89688: Statistical Machine Translation

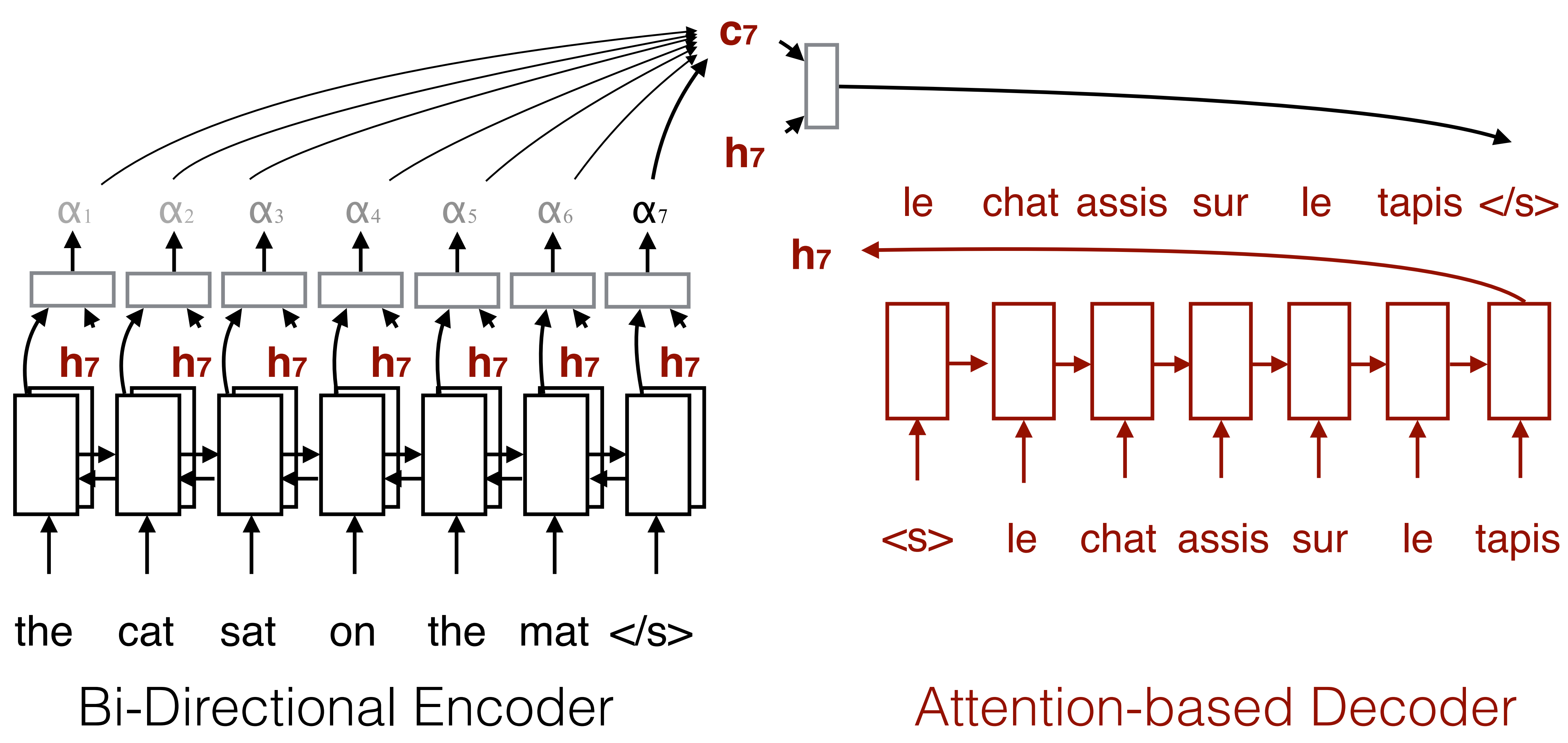
NMT: Making it Work

June 2020

Roee Aharoni
Computer Science Department
Bar Ilan University

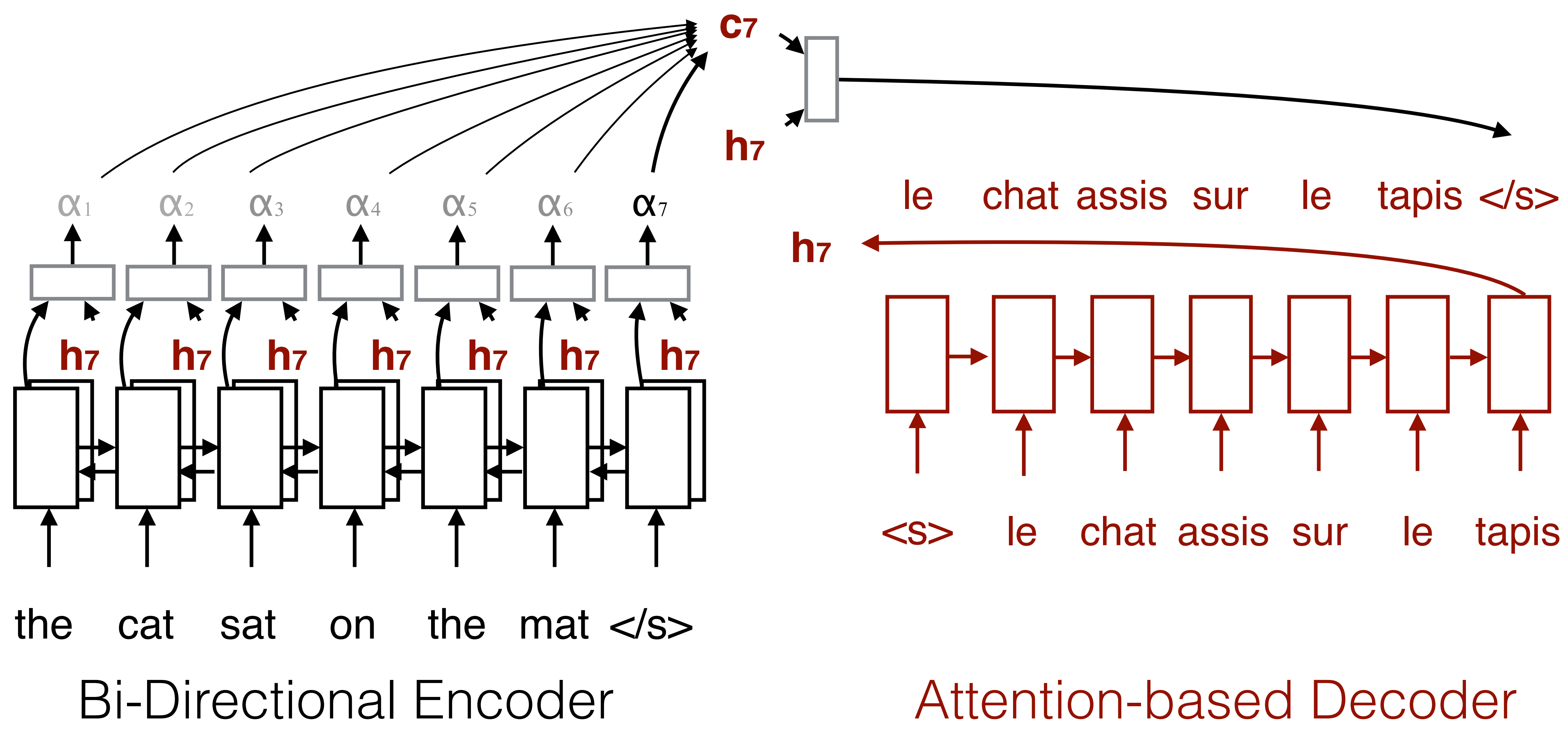
Based in part on slides by Rico Sennrich

NMT is all you need?



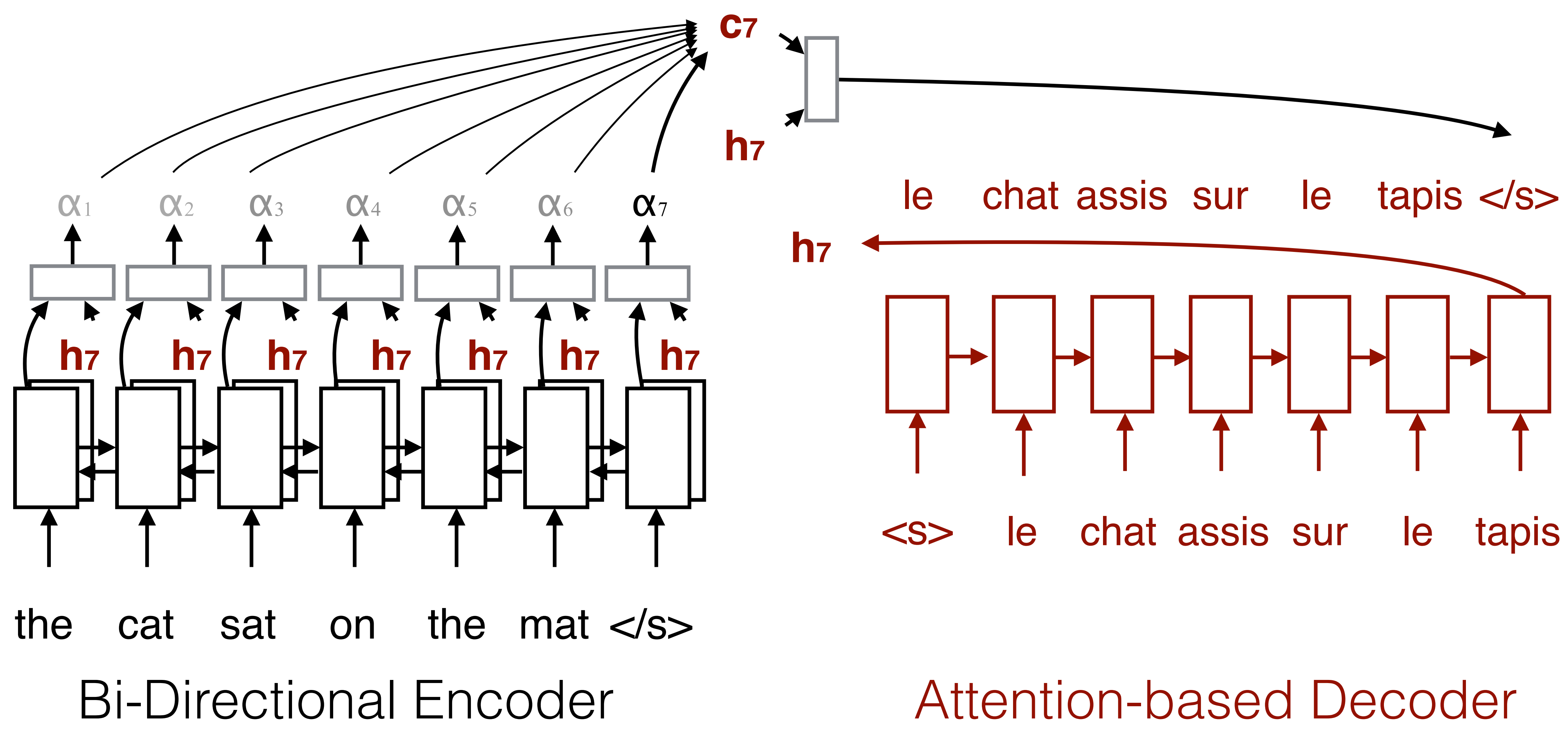
NMT is all you need?

- Neural machine translation (NMT) has strong advantages:



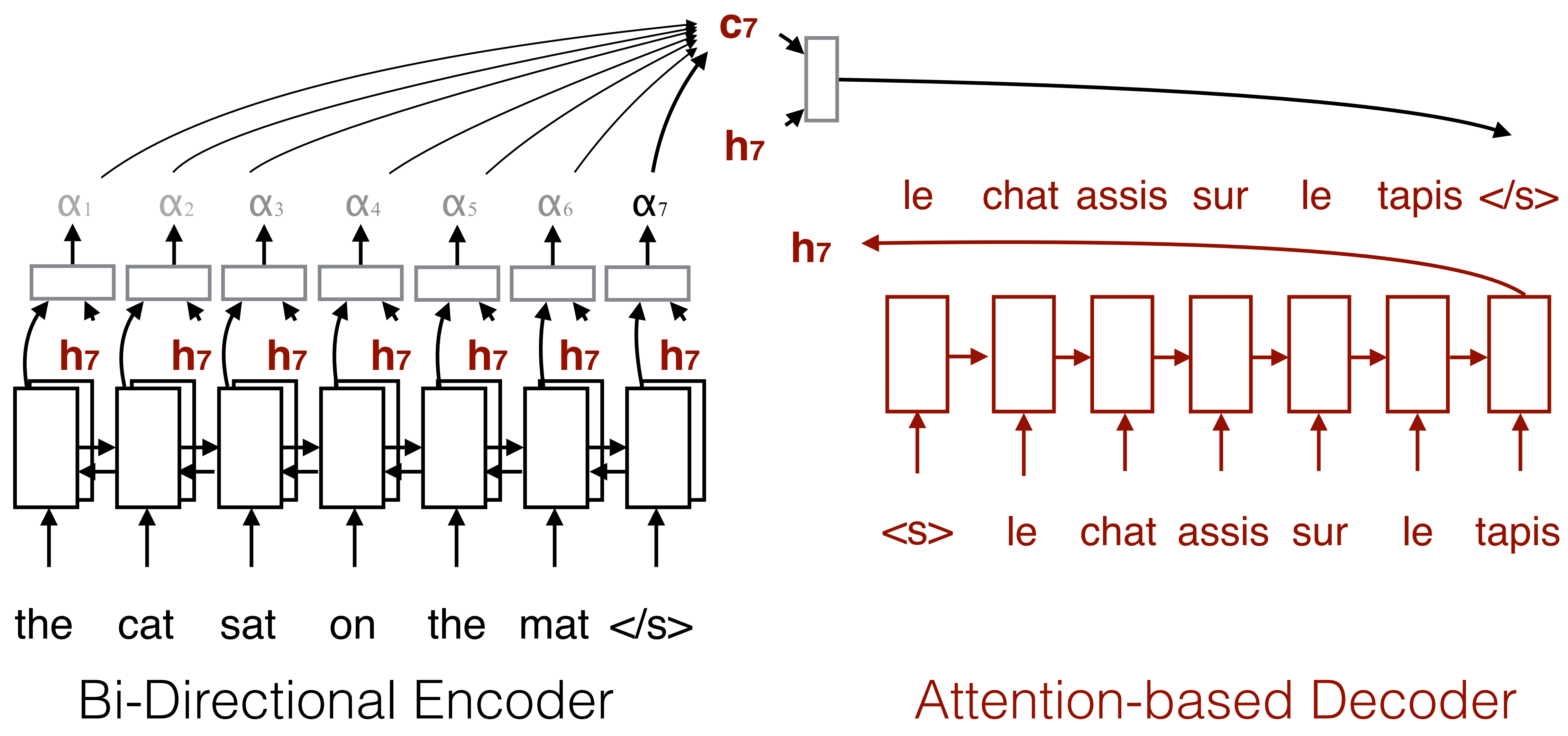
NMT is all you need?

- Neural machine translation (NMT) has strong advantages:
  - **Simple** to train - “end-to-end"



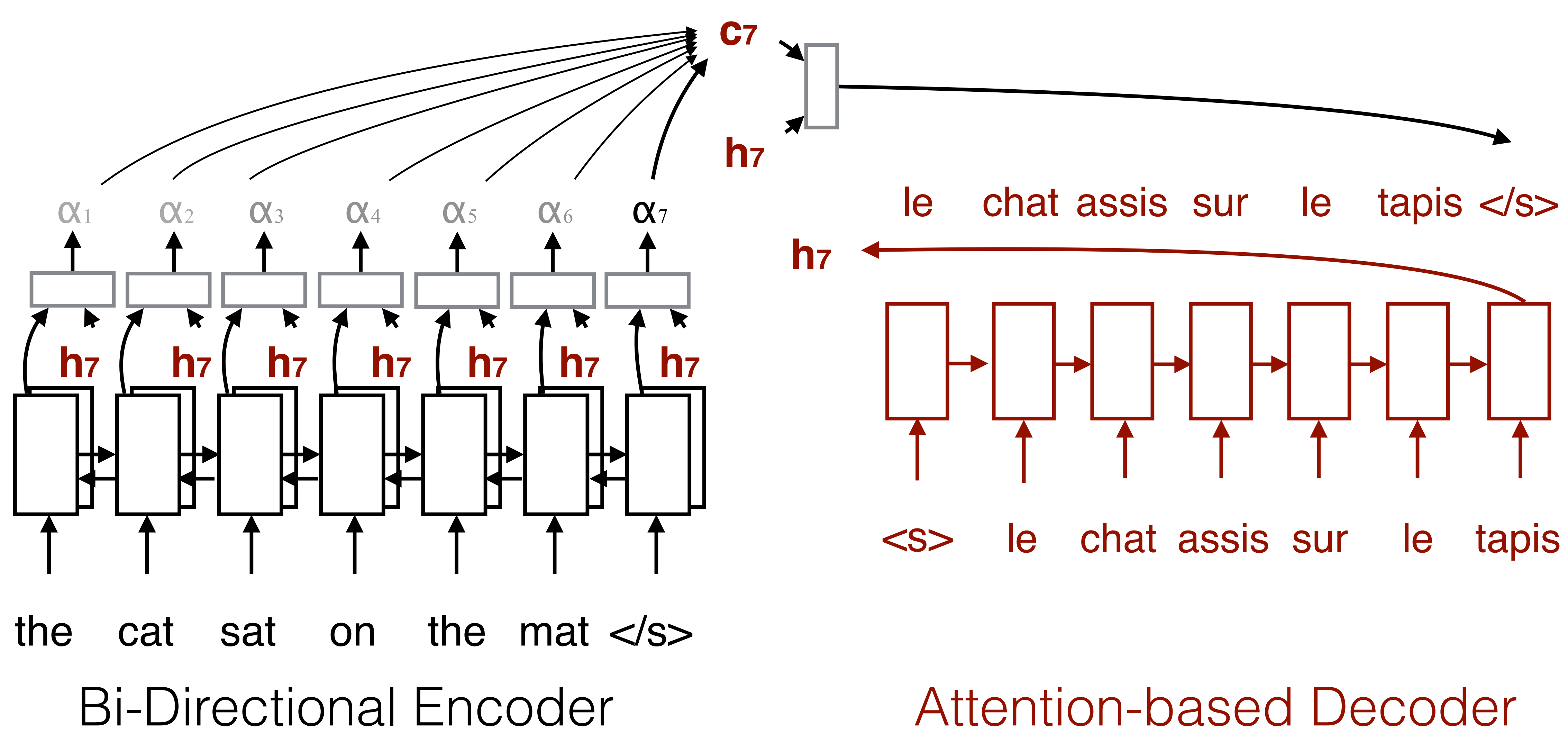
NMT is all you need?

- Neural machine translation (NMT) has strong advantages:
    - **Simple** to train - “end-to-end"
    - Fully **context-aware**



NMT is all you need?

