatorname{argmin}} \; H_{LM_{IN}}(s) - H_{LM_{POOL}}(s)$$

(Also called "XEDiff" or "Moore-Lewis")

Cross-Entropy Difference

- Cross-entropy H relates to perplexity by: $ppl = 2^{H}$
- Score and rank by cross-entropy difference:

$$\underset{s\,\in POOL}{\operatorname{argmin}} \; H_{LM_{IN}}(s) - H_{LM_{POOL}}(s)$$

(Also called "XEDiff" or "Moore-Lewis")

- Prefers sentences that both:
  - Are <u>like</u> the target task
  - Are <u>unlike</u> the pool average.

Data Selection Performance

- Training on only the most relevant subset of training data (1%-20%) yields translation systems that are smaller, cheaper, faster, and (often) better.

Cross-Entropy Difference, Again

- Score from Moore & Lewis (2010):

$$\underset{s \in Pool}{\operatorname{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

- Why does this work?
- "If score < 0, then *s* is <u>like</u> Task and <u>unlike</u> Pool."

Cross-Entropy Difference, Again

- Score from Moore & Lewis (2010):

$$\underset{s \,\in\, Pool}{\operatorname{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

- Why does this work?
- "If score < 0, then *s* is <u>like</u> Task and <u>unlike</u> Pool."

  --> this causality is backwards

Cross-Entropy Difference Trick!

- Moore-Lewis only works because we have outside information

$$\underset{s \in Pool}{\operatorname{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

- If : Task and Pool corpora differ significantly
- Then : Task and Pool models disagree on what is "good"
-
-

Cross-Entropy Difference Trick!

- Moore-Lewis only works because we have outside information

$$\underset{s \in Pool}{\operatorname{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

- If : Task and Pool corpora differ significantly
- Then : Task and Pool models disagree on what is "good"
- Else : No adaptation needed = wrong scenario!
- Trick : Assume the disagreement exists! Then exploit it.

Cross-Entropy Difference Trick!

- Moore-Lewis only works because we have outside information

$$\underset{s \in Pool}{\text{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

not always!!!!

- If : Task and Pool corpora differ significantly
- Then : Task and Pool models disagree on what is "good"
- Else : No adaptation needed = wrong scenario!
- Trick : Assume the disagreement exists! Then exploit it.

The Only Math in this Talk

- From definition of cross-entropy difference:

$$\text{score}(s) = H_{LM_{Task}} - H_{LM_{Pool}}$$

The Only Math in this Talk

- From definition of cross-entropy difference:

$$\begin{aligned}
\text{score}(s) &= H_{LM_{Task}} - H_{LM_{Pool}} \\
&= -\frac{1}{N}\sum_{w\in s}\log LM_{Task}(w) - -\frac{1}{N}\sum_{w\in s}\log LM_{Pool}(w) \\
&= -\frac{1}{N}\sum_{w\in s}[\log LM_{Task}(w) - \log LM_{Pool}(w)]
\end{aligned}$$

The Only Math in this Talk

- From definition of cross-entropy difference:

$$\begin{aligned}
\text{score}(s) &= H_{LM_{Task}} - H_{LM_{Pool}} \\
&= -\frac{1}{N}\sum_{w \in s} \log LM_{Task}(w) - -\frac{1}{N}\sum_{w \in s} \log LM_{Pool}(w) \\
&= -\frac{1}{N}\sum_{w \in s} [\log LM_{Task}(w) - \log LM_{Pool}(w)] \\
&\propto \sum_{w \in s} \log \frac{LM_{Task}(w)}{LM_{Pool}(w)}
\end{aligned}$$

$$\boxed{\text{score}(s) \propto \sum_{w \in s} \log \frac{P_{Task}(w)}{P_{Pool}(w)}}$$

unigram frequency ratio

Implication

- If : Task and Pool frequencies of a word w_1 differ
- Then : Task and Pool disagree whether w_1 is "good"

  ==> w_1 indicates either Task xor Pool -- not both!

-

Implication

- If : Task and Pool frequencies of a word w_1 differ
- Then : Task and Pool disagree whether w_1 is "good"

==> w_1 indicates either Task xor Pool -- not both!

- Else : Task and Pool frequencies of w_2 are similar

==> Corpora agree whether w_2 is "good" (or not!)
==> freq. ratio = 1, so log (freq. ratio) = 0
==> w_2 does not affect difference score

New Trick

$$\text{score}\ (w) \quad \propto \quad \log \frac{P_{Task}(w)}{P_{Pool}(w)}$$

- Old trick: Assume the domains and corpora disagree.
- New trick: Assume similarity score is cross-entropy difference.
- Exploit: Differentiate between biased and non-biased words.