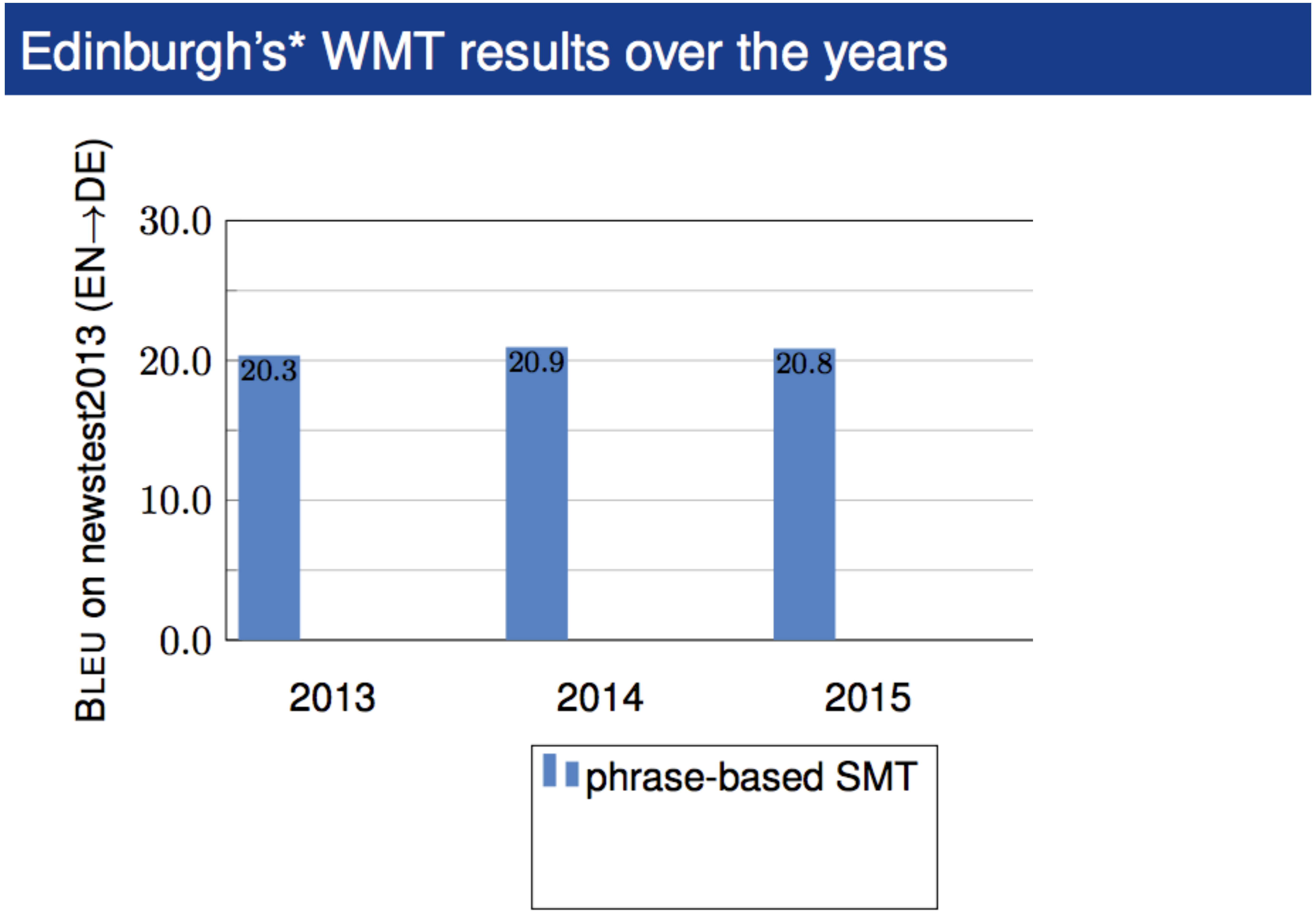
- Neural machine translation (NMT) has strong advantages:
  - **Simple** to train - “end-to-end"
  - Fully **context-aware**
- But how does it perform?



Back to WMT

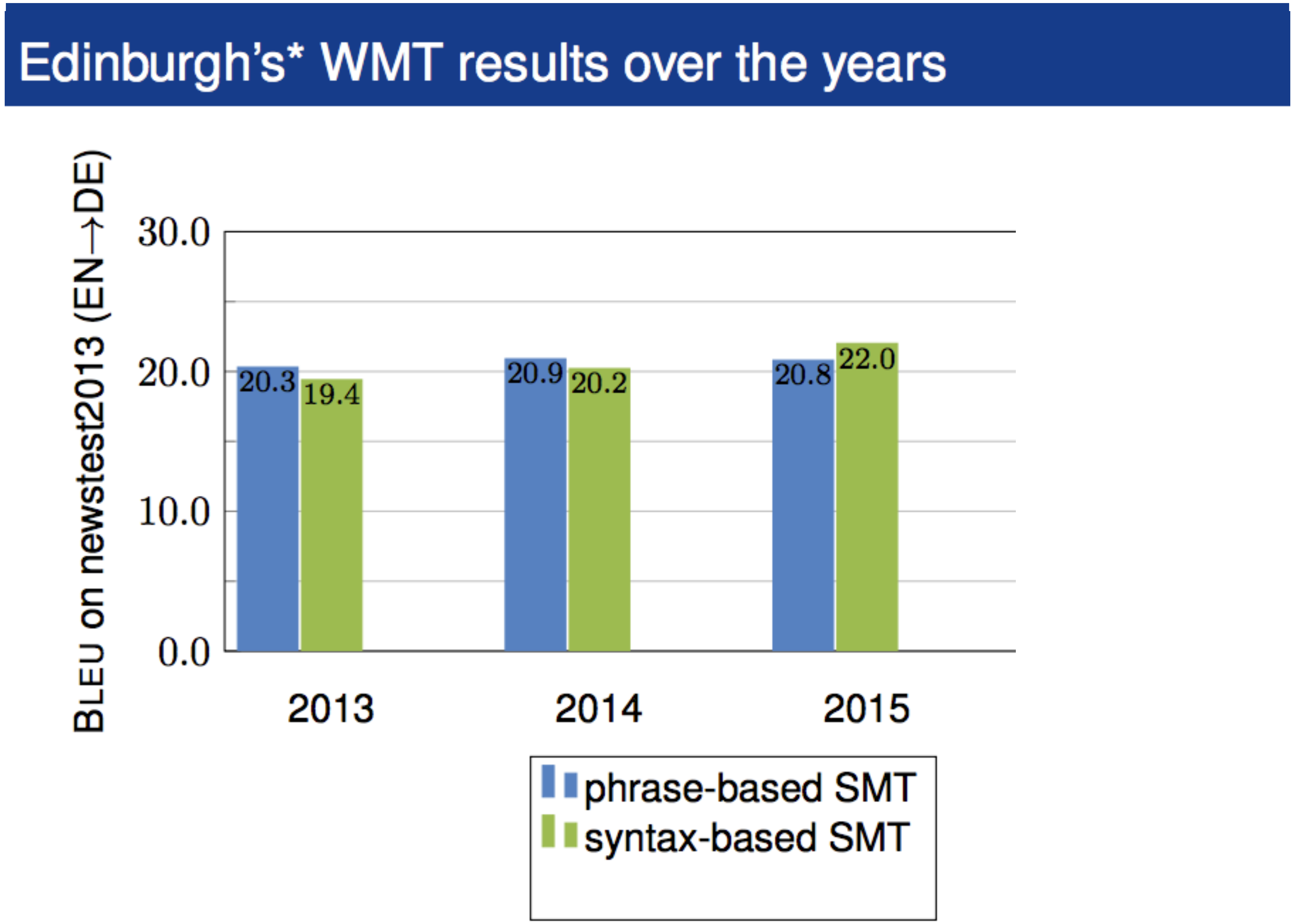
Back to WMT

- Main benchmark for MT



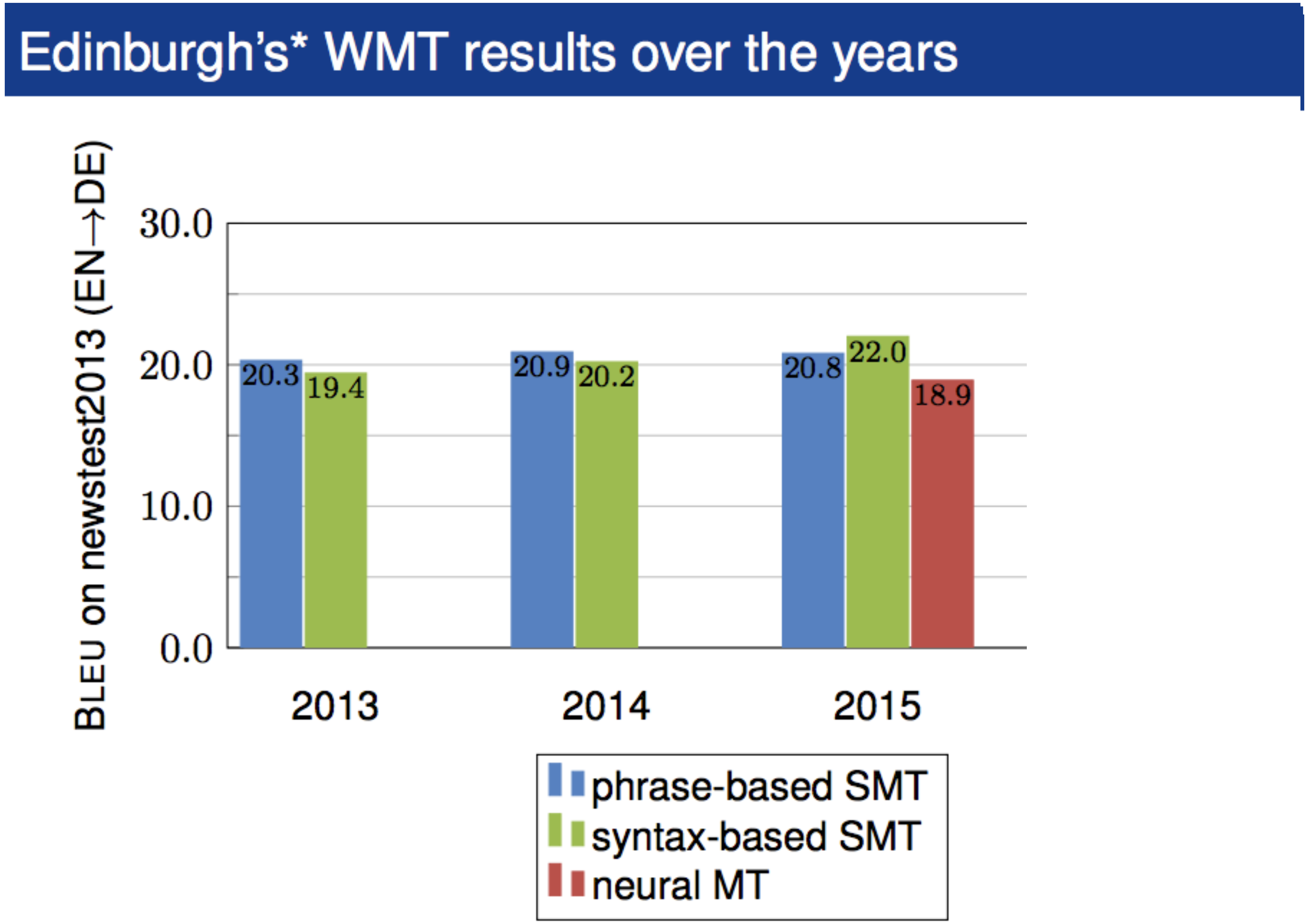
Back to WMT

- Main benchmark for MT
- 2015 - first time a syntax-based system "wins"



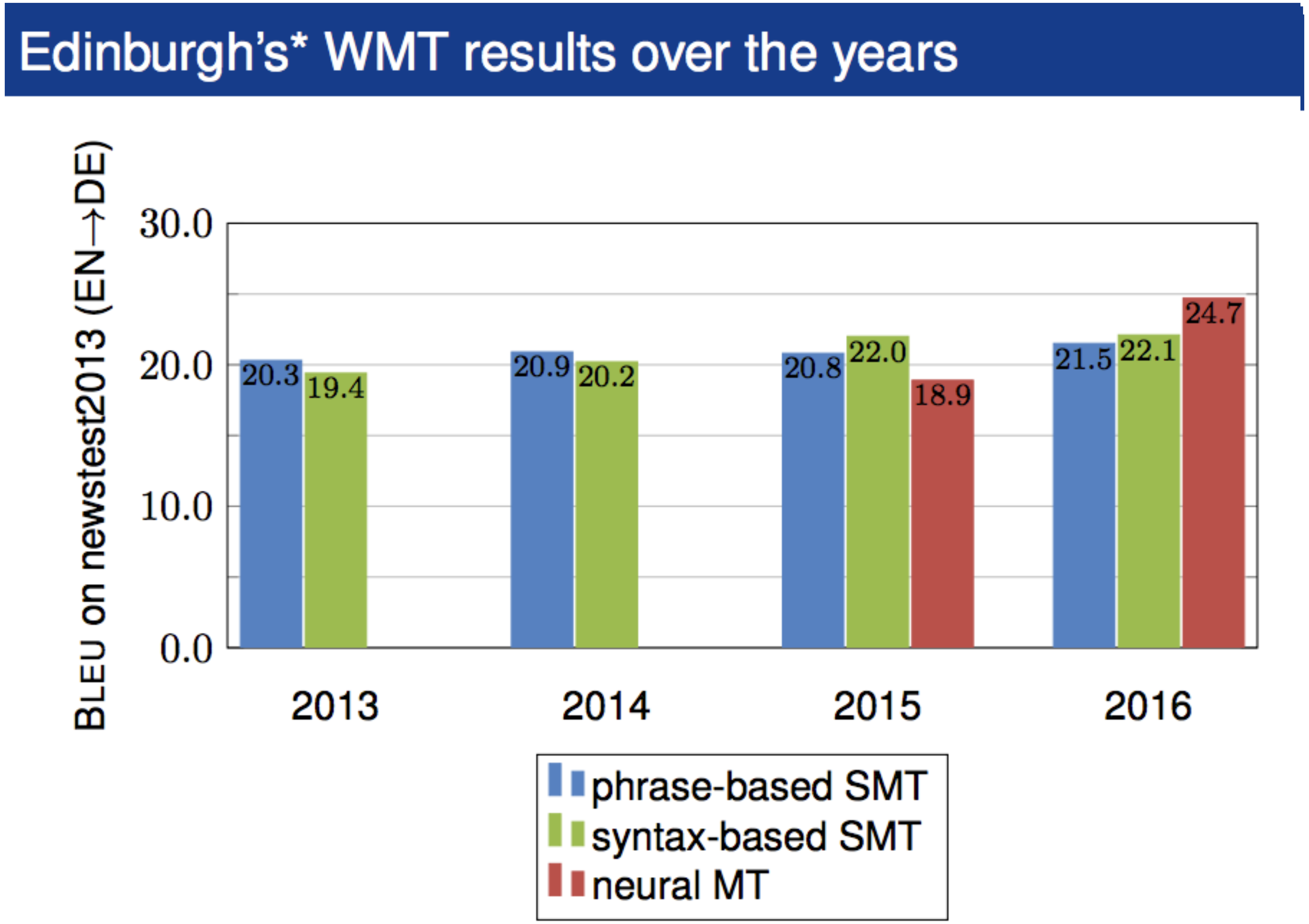
Back to WMT

- Main benchmark for MT
- 2015 - first time a syntax-based system “wins”
- Also 2015 - First time an NMT system (MILA) competes

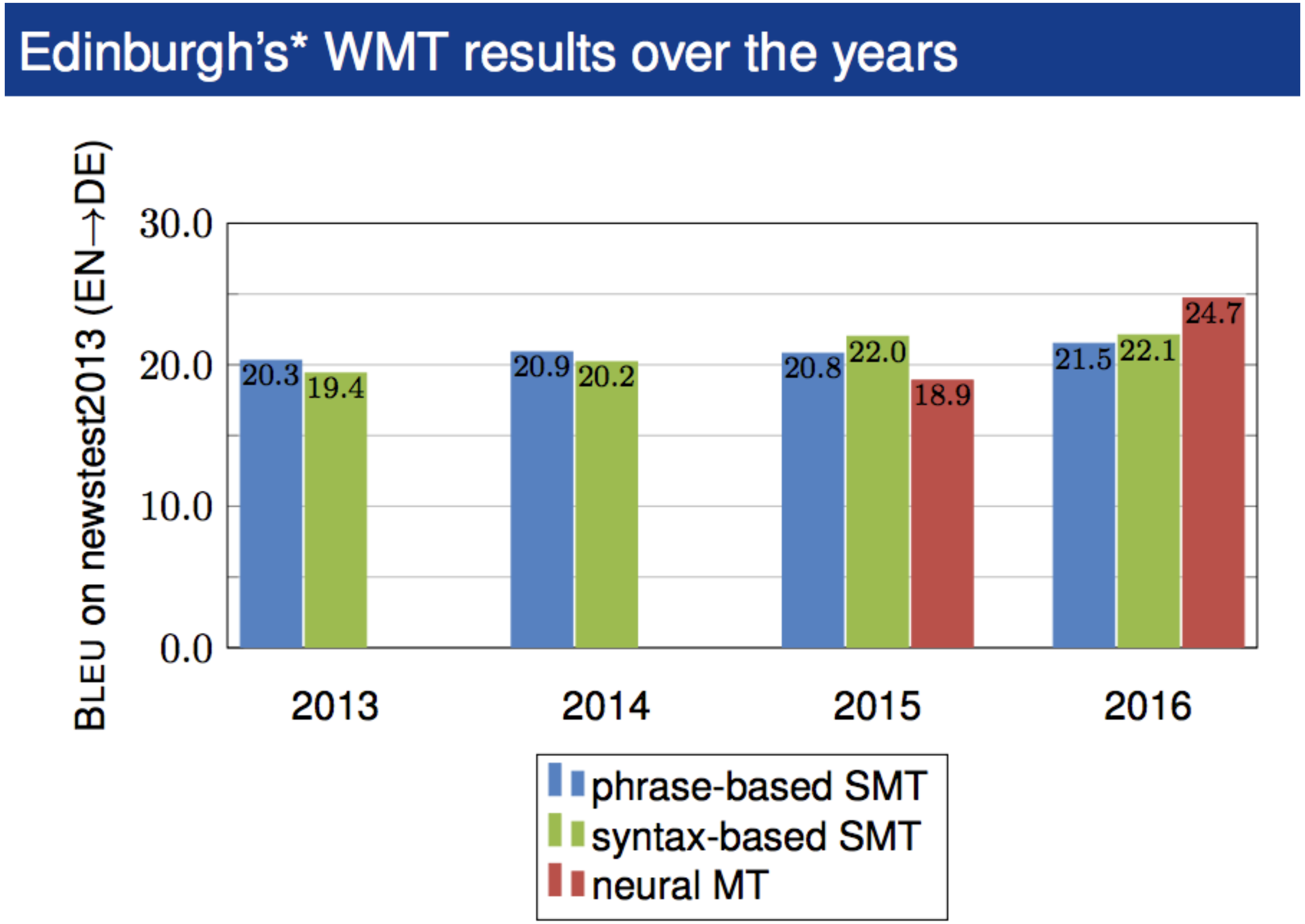


Back to WMT

- Main benchmark for MT
- 2015 - first time a syntax-based system “wins”
- Also 2015 - First time an NMT system (MILA) competes
- 2016 - NMT system wins! (Edinburgh)

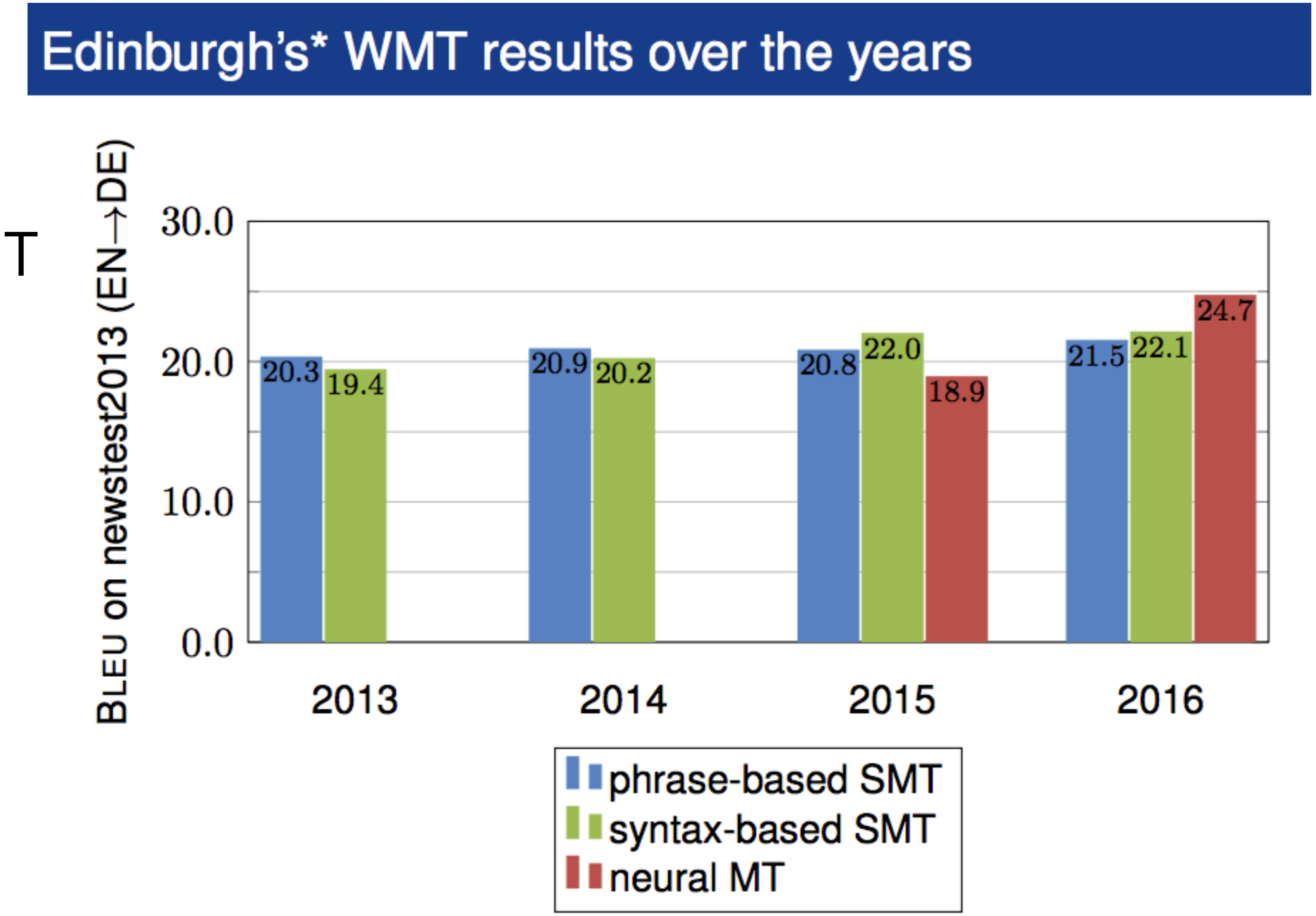


What made NMT win?