New Trick

- Could build a classifier, but don't want to.
- Instead of a discriminative model on a standard representation, train a standard model on a discriminative representation.
- How?
-

New Trick

- Could build a classifier, but don't want to.
- Instead of a discriminative model on a standard representation,
  train a standard model on a discriminative representation.
- How?
  Mark words with suffix indicating how biased they are.
- Now each corpus includes information about the <u>other</u> one.

Choosing Parameters

- Marking bias is not change word statistics;
  does not do anything by itself.

- Collapse all non-contributing words together for the purpose of computing similarity:

  Replace words with POS tags (motivated in paper)

- Questions:
  - How to decide the threshold for replacing?
  - How granular should the bias suffix be?

Choosing Parameters

- Construct pathological case:
  - Ignore threshold. Just replace all words!
  - Bucket frequency ratio bias by powers of 10.
- "supermassive black holes"

  ==> JJ/+++ JJ/0 NNS/+
- These are not ideal settings!

Choosing Parameters

- Construct pathological case:
  - Ignore threshold. Just replace all words!
  - Bucket frequency ratio bias by powers of 10.
- "supermassive black holes"
  ==> JJ/+++ JJ/0 NNS/+
- These are not ideal settings!

  But if <u>this</u> works, <u>anything</u> should work.

Relevant Dimensions, not More

- In modern terms:
  projection into a discrete 2-D embedding:

freq. ratio

	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}	10^{3}	low
JJ	JJ/---	JJ/--	JJ/-	JJ/0	JJ/+	JJ/++	JJ/+++	JJ/low
NNS	NNS/---	NNS/--	NNS/-	NNS/0	NNS/+	NNS/++	NNS/+++	NNS/low
VBZ								
...								

POS tags

< .0001x

0.0001 to .001x

0.001 to .01x

0.1 to 10x

10 to 100x

100 to 1000x

1000x <

count < 10

Procedure

1. POS tag the corpora <-- could try cluster labels
2. Compute vocab statistics and ratios <-- unigram LM
3. Transform text <-- perl script
4. Do cross-entropy difference <-- reuse code
5. Put words back in <-- perl script

Transformed Text

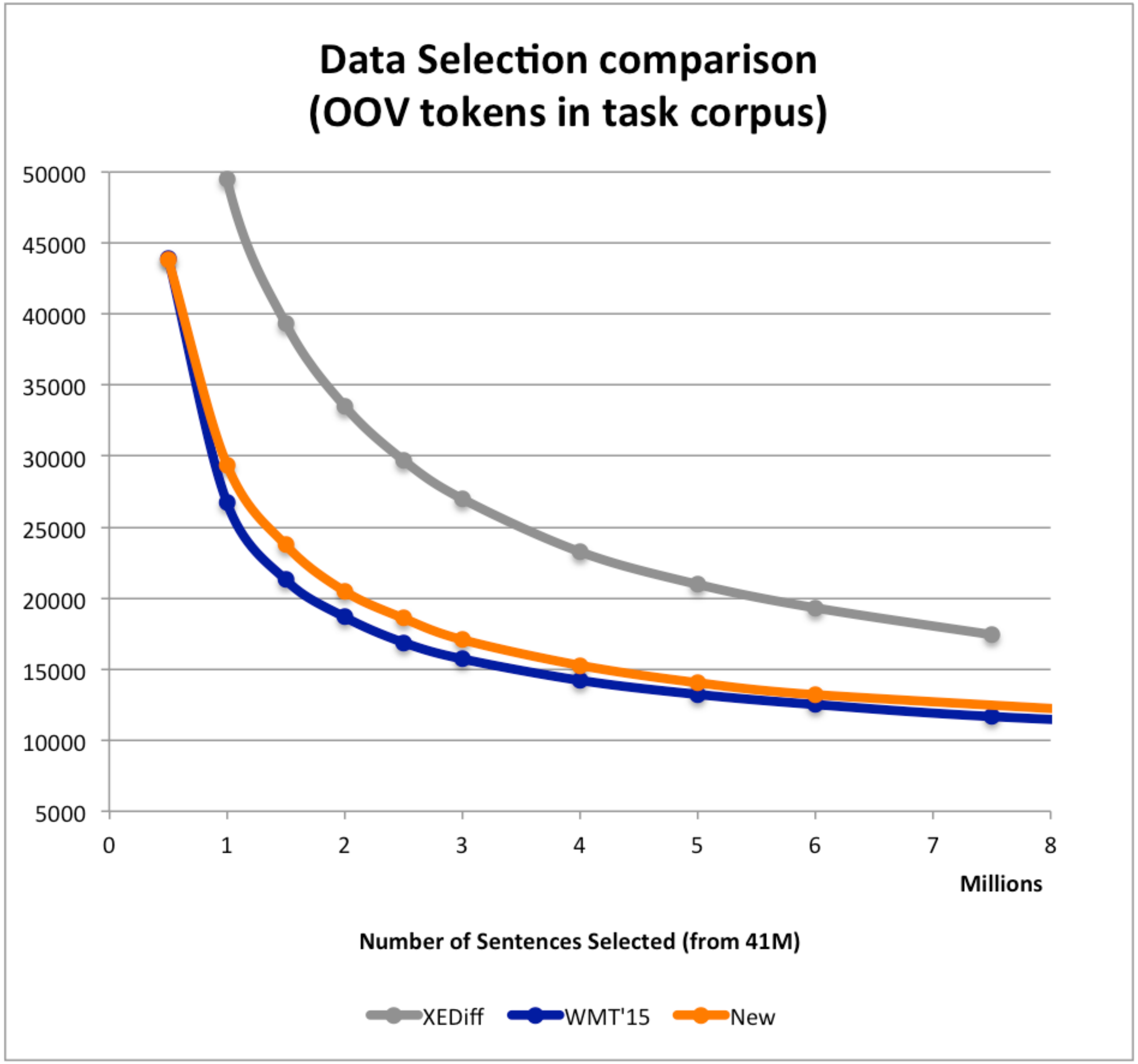
- Task corpus:
  - **PRP/+ VBP/++ VBG/+** TO/0 **VB/+ PRP/+** VB/0 NN/0 ./0
  - PRP/0 **VBP/++ VBG/+** TO/0 VB/0 NN/0 ./0
- Pool corpus:
  - MD/0 **CD/- NN/-- NN/low** CD/0 VBZ/0 DT/0 **JJ/low NNS/low** IN/0 DT/0 VBN/0 **NN/low** ./0
  - MD/0 CD/0 **NN/--** NNP/0 NNP/0 **NNP/low** VBZ/0 DT/0 JJ/0 **RB/low** VBN/0 NN/0 IN/0 **NNP/low** IN/0 RB/0 CD/ 0 NNS/0 ./0

Experimental Setup

- French --> English translation
- Task: TED, 207 k lines (4.2m tokens)
  Pool: LDC, 41 M lines (1.2b tokens)
- Vocab (En): 3.9m
  Vocab (Fr): 3.6m
- Penn POS tagset (43 tags)
- 344 possible tag/bias labels, only use < 200

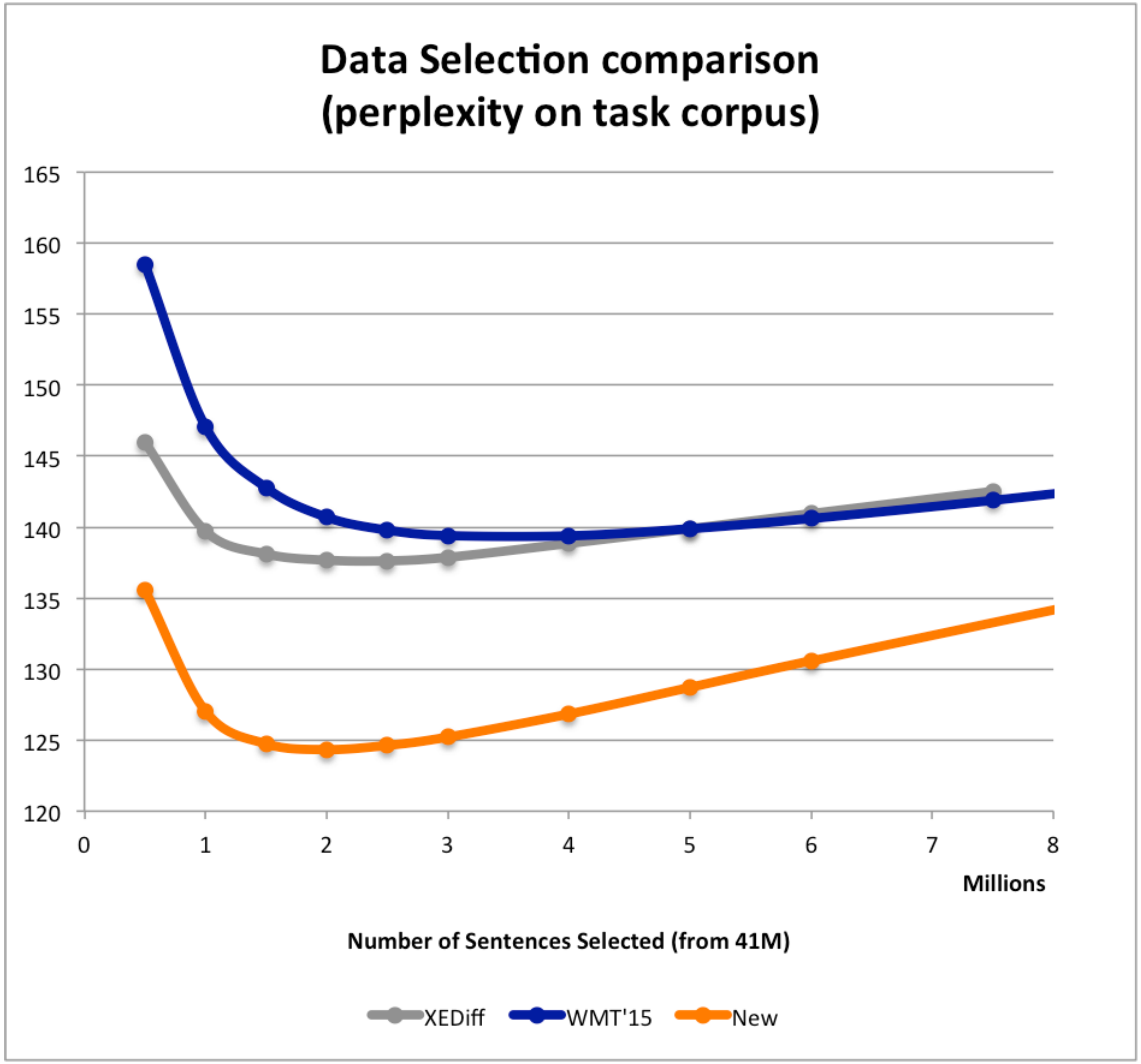
In-Domain Lexical Coverage

- Using only selected data (orange)
- -35% OOV compared to Moore-Lewis (grey)