What made NMT win?

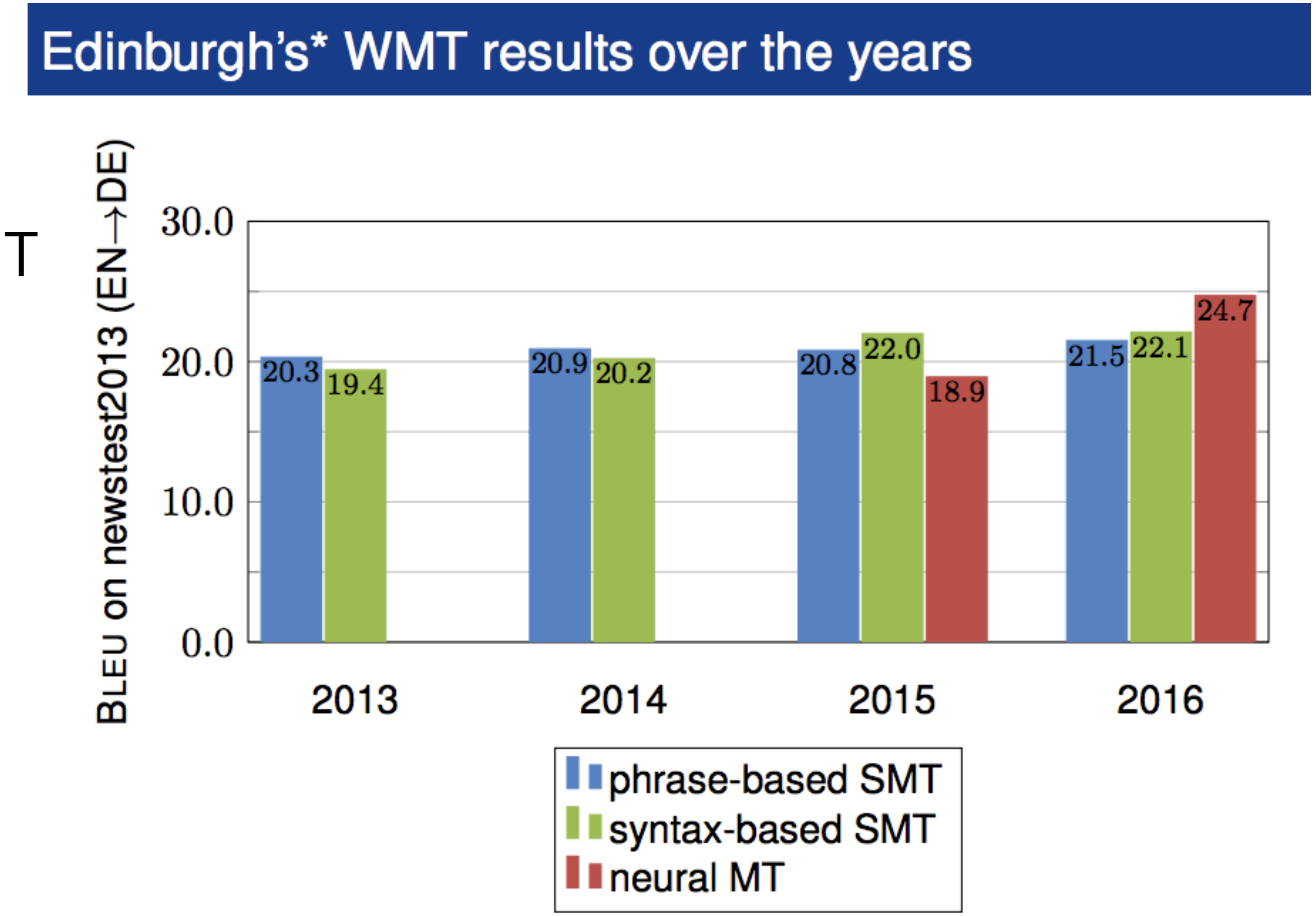
- Several important methods were introduced in 2015-2016 to make NMT outperform PBMT



What made NMT win?

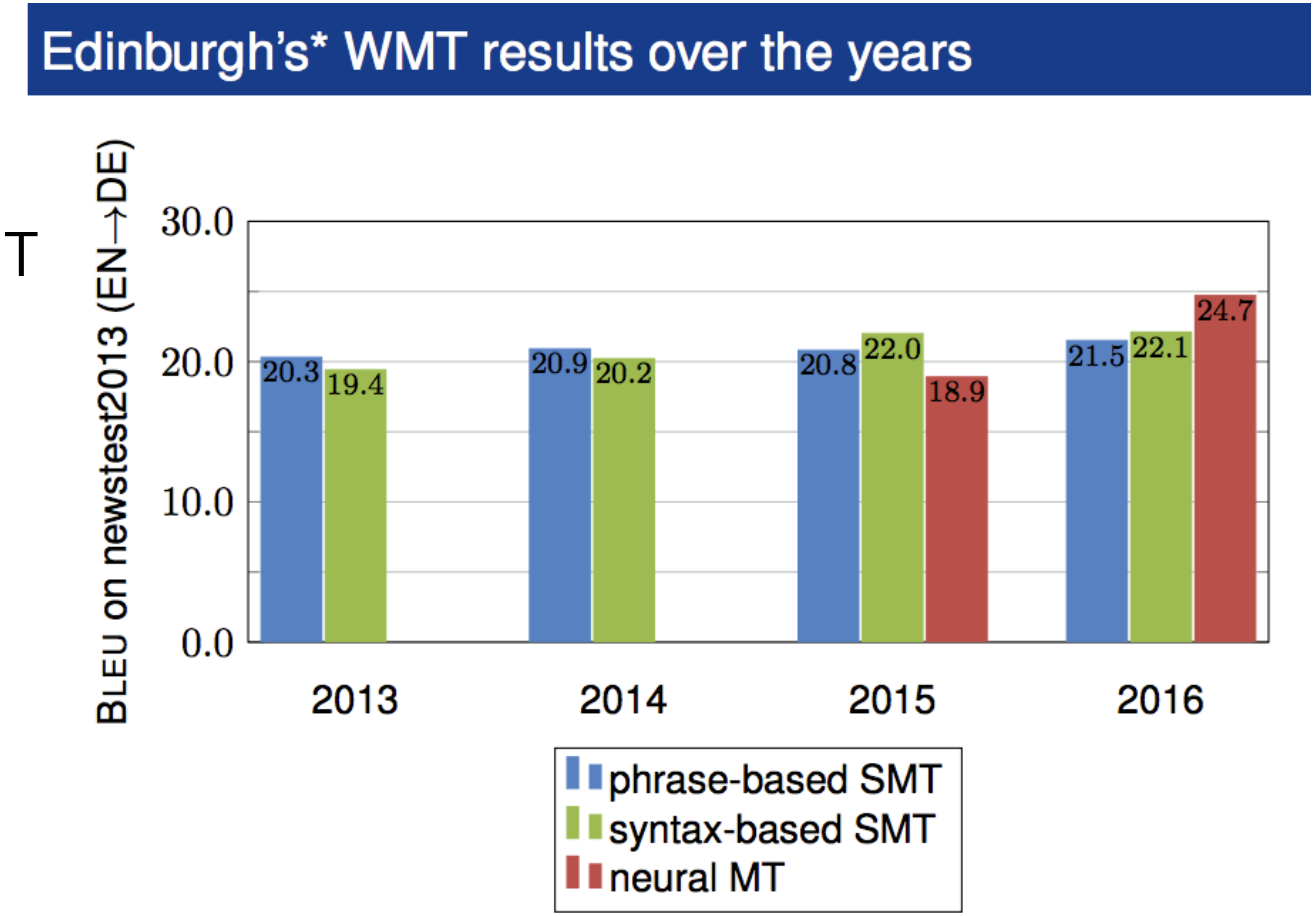
- Several important methods were introduced in 2015-2016 to make NMT outperform PBMT

- Main issues to address:



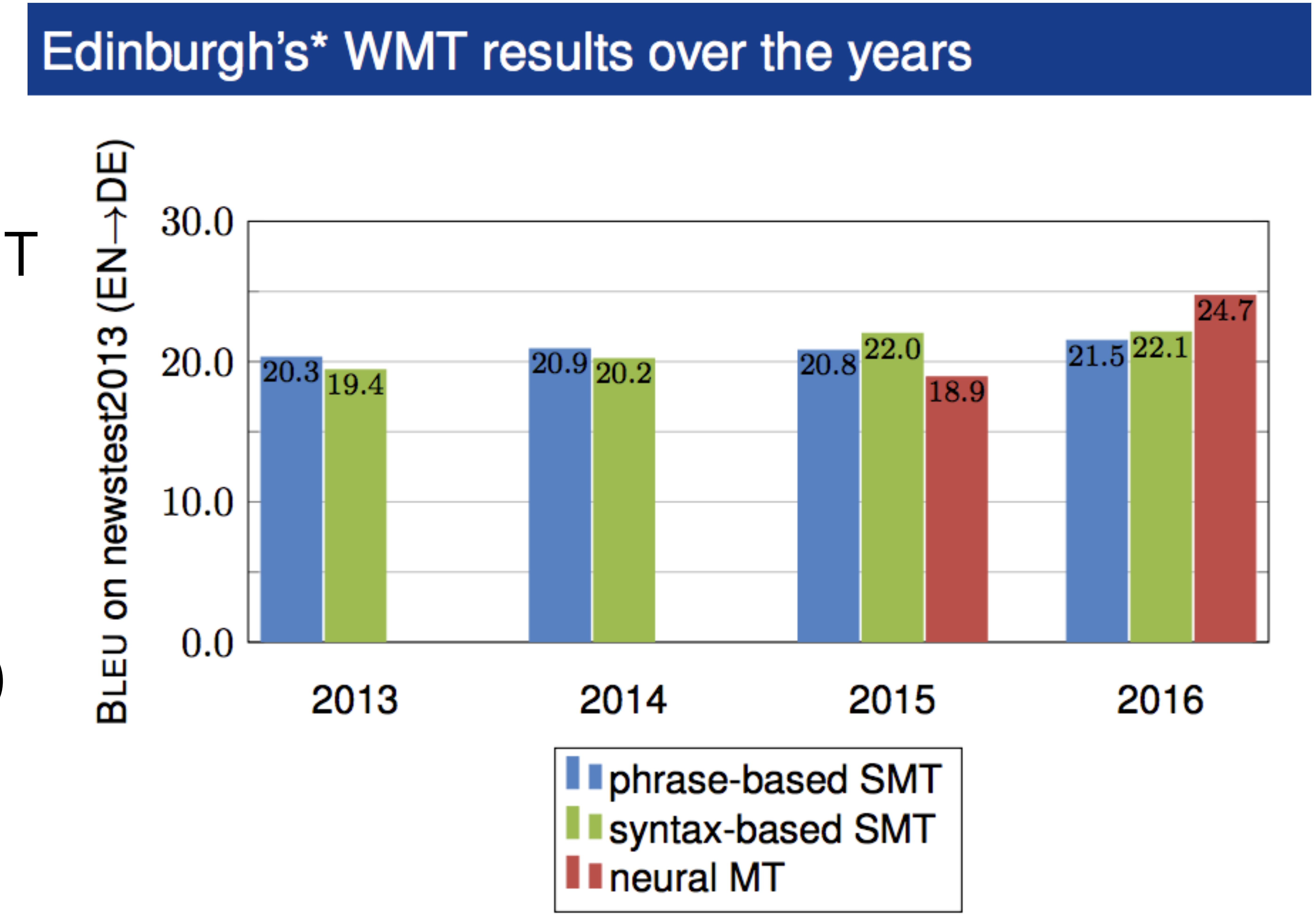
What made NMT win?

- Several important methods were introduced in 2015-2016 to make NMT outperform PBMT
- Main issues to address:
    - Handling **large vocabularies**



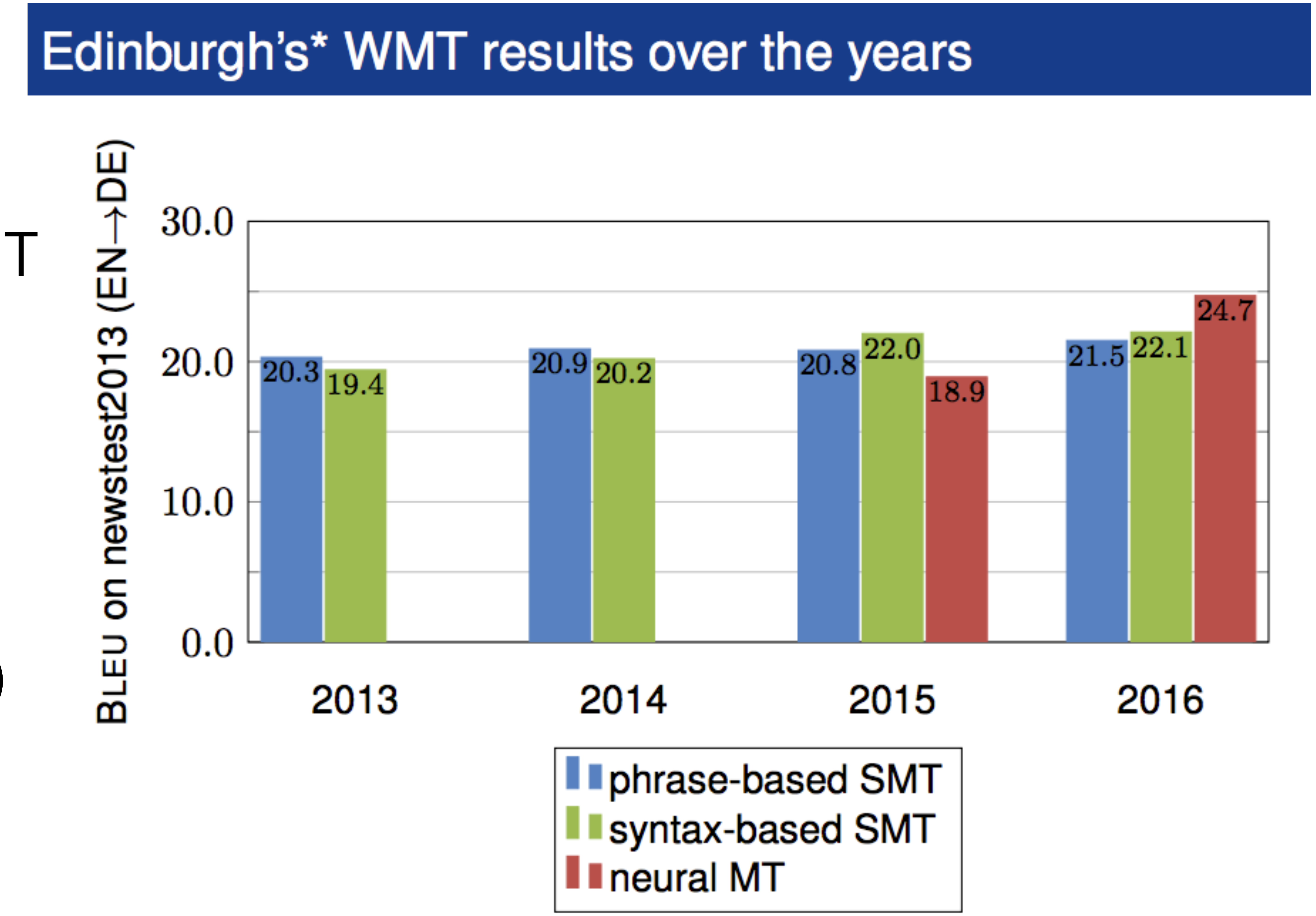
What made NMT win?

- Several important methods were introduced in 2015-2016 to make NMT outperform PBMT
- Main issues to address:
  - Handling **large vocabularies**
  - Using **unlabelled data** ("LM")



What made NMT win?

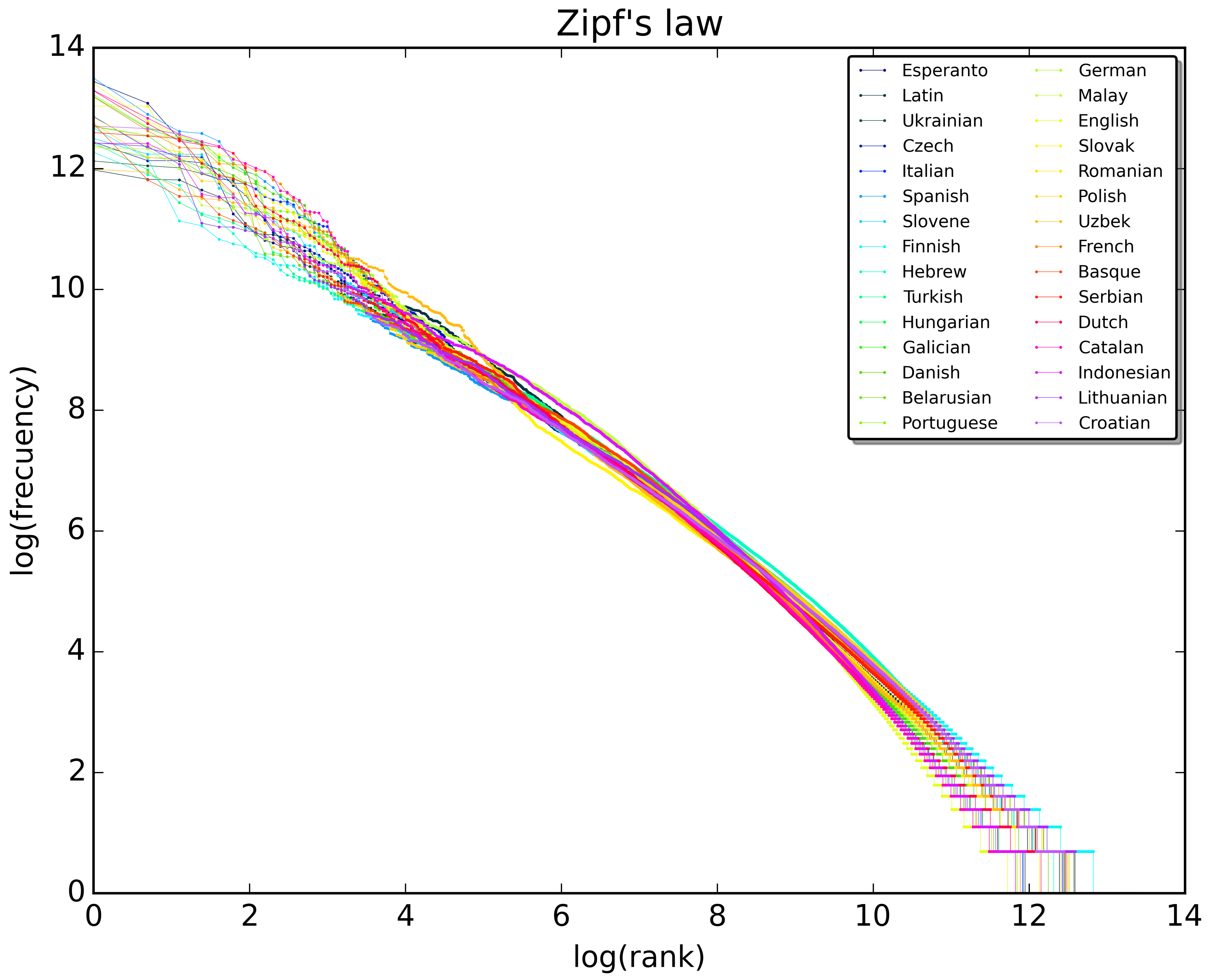
- Several important methods were introduced in 2015-2016 to make NMT outperform PBMT
- Main issues to address:
  - Handling **large vocabularies**
  - Using **unlabelled data** ("LM")
- We will discuss both



Handling Large Vocabularies

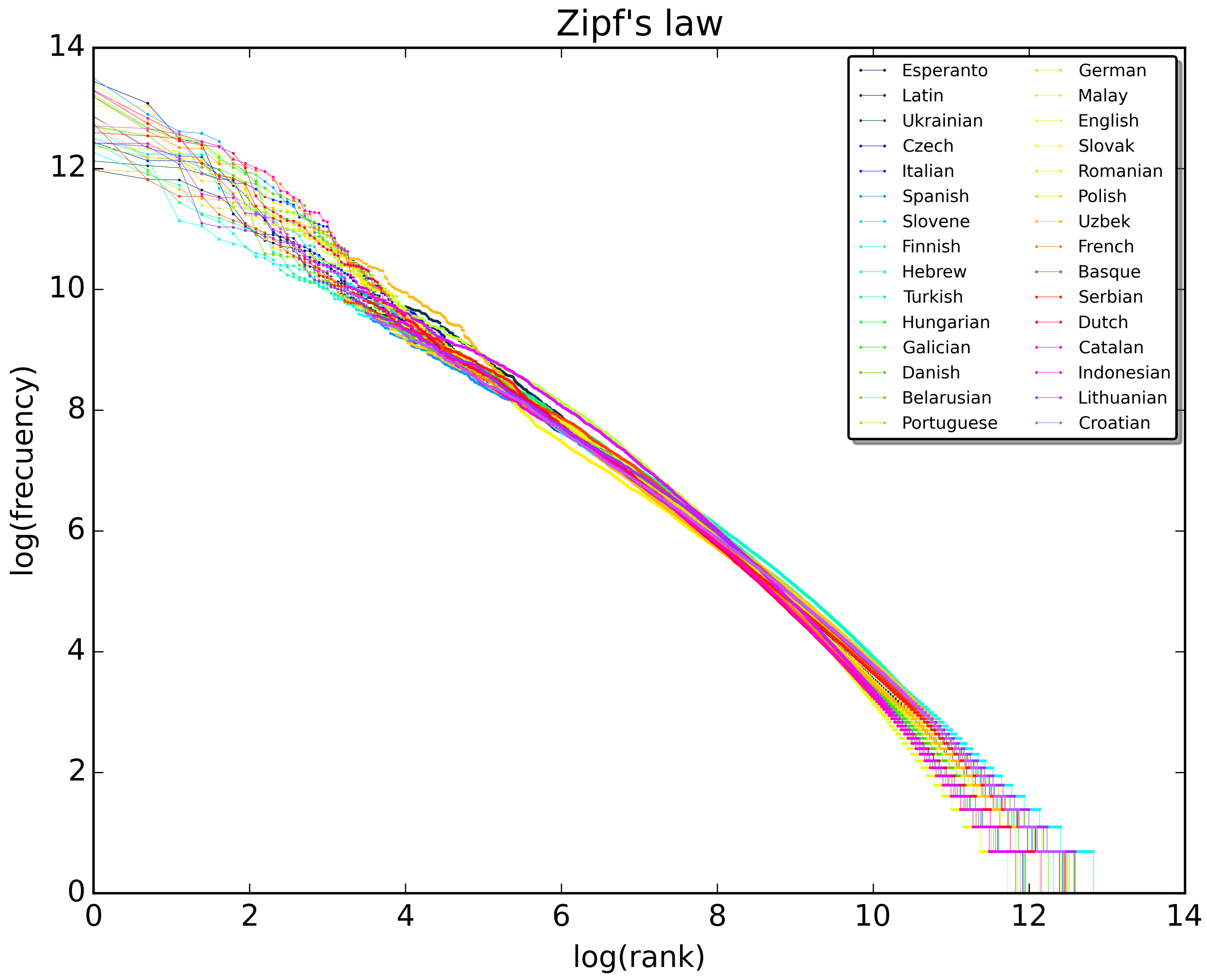
Handling Large Vocabularies

- Natural language is diverse



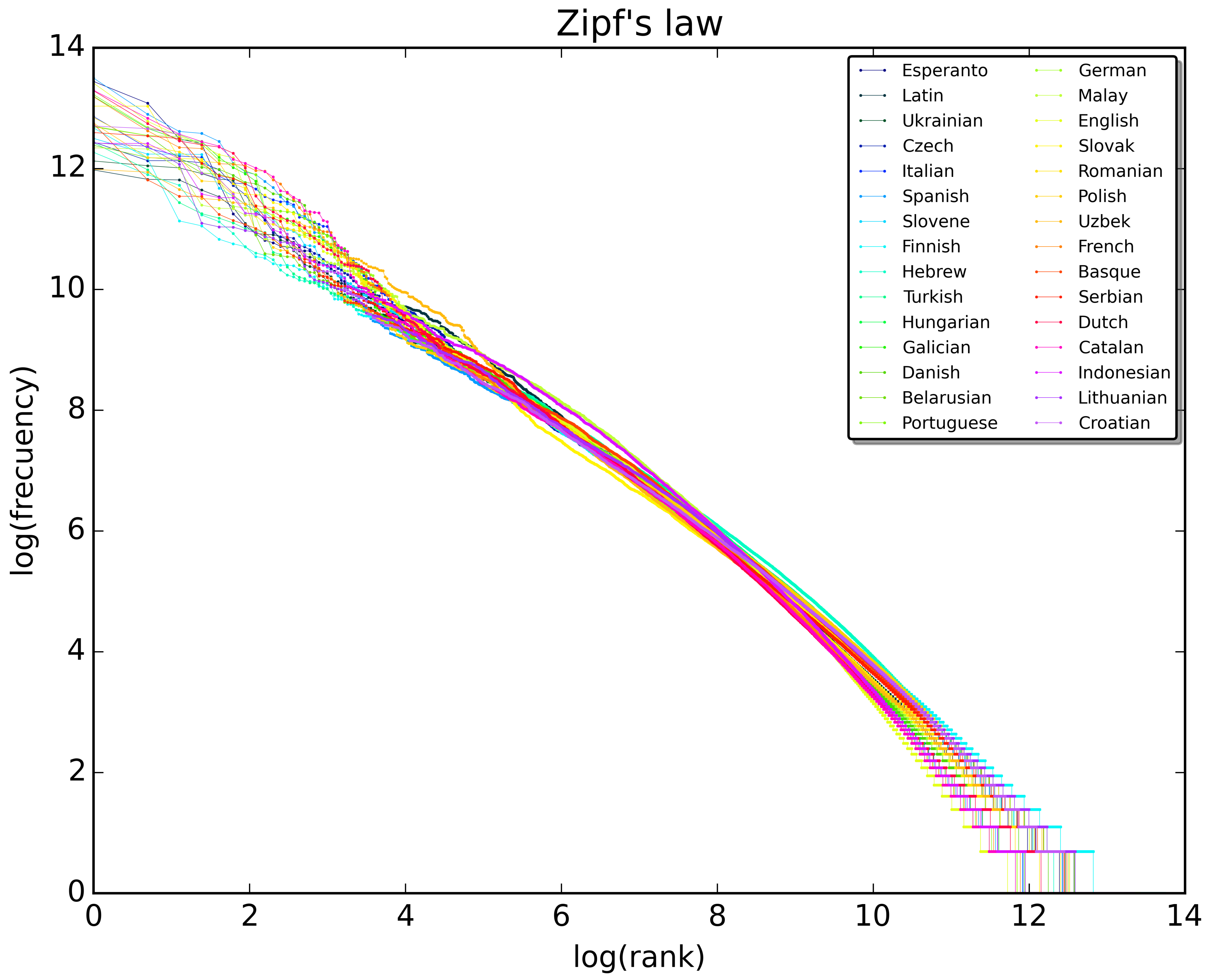
Handling Large Vocabularies

- Natural language is diverse
- We need to cover both **common** words and **rare** words



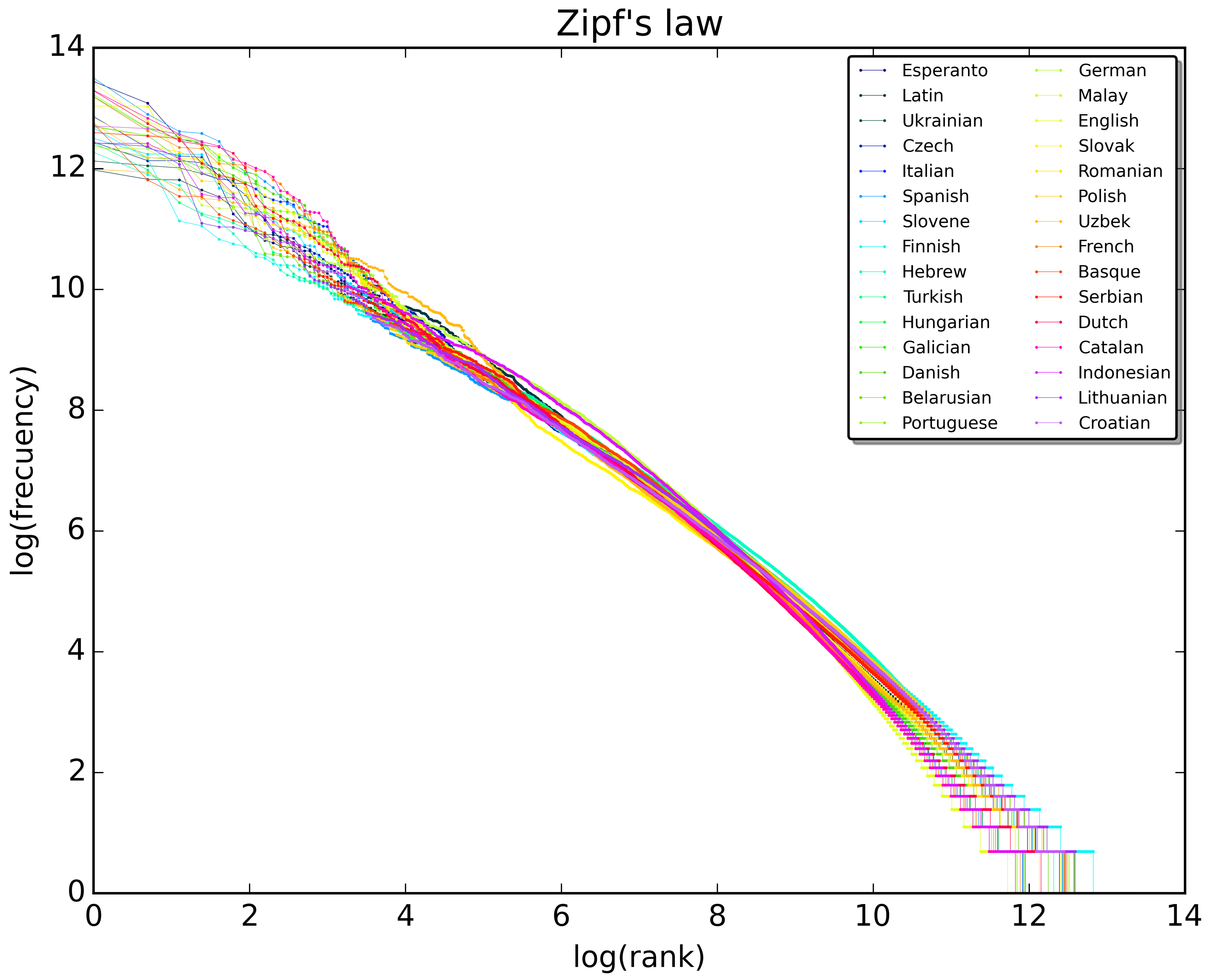
Handling Large Vocabularies

- Natural language is diverse
- We need to cover both **common** words and **rare** words
  - Using a small vocabulary (top k words) - low coverage, many unknown words



Handling Large Vocabularies

- Natural language is diverse
- We need to cover both **common** words and **rare** words
  - Using a small vocabulary (top k words) - low coverage, many unknown words
  - Using a large vocabulary - sparse, requires more parameters - slow



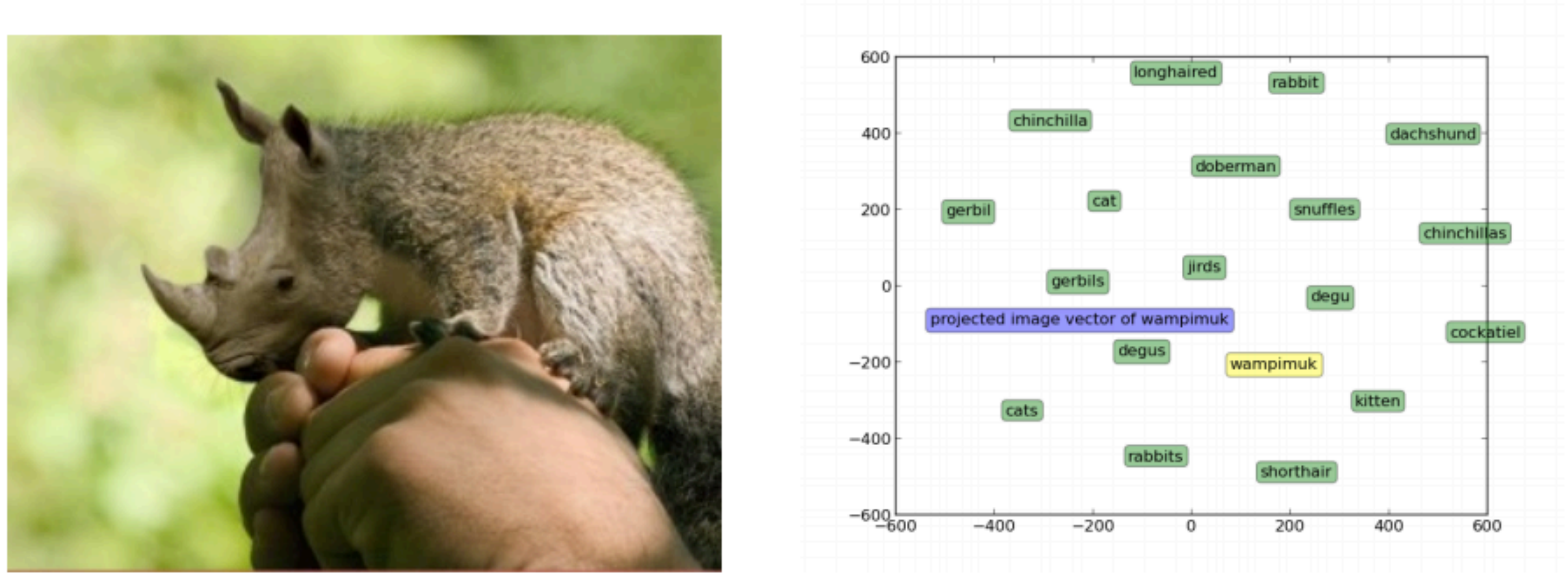
How do we handle “unknown” words?

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:



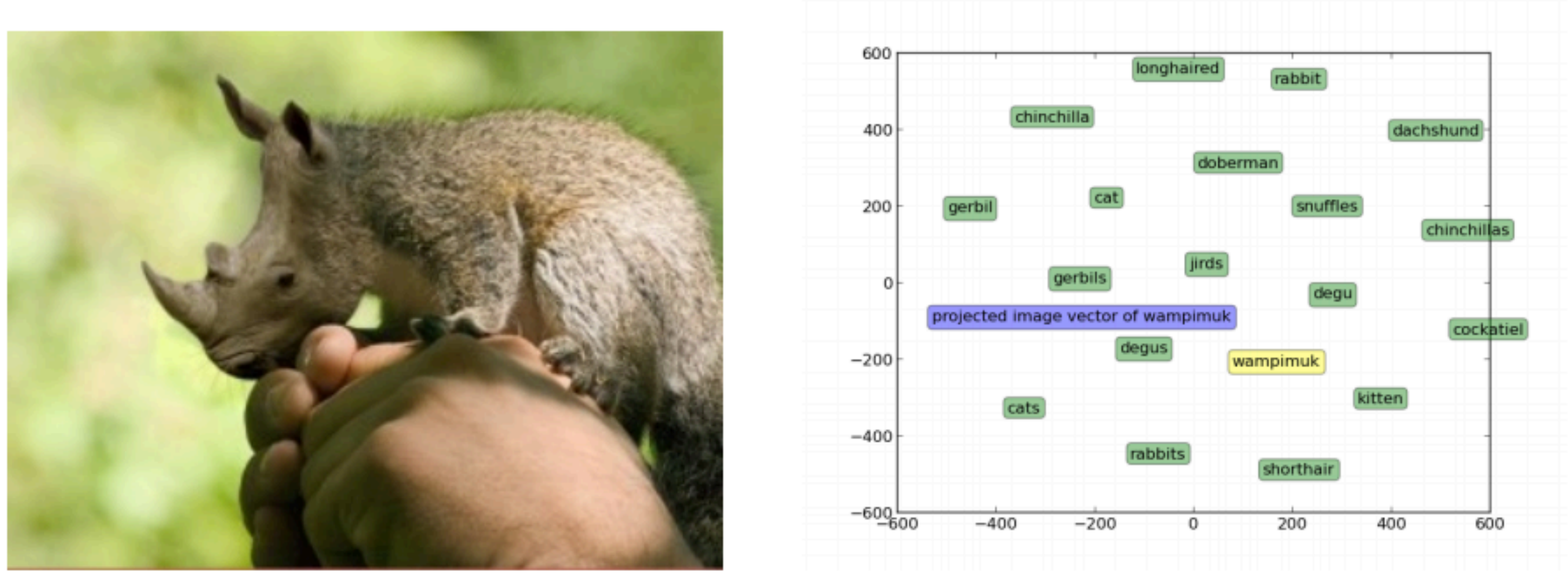
(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:
- We can’t use an inifinite vocabulary…