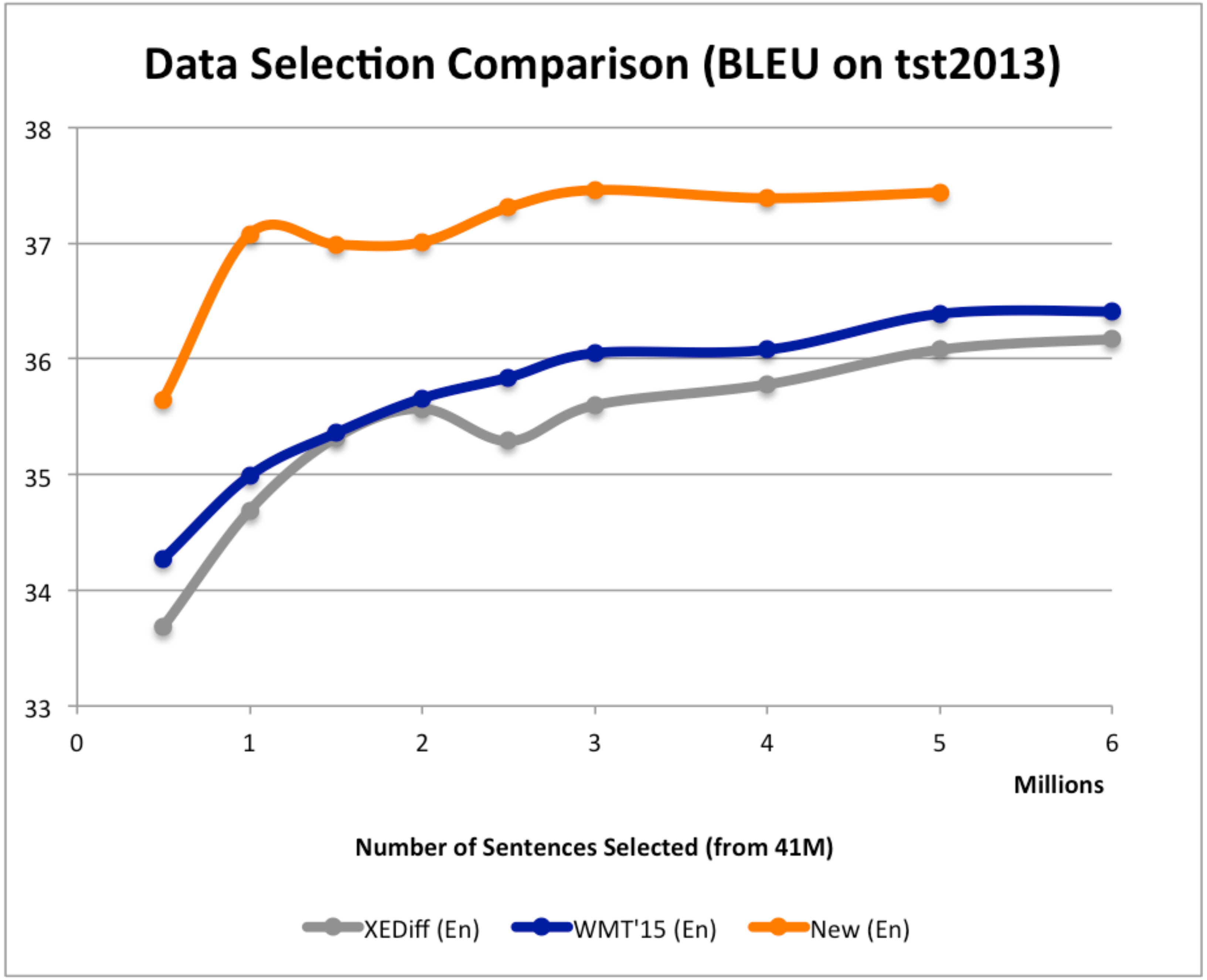
In-Domain Perplexity

- Using only selected data (orange)
- -10% ppl compared to Moore-Lewis (grey)
- Despite using zero words!