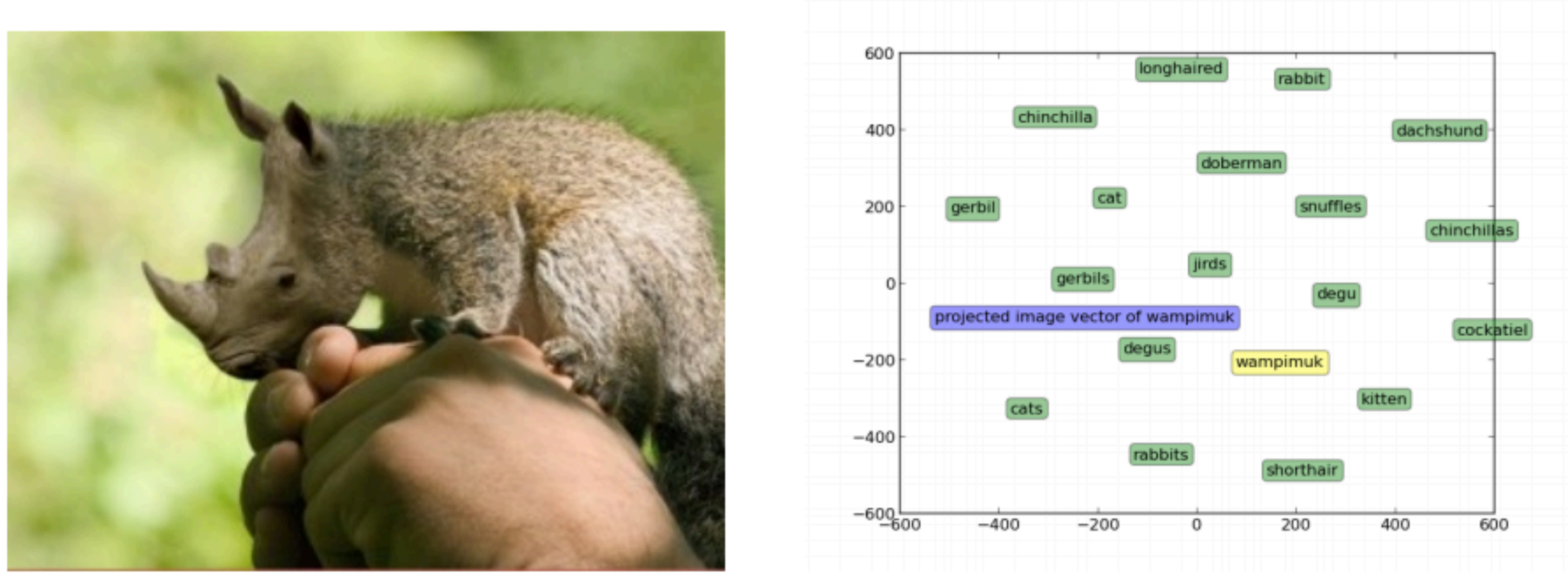
(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:
- We can’t use an inifinite vocabulary…
- “UNK” token - replace each unknown word with an “UNK” symbol



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

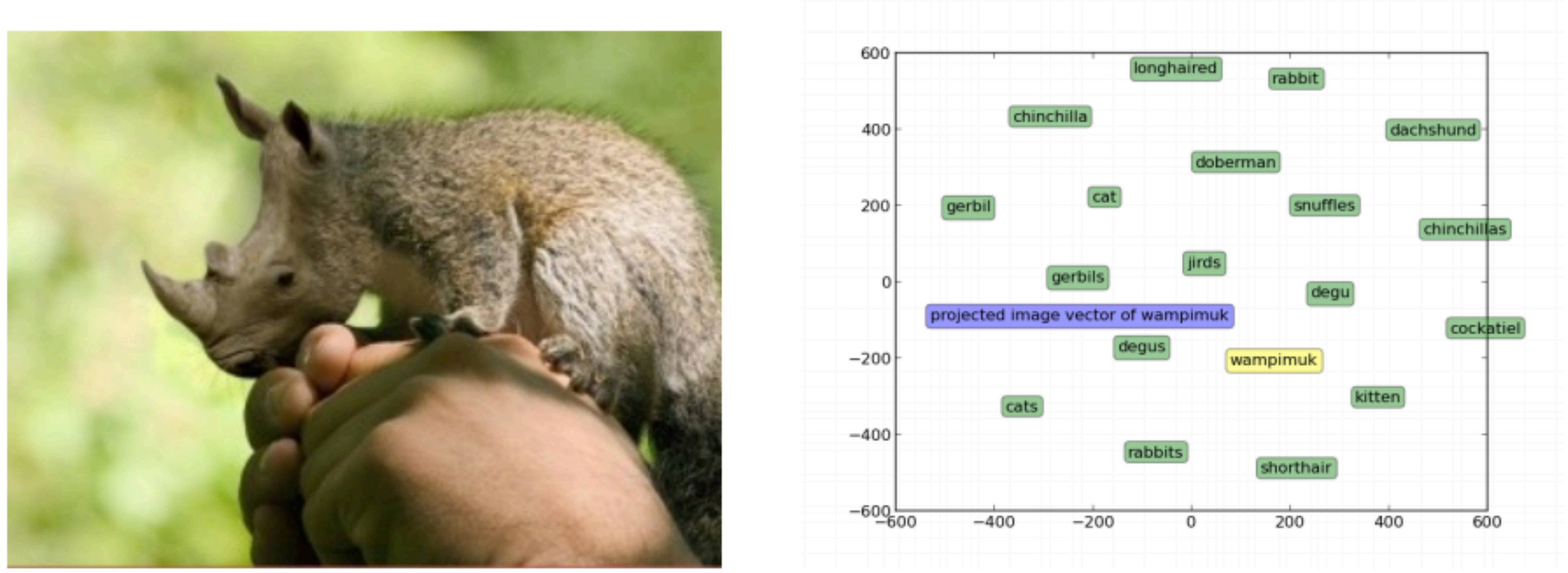
from Lazaridou et al. 2014

This is a Wampimuk in the wild .

This is a *UNK* in the wild .

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:
- We can’t use an inifinite vocabulary…
- “UNK” token - replace each unknown word with an “UNK” symbol
    - Good: Enables to encode any sentence



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

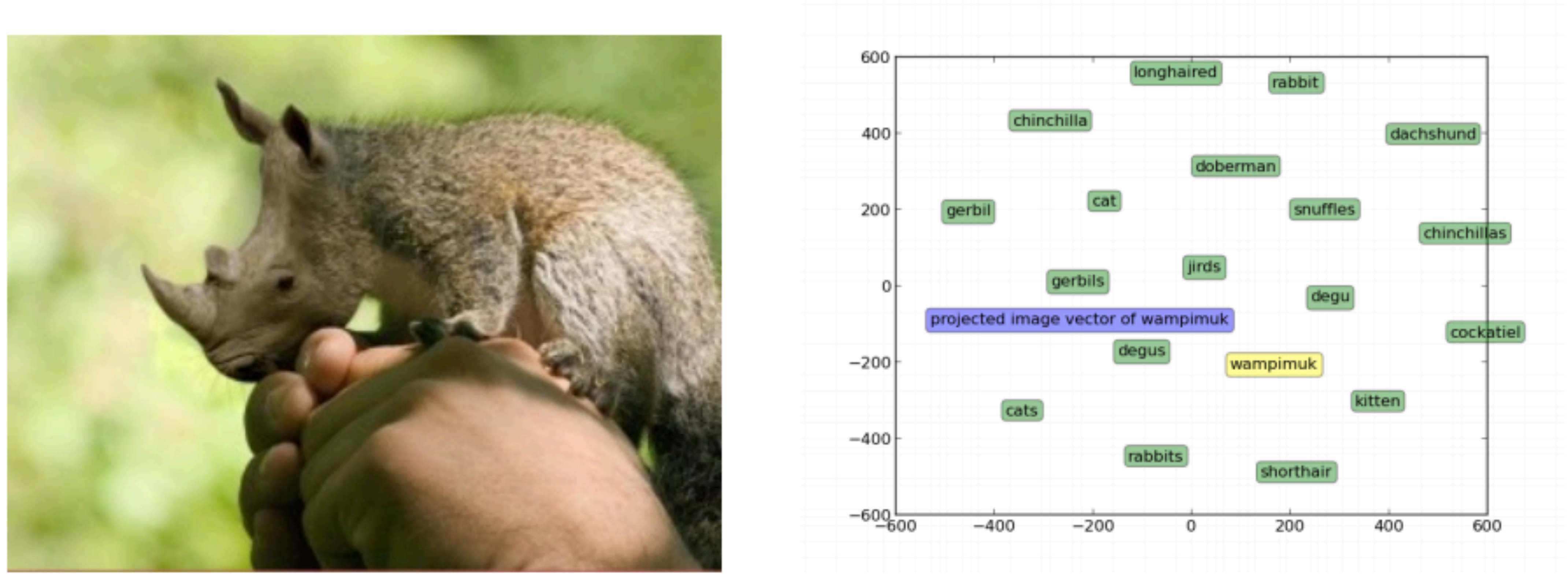
<u>**from Lazaridou et al. 2014**</u>

This is a Wampimuk in the wild .

This is a *UNK* in the wild .

How do we handle “unknown” words?

- Unknown words are inevitable - new words are always invented around us:
- We can’t use an inifinite vocabulary…
- “UNK” token - replace each unknown word with an “UNK” symbol
  - Good: Enables to encode any sentence
  - Bad: Throws away information…



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

This is a Wampimuk in the wild .

↓

This is a *UNK* in the wild .

How do we handle "unknown" words?

- Unknown words are inevitable - new words are always invented around us:
- We can't use an inifinite vocabulary…
- "UNK" token - replace each unknown word with an "UNK" symbol
  - Good: Enables to encode any sentence
  - Bad: Throws away information…
- How can we do better?

(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

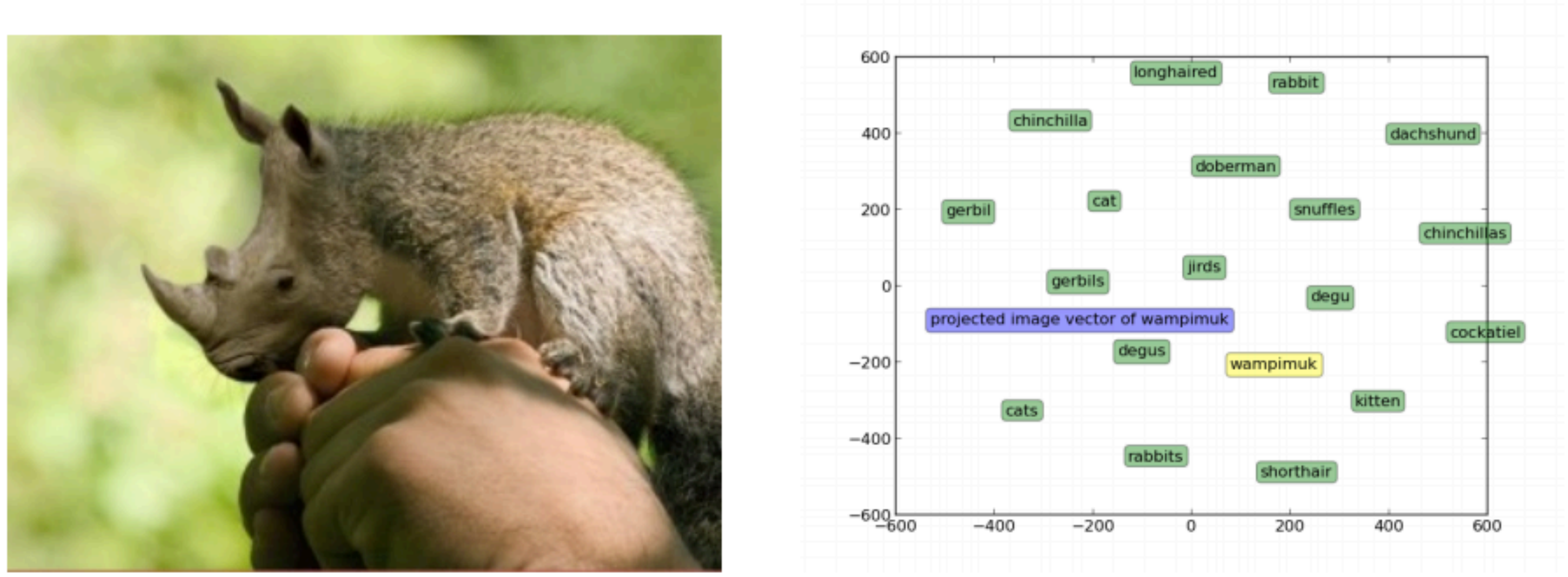
from Lazaridou et al. 2014

This is a Wampimuk in the wild .

↓

This is a *UNK* in the wild .

Character Level MT



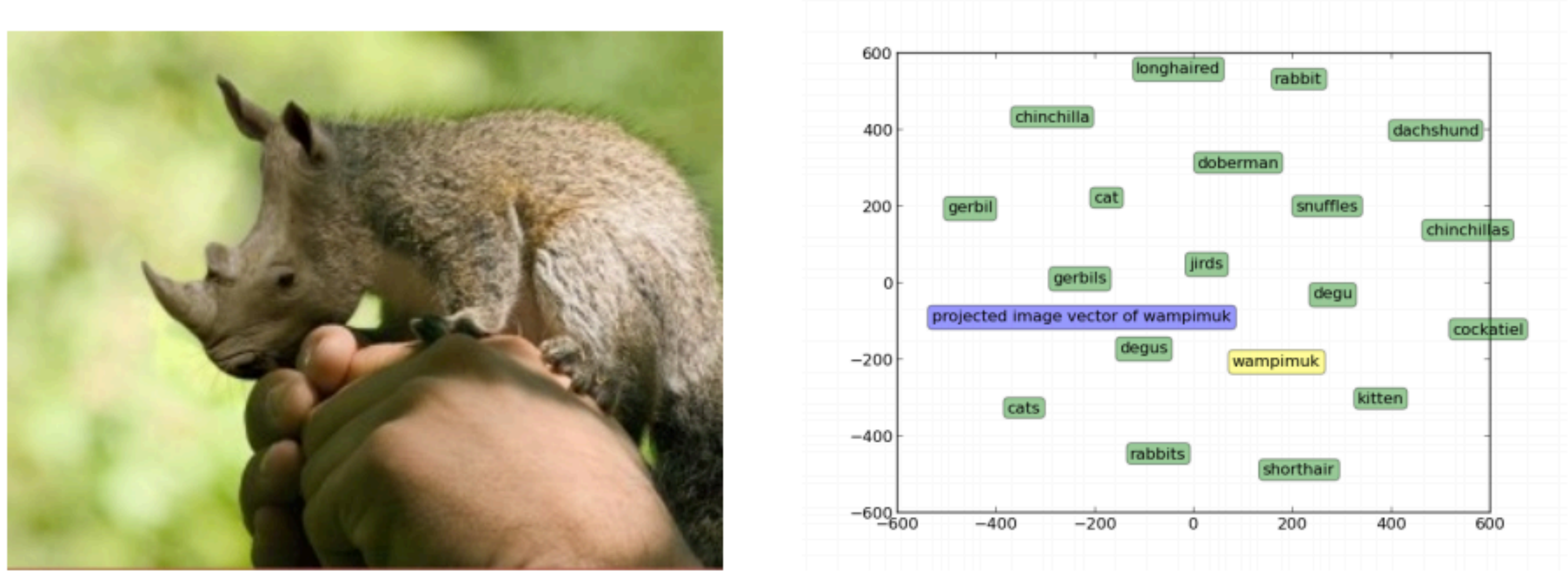
(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

Character Level MT

- An simple solution - work at the character level:



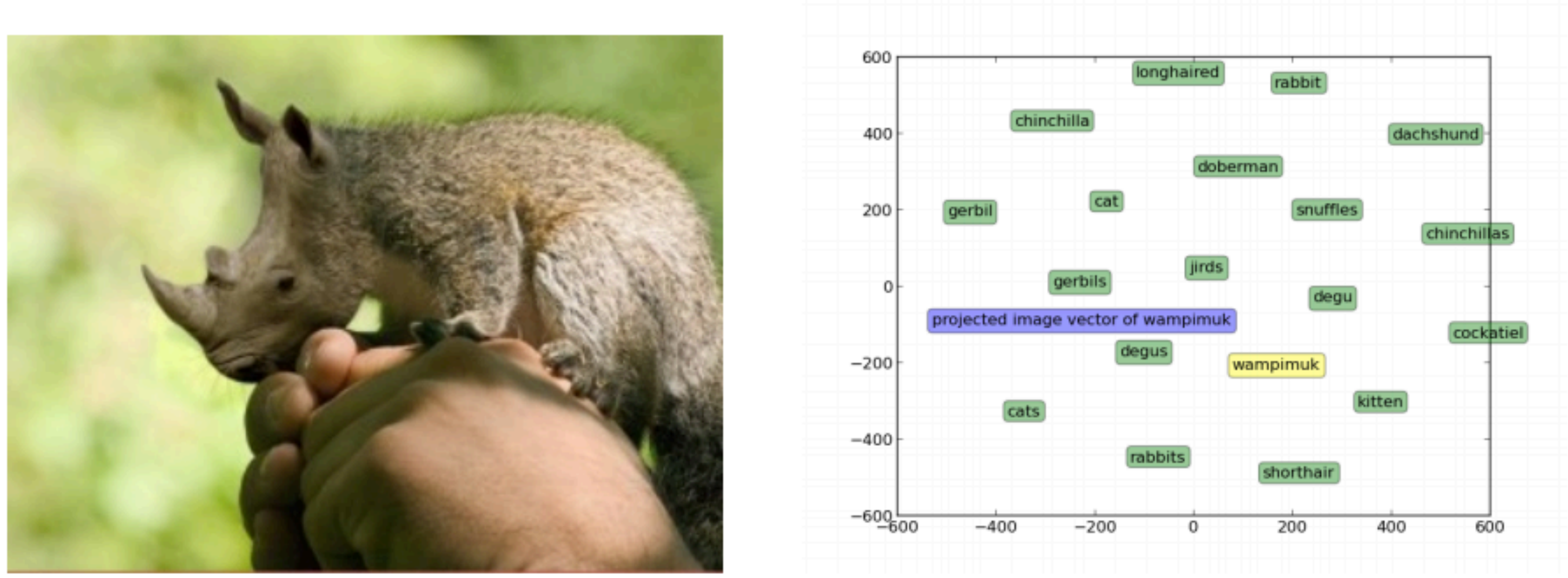
(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

Character Level MT

- An simple solution - work at the character level:



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

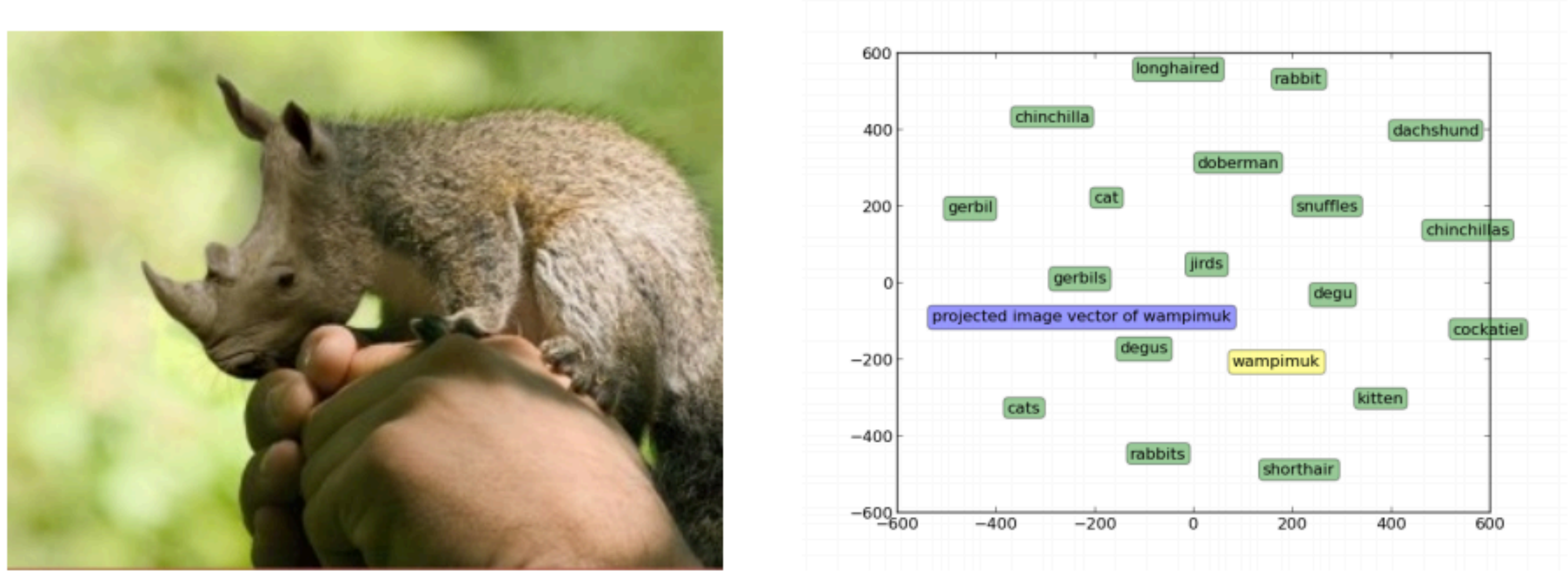
from Lazaridou et al. 2014

This is a Wampimuk in the wild .

T h i s _ i s _ a _ *W a m p i m u k* _ i n _ t h e _ w i l d _ .

Character Level MT

- An simple solution - work at the character level:
    - No unknown words! Models **morphology**



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

from Lazaridou et al. 2014

This is a Wampimuk in the wild .

↓

T h i s _ i s _ a _ *W a m p i m u k* _ i n _ t h e _ w i l d _ .

Character Level MT

- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow...

(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

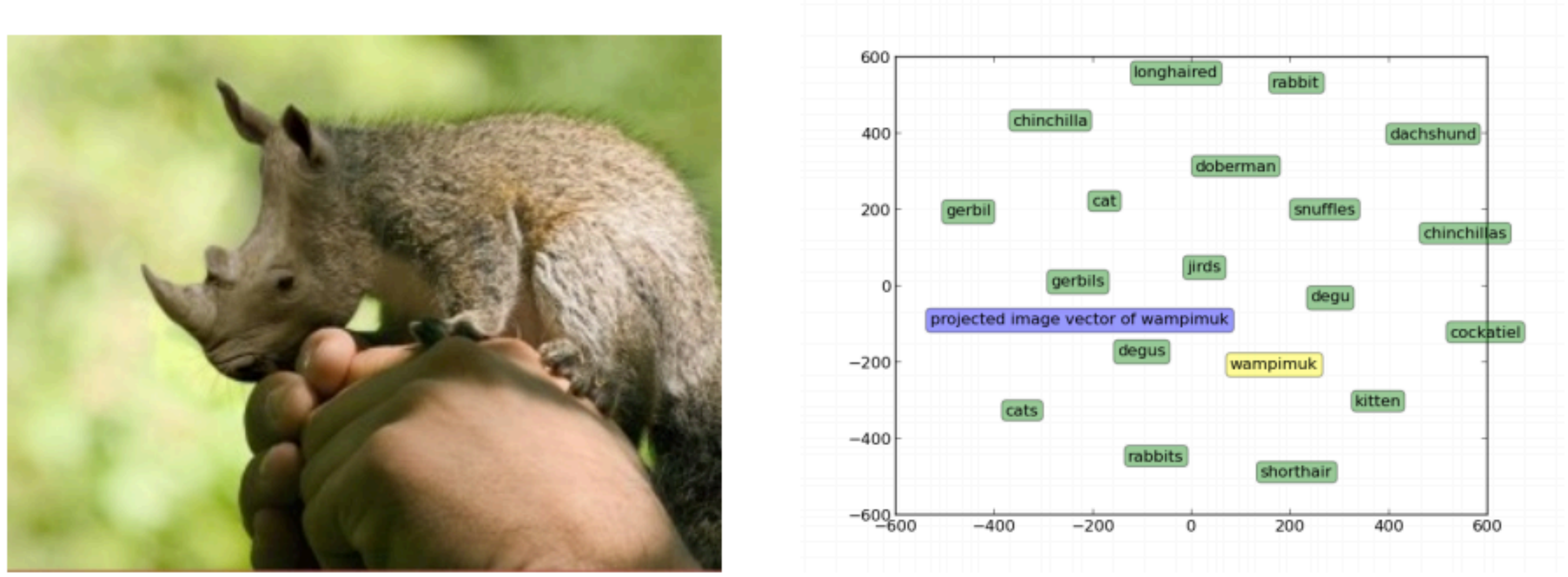
from Lazaridou et al. 2014

This is a Wampimuk in the wild .

T h i s _ i s _ a _ *W a m p i m u k* _ i n _ t h e _ w i l d _ .

Character Level MT

- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow…
- How do we model this?



(a) (b)

Figure 1: A potential *wampimuk* (a) together with its projection onto the linguistic space (b).

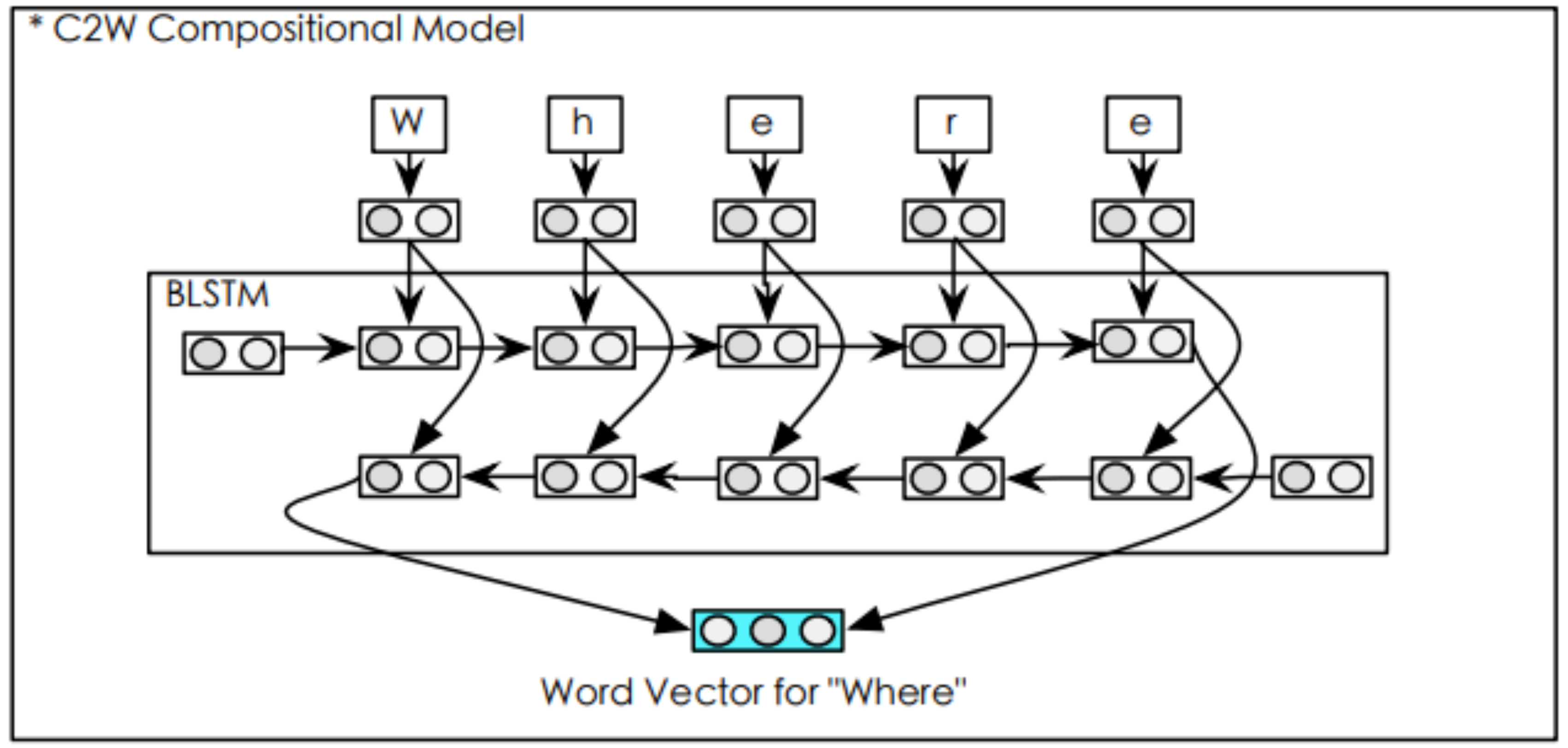
from Lazaridou et al. 2014

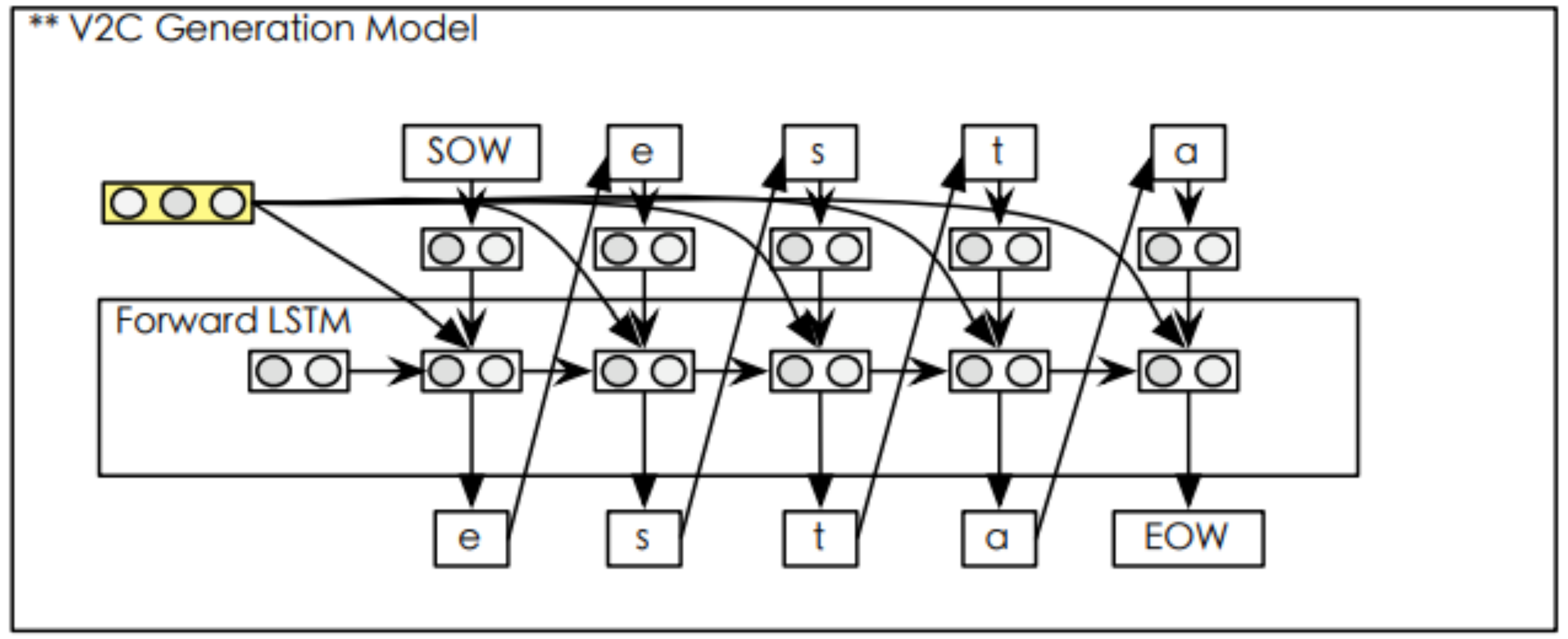
This is a Wampimuk in the wild .

T h i s _ i s _ a _ *W a m p i m u k* _ i n _ t h e _ w i l d _ .

Character Level MT

- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow...
- How do we model this?
  - Ling et al. 2015 - Char2Vec and Vec2Char with LSTMs





This _ is _ a _ *Wampimuk* _ in _ the _ wild _ .

Character Level MT

- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow...
- How do we model this?
  - Ling et al. 2015 - Char2Vec and Vec2Char with LSTMs
  - Costa Jussa et al 2016 - using word-level convolutions (faster)

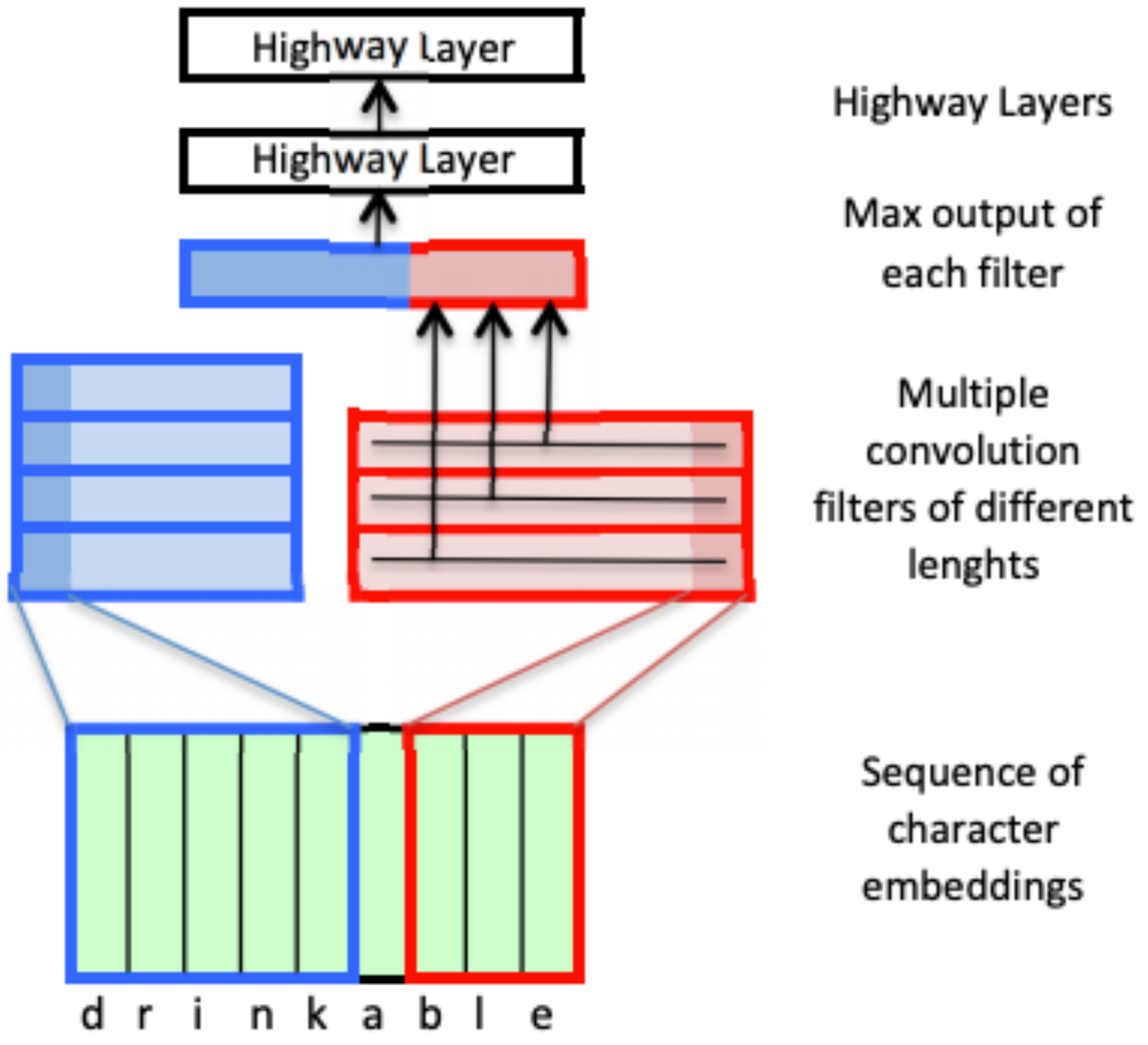
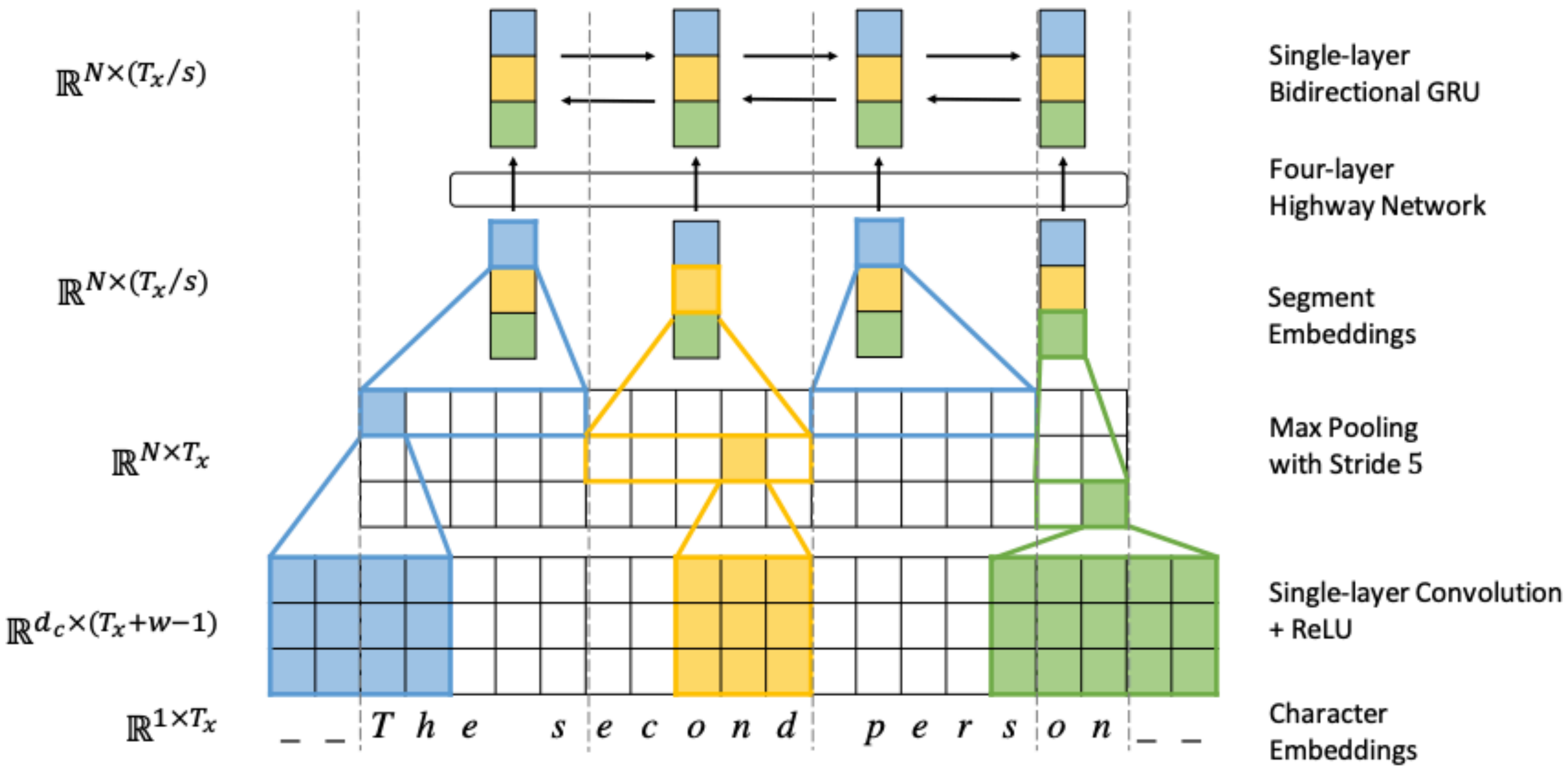


Figure 1: Character-based word embedding

T h i s _ i s _ a _ *W a m p i m u k* _ i n _ t h e _ w i l d _ .