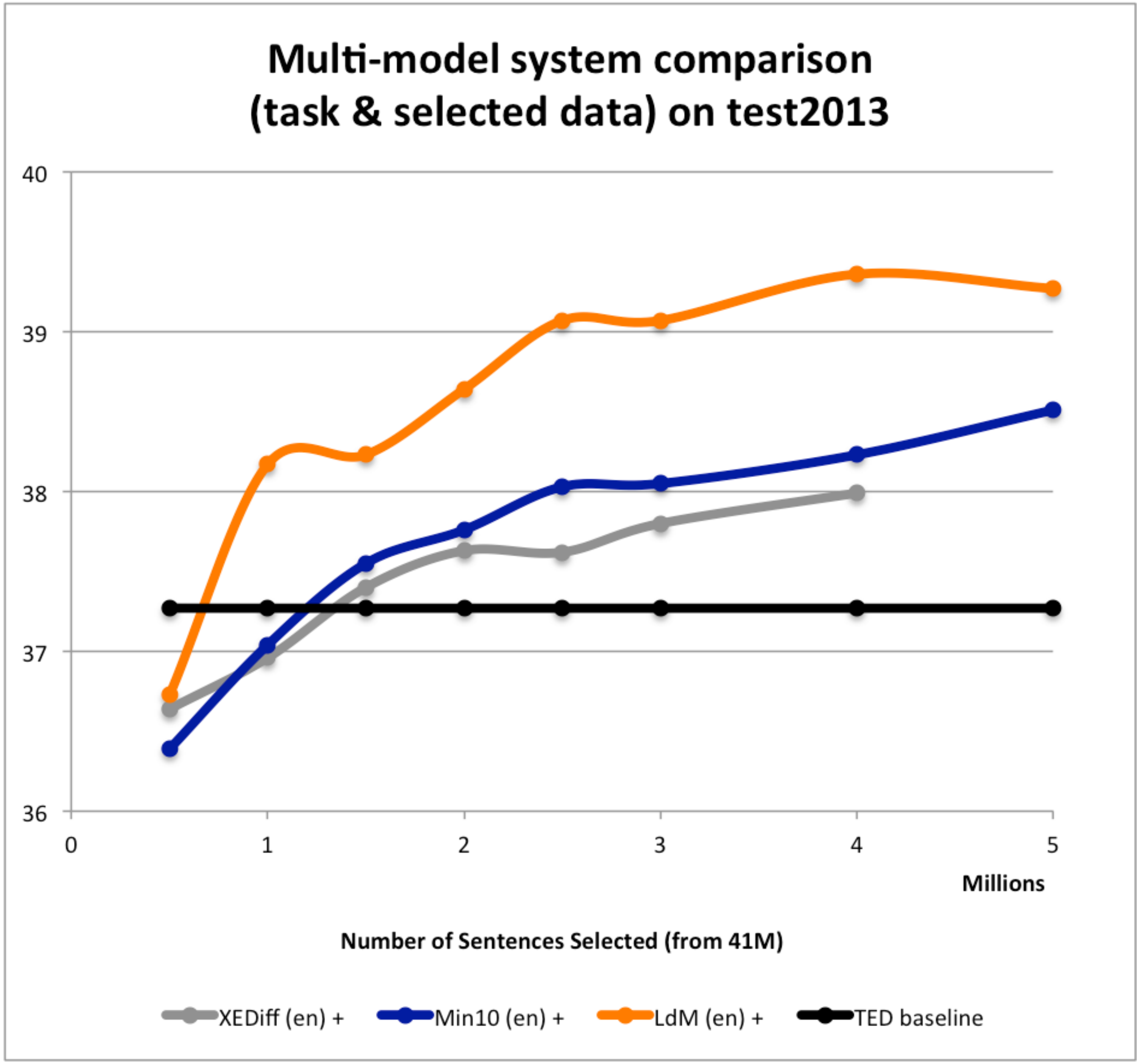
TED Fr-En Translation

- Using only selected data (orange)
- +1.85 BLEU compared to Moore-Lewis (grey)



Multi-Model System (2TM,2LM)

- TED + new selected data (orange)
- +1.3 BLEU compared to TED + Moore-Lewis (grey)



Summary

- Model each corpus relative to the other one.
- How: replace all words with POS tags,
  add discriminative suffixes,
  run regular Moore-Lewis data selection.
- Improves perplexity (-10%), BLEU (+1.5), OOV (-30%)
- Almost 100% reduction in active lexicon,
  to < 200 types with robust statistics.
- 99% reduction in LM size for selection process.

Thank You

- Questions?

- Sample implementation appearing soon:

  https://github.com/amittai/

- Contact: amittai@clsp.jhu.edu

[this slide intentionally left blank]

Domain Adaptation

- Training data doesn't always match desired tasks.
- Have bilingual:
    - Parliament proceedings
    - Newspaper articles
    - Web scrapings
- Want to translate:
    - Travel scenarios
    - Facebook updates
    - Realtime conversations
- Sometimes want a specific kind of language, not just breadth!

Perplexity-Based Filtering

- A language model LM_Q measures the likelihood of some text by its perplexity:

$$ppl_{LM_Q}(s) = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log LM_Q(w_i|h_i)} = 2^{H_{LM_Q}(s)}$$

- Intuition: Average branching factor of LM
- Cross-entropy H (of a text w.r.t. an LM) is *log*(ppl).

Cross-Entropy Difference

- Perplexity-based filtering:
  - Score and sort sentences in pool
    by perplexity with in-domain LM.
  - Then rank, select, etc.
- However!
  The data pool does not match the target task.

Bilingual Cross-Entropy Diff.

- Extend the Moore-Lewis similarity score for use with bilingual data, and apply to SMT:

$$\begin{aligned} &H_{L1}(s_1, LM_{Task}) - H_{L1}(s_1, LM_{Pool}) \\ +&H_{L2}(s_2, LM_{Task}) - H_{L2}(s_2, LM_{Pool}) \end{aligned}$$

- Training on only the most relevant subset of training data (1%-20%) yields translation systems that are smaller, cheaper, faster, and (often) better.

Are All Words Useful?

- How much can we trust rare words?
- If a word is seen 2 times in the general corpus
  and 3 in the in-domain one,
  is it really 50% more likely?
- Volatile difference in LM probabilities.

Using Fewer Words

- Low-frequency words often ignored (*e.g.* Good-Turing smoothing, singleton pruning, ...)

Using Fewer Words

- Low-frequency words often ignored (*e.g.* Good-Turing smoothing, singleton pruning, ...)
- ==> All words contribute, but not equally.
-
-

Using Fewer Words

- Low-frequency words often ignored (*e.g.* Good-Turing smoothing, singleton pruning, ...)
- ==> All words contribute, but not equally.
- However, "ignored" does not mean "deleted". (*e.g.* discounted counts, *<unk>* token, ...)

Using Fewer Words

- Low-frequency words often ignored (*e.g.* Good-Turing smoothing, singleton pruning, ...)
- ==> All words contribute, but not equally.
- However, "ignored" does not mean "deleted". (*e.g.* discounted counts, *<unk>* token, ...)
- ==> Aggregate rare words!

Hybrid word/POS Corpora

- In stylometry,
  syntactic structure = proxy for style.

- POS-tag n-grams used as features to determine authorship, genre, etc.

- Incorporate this idea as a pre-processing step to data selection:

Hybrid word/POS Corpora

- In stylometry,
  syntactic structure = proxy for style.

- POS-tag n-grams used as features to determine authorship, genre, etc.

- Incorporate this idea as a pre-processing step to data selection:

  Replace rare words with POS tags.

Hybrid word/POS Corpora

- Replace rare words with POS tags:
  - an earthquake in Port-au-Prince
  - an earthquake in NNP

Hybrid word/POS Corpora

- Replace rare words with POS tags:
    - an earthquake in Port-au-Prince
    - an NN in NNP
    -
-

Hybrid word/POS Corpora

- Replace rare(?) words with POS tags:
  - an earthquake in Port-au-Prince
  - DT NN IN NNP
  -
-

Hybrid word/POS Corpora

- Replace rare words with POS tags:
  - an earthquake in Port-au-Prince
  - an earthquake in NNP
  - an earthquake in Kodari
-

Hybrid word/POS Corpora

- Replace rare words with POS tags:
  - an earthquake in Port-au-Prince
  - an earthquake in NNP
  - an earthquake in Kodari
- Threshold: (if $Count < 10$) in <u>either</u> corpus

Using Fewer Words

- Use the hybrid word/POS texts instead of the original corpora.
- Train LMs on the corpora, compute sentence scores, and re-rank the original general corpus.
- This is just normal Cross-Entropy Diff scoring, but using a different corpus representation.
- It works!
  Results in Axelrod/Resnik/He/Ostendorf, WMT 2015.

Hybrid Word/POS Selection

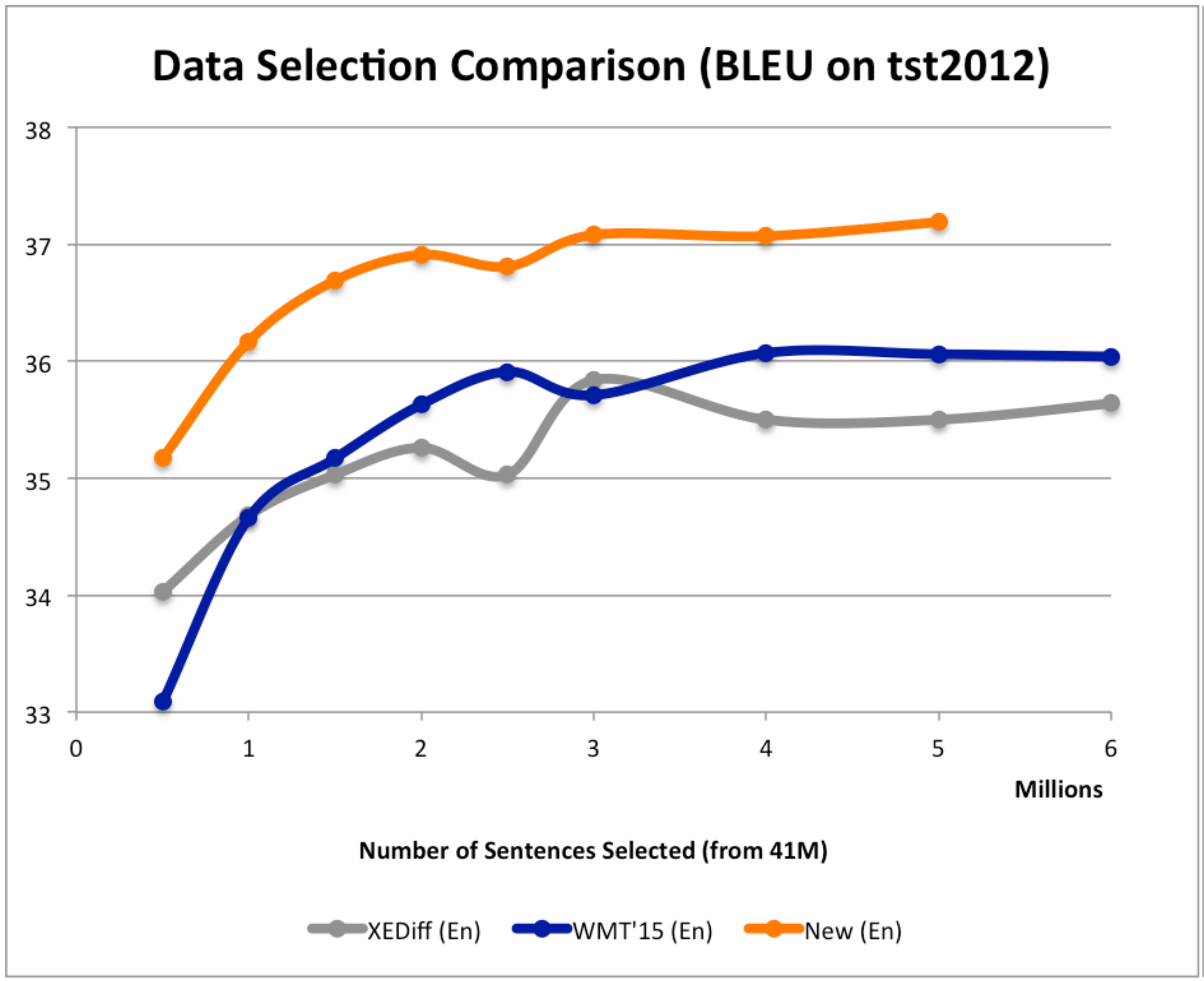
- Must re-compute for every task/pool, but vocabulary statistics are easy.
- Aggregating the statistics for rare terms allows generalizing to other unseen words.
- Perhaps preserving sentence structure, picking up words that fill similar roles/patterns in the sentence?