Character Level MT

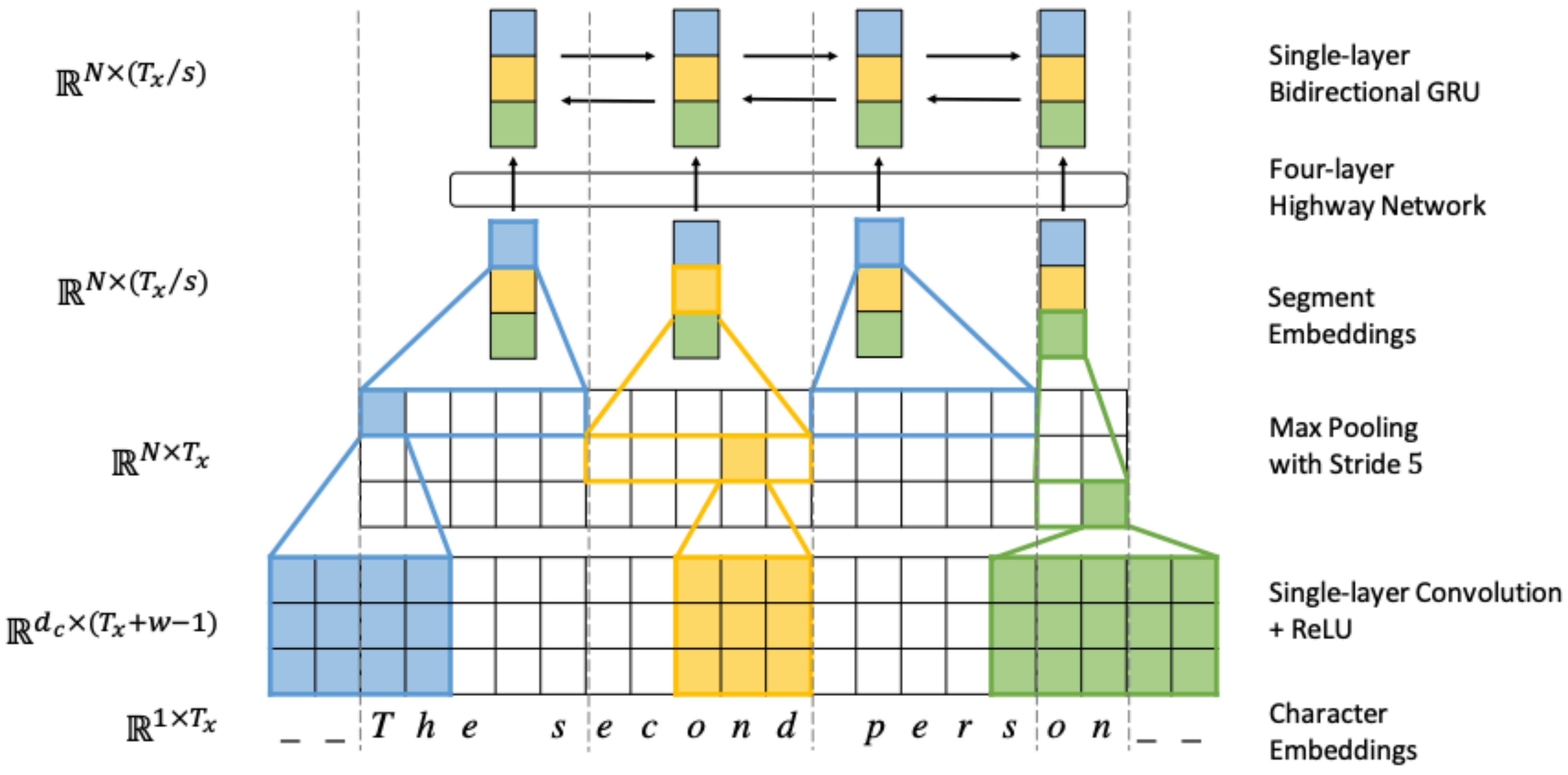
- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow...
- How do we model this?
  - Ling et al. 2015 - Char2Vec and Vec2Char with LSTMs
  - Costa Jussa et al 2016 - using word-level convolutions (faster)
  - Chung et al. 2016, Lee et al. 2016 - No need for word segmentation!



T h i s i s a *W a m p i m u k* i n t h e w i l d .

Character Level MT

- An simple solution - work at the character level:
  - No unknown words! Models **morphology**
  - Very long sequences - slow…
- How do we model this?
  - Ling et al. 2015 - Char2Vec and Vec2Char with LSTMs
  - Costa Jussa et al 2016 - using word-level convolutions (faster)
  - Chung et al. 2016, Lee et al. 2016 - No need for word segmentation!
- Requires **deep models** to work well (Cherry et al 2018)

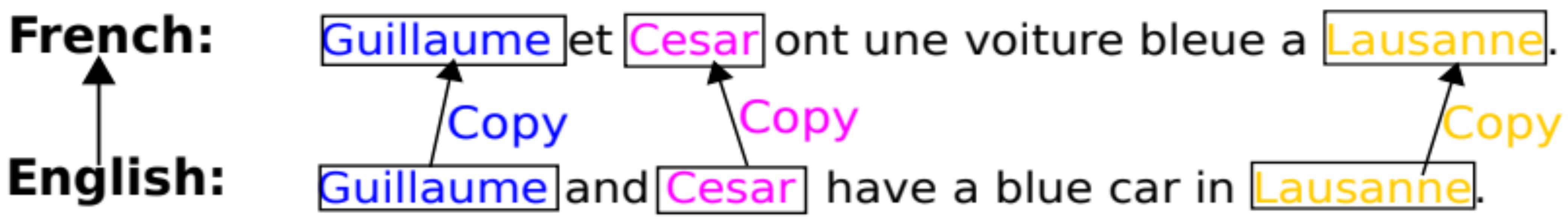


T h i s i s a *W a m p i m u k* i n t h e w i l d .

Seq2Seq with a Copy Mechanism

Seq2Seq with a Copy Mechanism

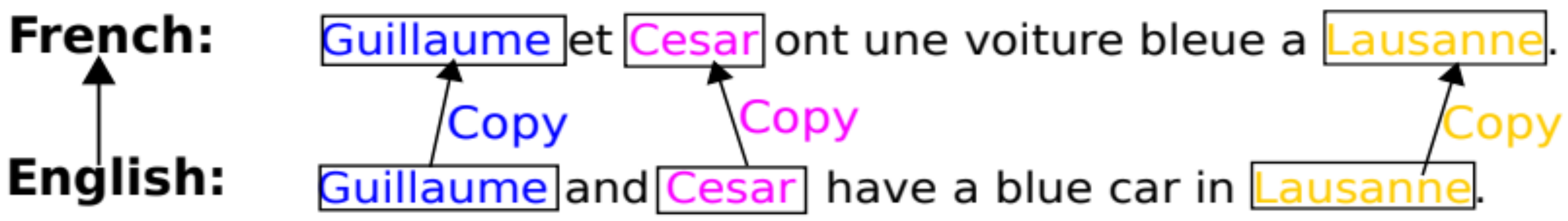
- Another solution for unknown words: “copy” them “as-is" from source



Seq2Seq with a Copy Mechanism

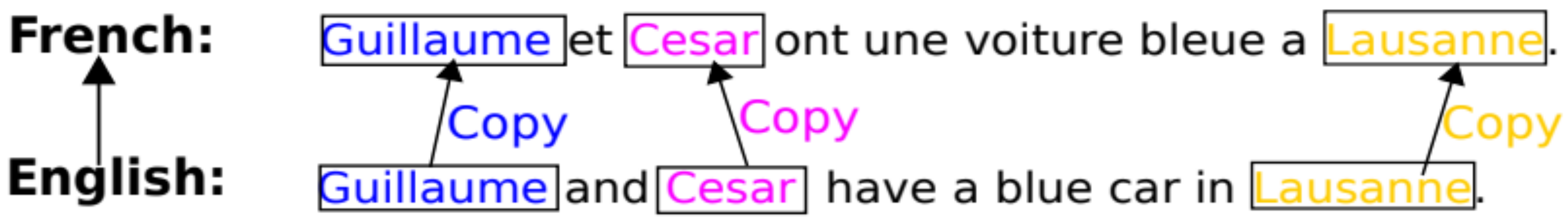
- Another solution for unknown words: “copy” them “as-is" from source

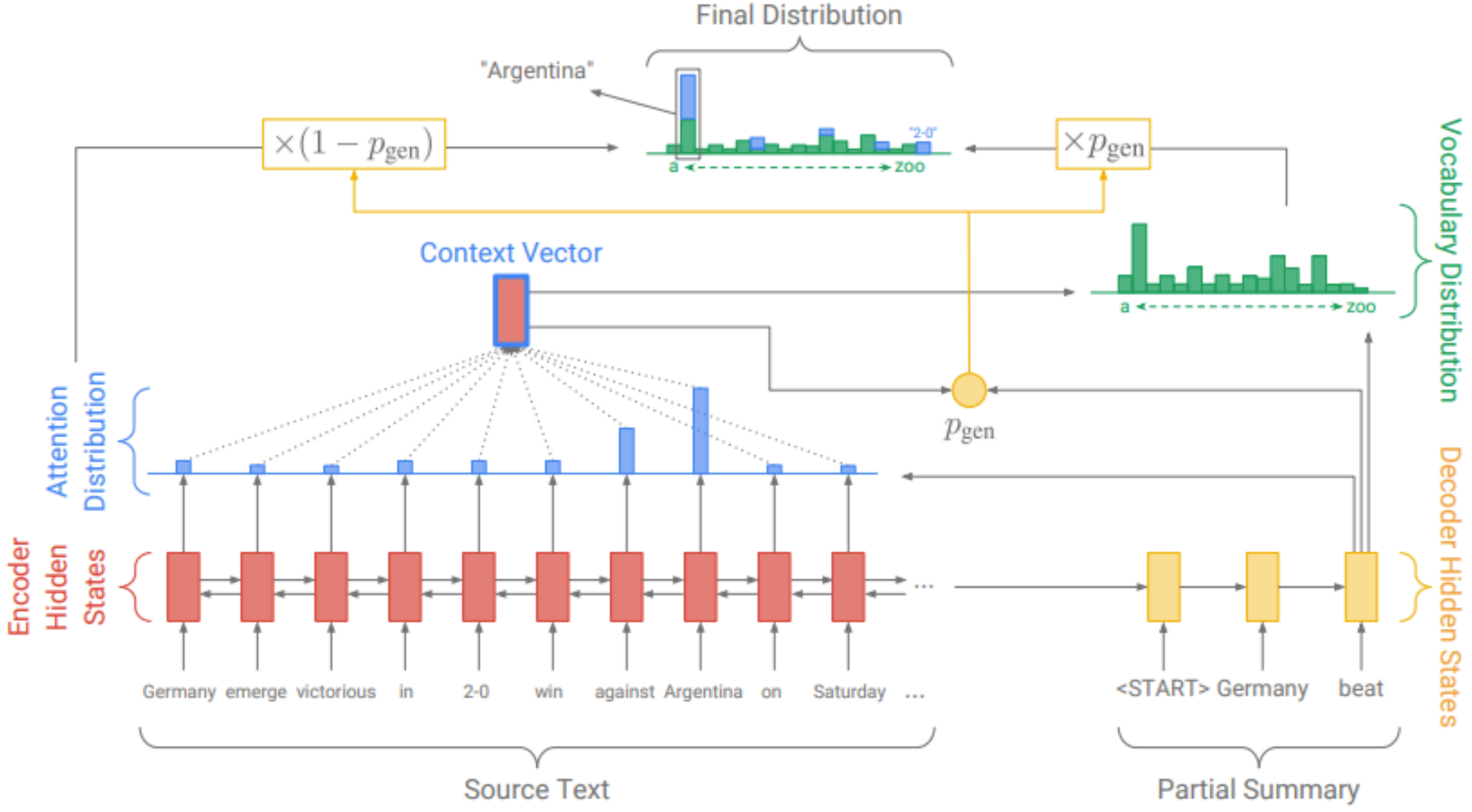
- “Pointing the Unknown Words” (Gulchere et al 2016)



Seq2Seq with a Copy Mechanism

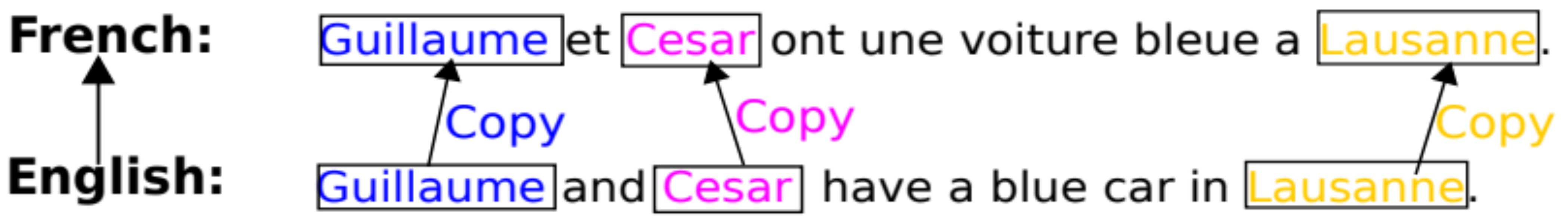
- Another solution for unknown words: “copy” them “as-is" from source
- “Pointing the Unknown Words” (Gulchere et al 2016)
- Interpolate the attention distribution and the softmax distribution

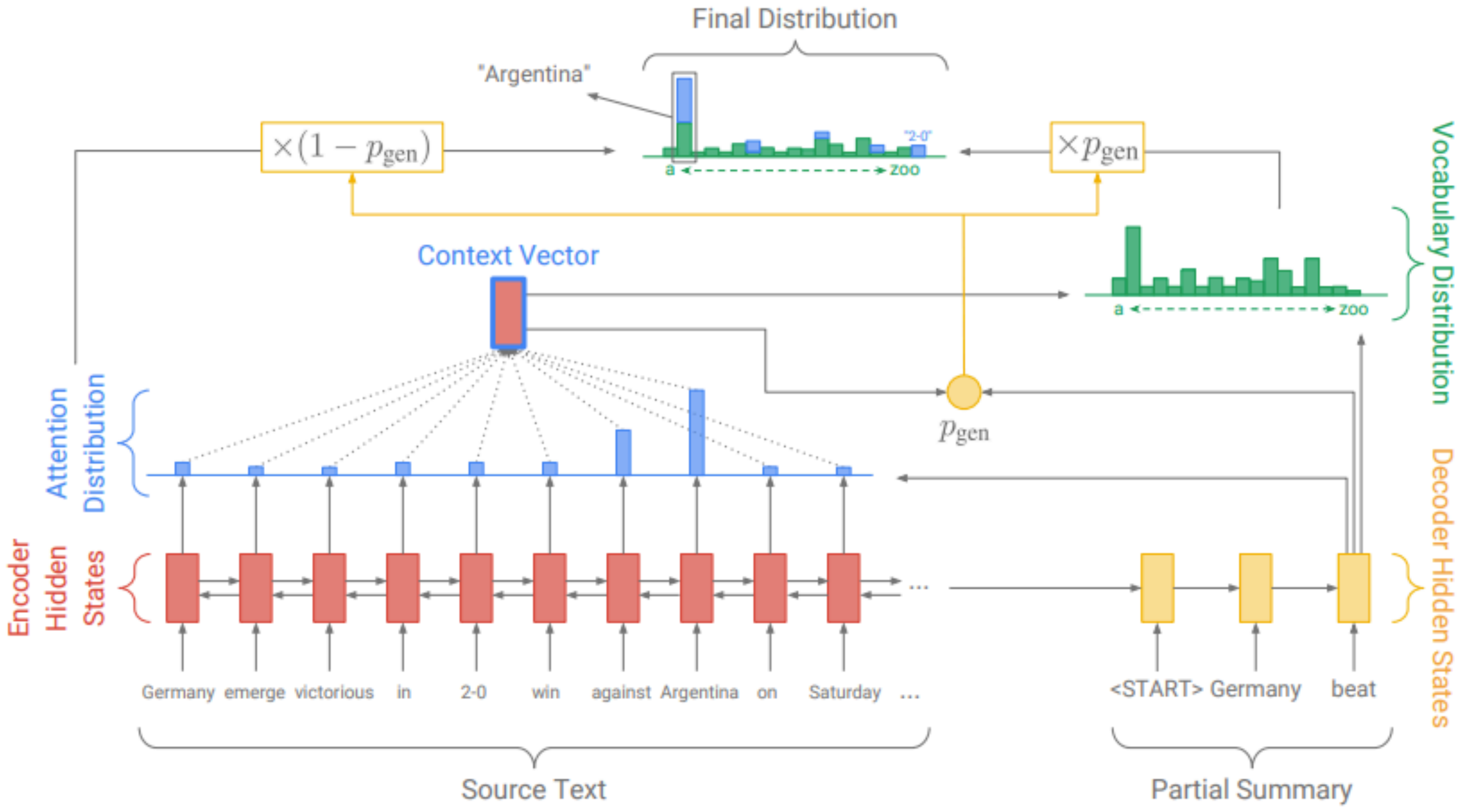




Seq2Seq with a Copy Mechanism

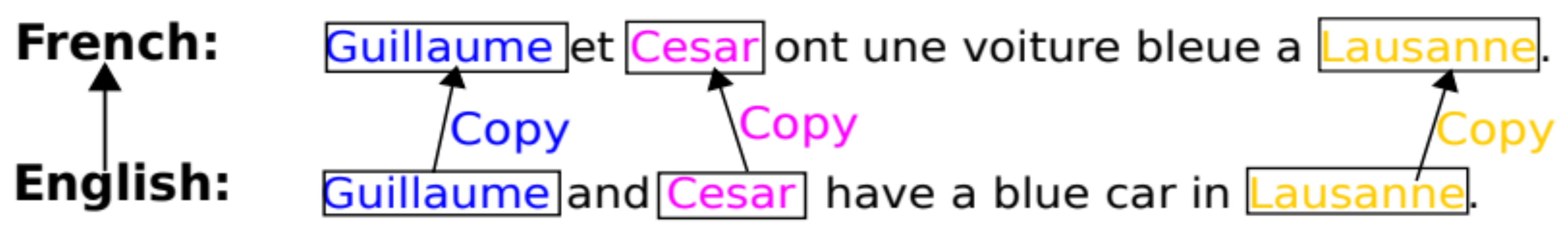
- Another solution for unknown words: "copy" them "as-is" from source
- "Pointing the Unknown Words" (Gulchere et al 2016)
- Interpolate the attention distribution and the softmax distribution
- Useful in summarization tasks (Gu et al 2016, See et al. 2017)

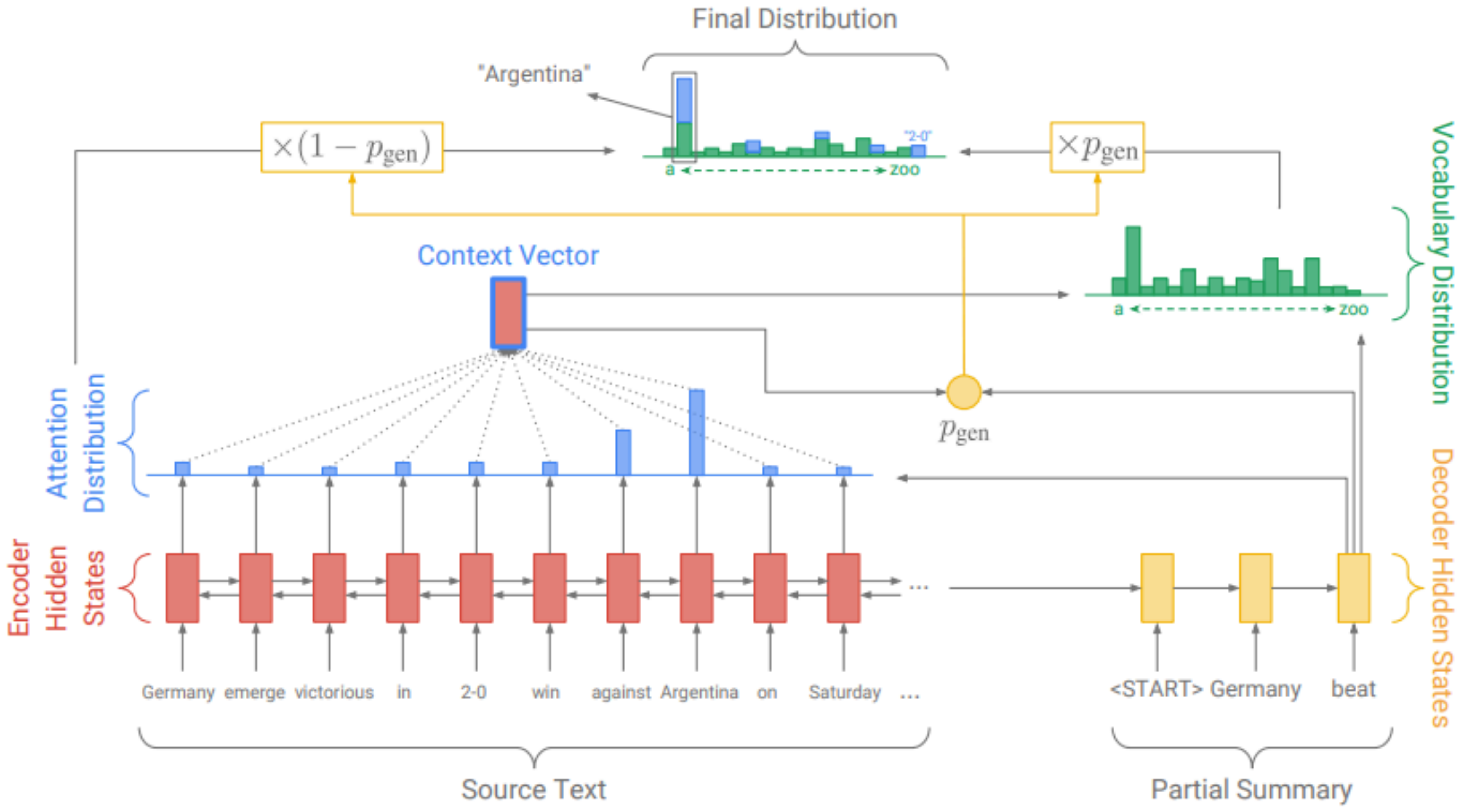




Seq2Seq with a Copy Mechanism

- Another solution for unknown words: “copy” them “as-is" from source
- “Pointing the Unknown Words” (Gulchere et al 2016)
- Interpolate the attention distribution and the softmax distribution
- Useful in summarization tasks (Gu et al 2016, See et al. 2017)
- Problem - can’t copy in all cases





A practical middle ground: BPE

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- "Neural Machine Translation of Rare Words with Subword Units" Sennrich et al, 2015
- Uses the "Byte-Pair Encoding" compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair
  - Merge it to a new symbol

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- "Neural Machine Translation of Rare Words with Subword Units" Sennrich et al, 2015
- Uses the "Byte-Pair Encoding" compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair
  - Merge it to a new symbol
  - Repeat until the desired vocal size

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair
  - Merge it to a new symbol
  - Repeat until the desired vocal size

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
```

This is a shot of C@@ ann@@ ery R@@ ow in 19@@ 32 .

זהו צילום של ק@@ אנ@@ ארי רו ב-19@@ 32 .