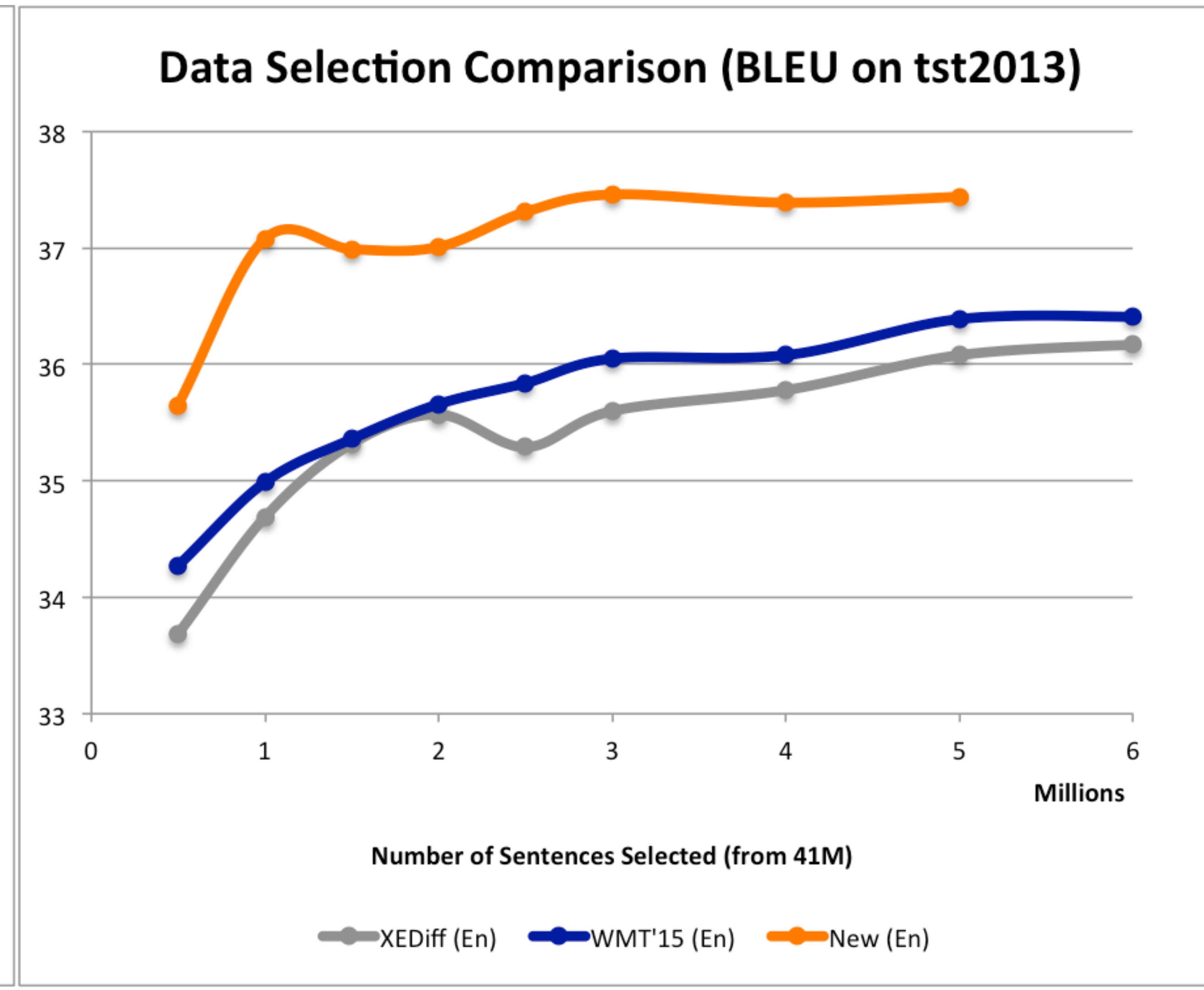
- We used two kinds of models!
  (well, two kinds of representations)

- "Modeling" != "Modeling similarity":

  "Characterizing a corpus" <-- fluency
  vs.
  "Matching a corpus" <-- relevance

TED Fr-En Translation





- +1.8 BLEU