```
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair
  - Merge it to a new symbol
  - Repeat until the desired vocal size
- The current standard for word segmentation in NLP applications (1900+ citations)

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
```

This is a shot of C@@ ann@@ ery R@@ ow in 19@@ 32 .

זהו צילום של ק@@ אנ@@ ארי רו ב-19@@ 32 .

```
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

A practical middle ground: BPE

- “Neural Machine Translation of Rare Words with Subword Units” Sennrich et al, 2015
- Uses the “Byte-Pair Encoding” compression algorithm (Gage, 1994):
  - Start bottom up from characters as symbols
  - Pick the most common symbol pair
  - Merge it to a new symbol
  - Repeat until the desired vocal size
- The current standard for word segmentation in NLP applications (1900+ citations)
- **Controllable vocabulary size, no UNKs!**

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
```

This is a shot of C@@ ann@@ ery R@@ ow in 19@@ 32 .

זהו צילום של ק@@ אנ@@ ארי רו ב-19@@ 32 .

```
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

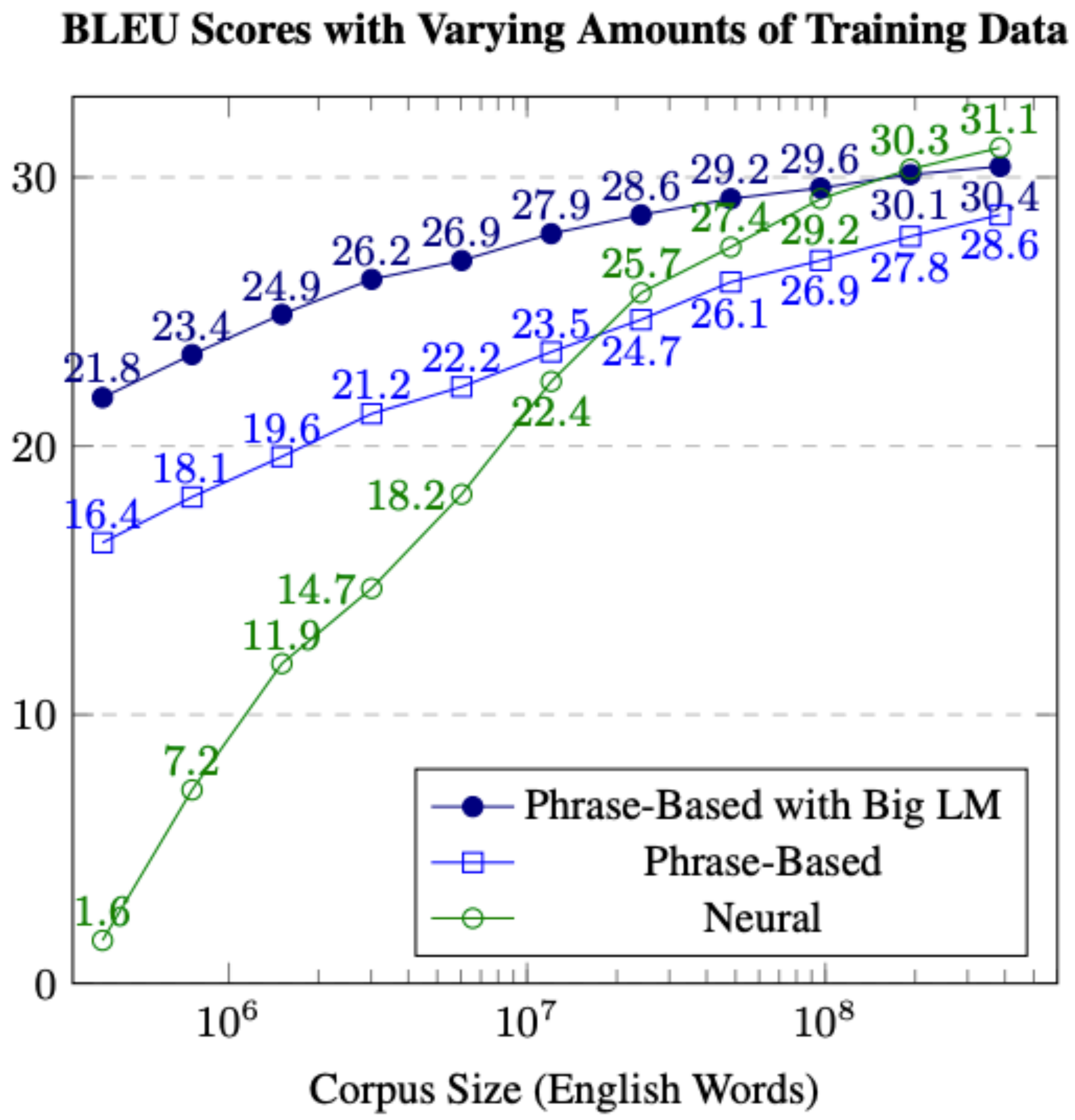
Using Monolingual Data

Using Monolingual Data

- Statistical MT used language models extensively. What about NMT?

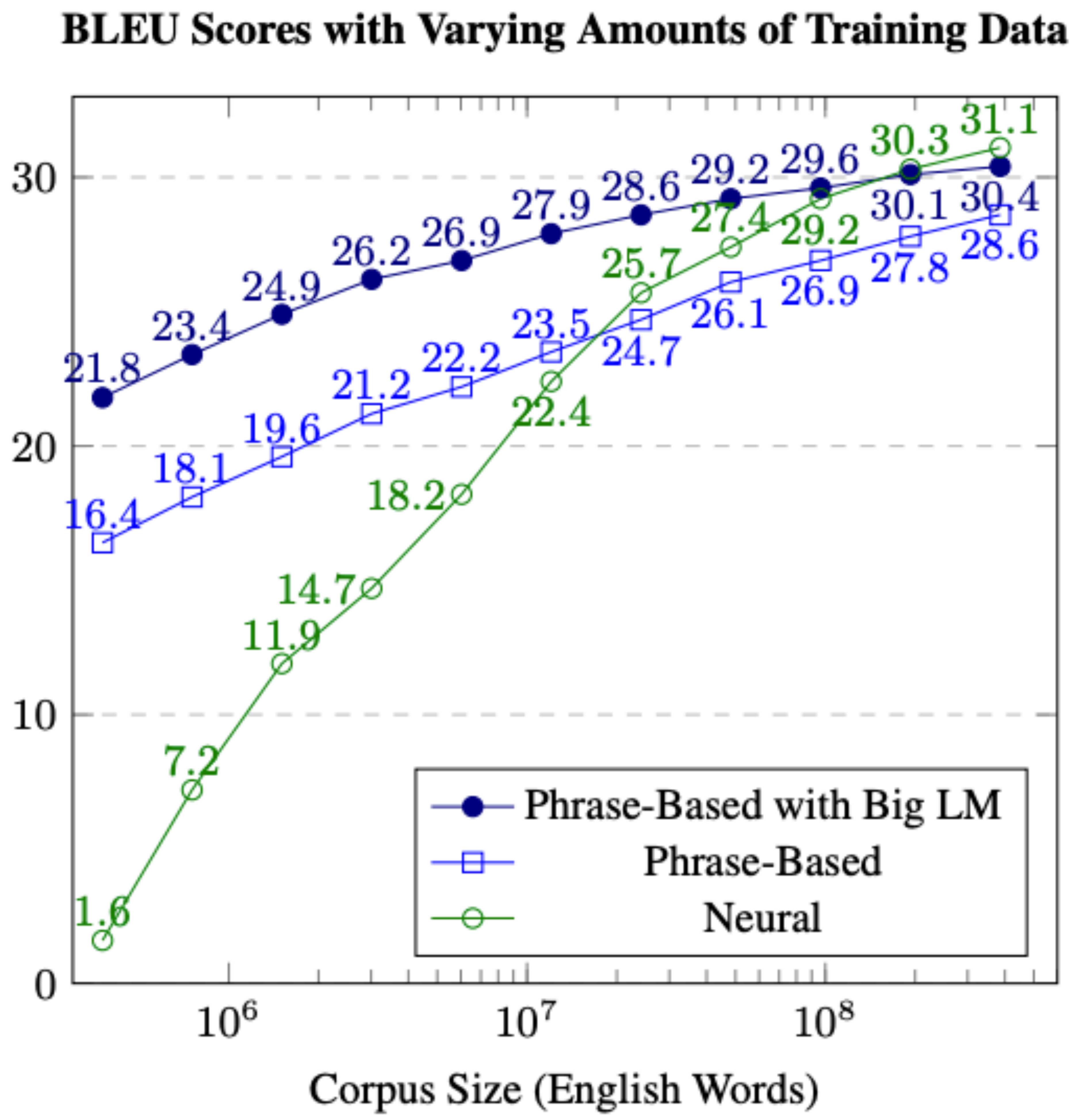
Using Monolingual Data

- Statistical MT used language models extensively. What about NMT?
- Koehn & Knowles 2017



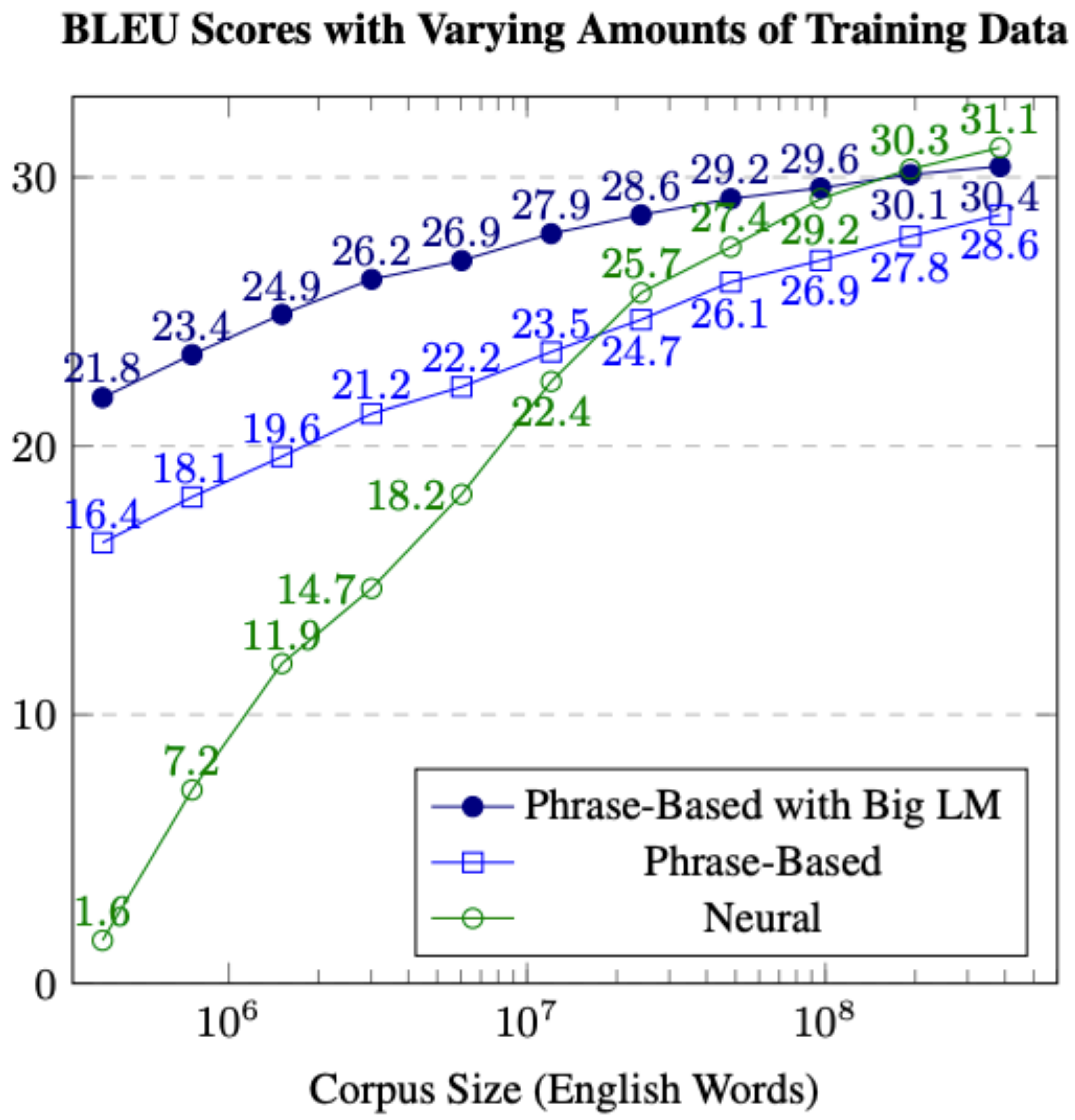
Using Monolingual Data

- Statistical MT used language models extensively. What about NMT?
- Koehn & Knowles 2017
    - SMT is better in low resource settings



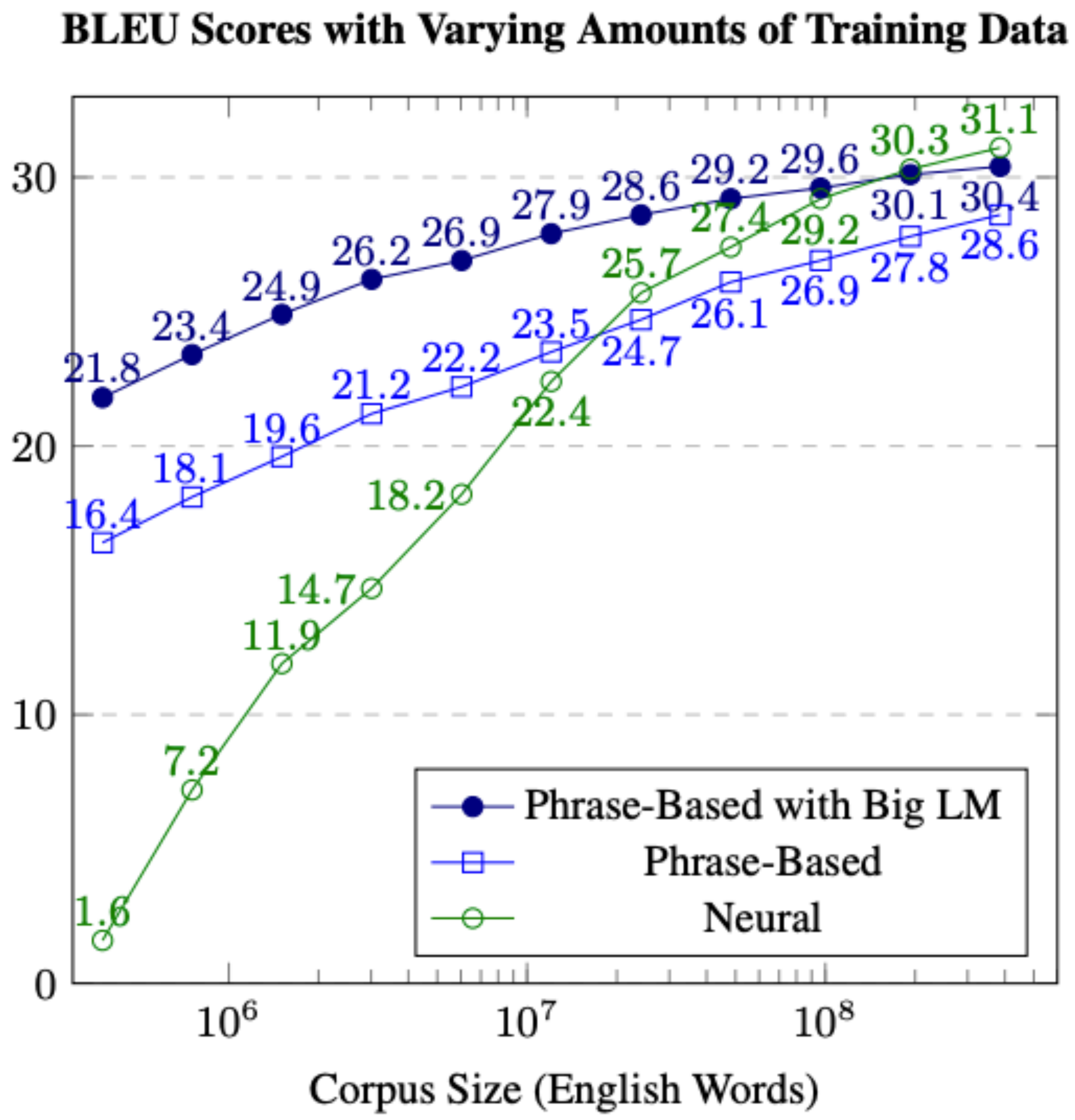
Using Monolingual Data

- Statistical MT used language models extensively. What about NMT?
- Koehn & Knowles 2017
  - SMT is better in low resource settings
  - Especially with a LM



Using Monolingual Data

- Statistical MT used language models extensively. What about NMT?
- Koehn & Knowles 2017
  - SMT is better in low resource settings
  - Especially with a LM
- How can we incorporate a LM into NMT?



Back-Translation

Back-Translation

- Sennrich et al. 2016

Back-Translation

- Sennrich et al. 2016
- A simpler approach - **synthesize** parallel data from monolingual data:

Back-Translation

English (real-parallel) $\longrightarrow$ German (real-parallel) **Train**

- Sennrich et al. 2016
- A simpler approach - **synthesize** parallel data from monolingual data:
    - Train a “reverse” model with the available parallel data

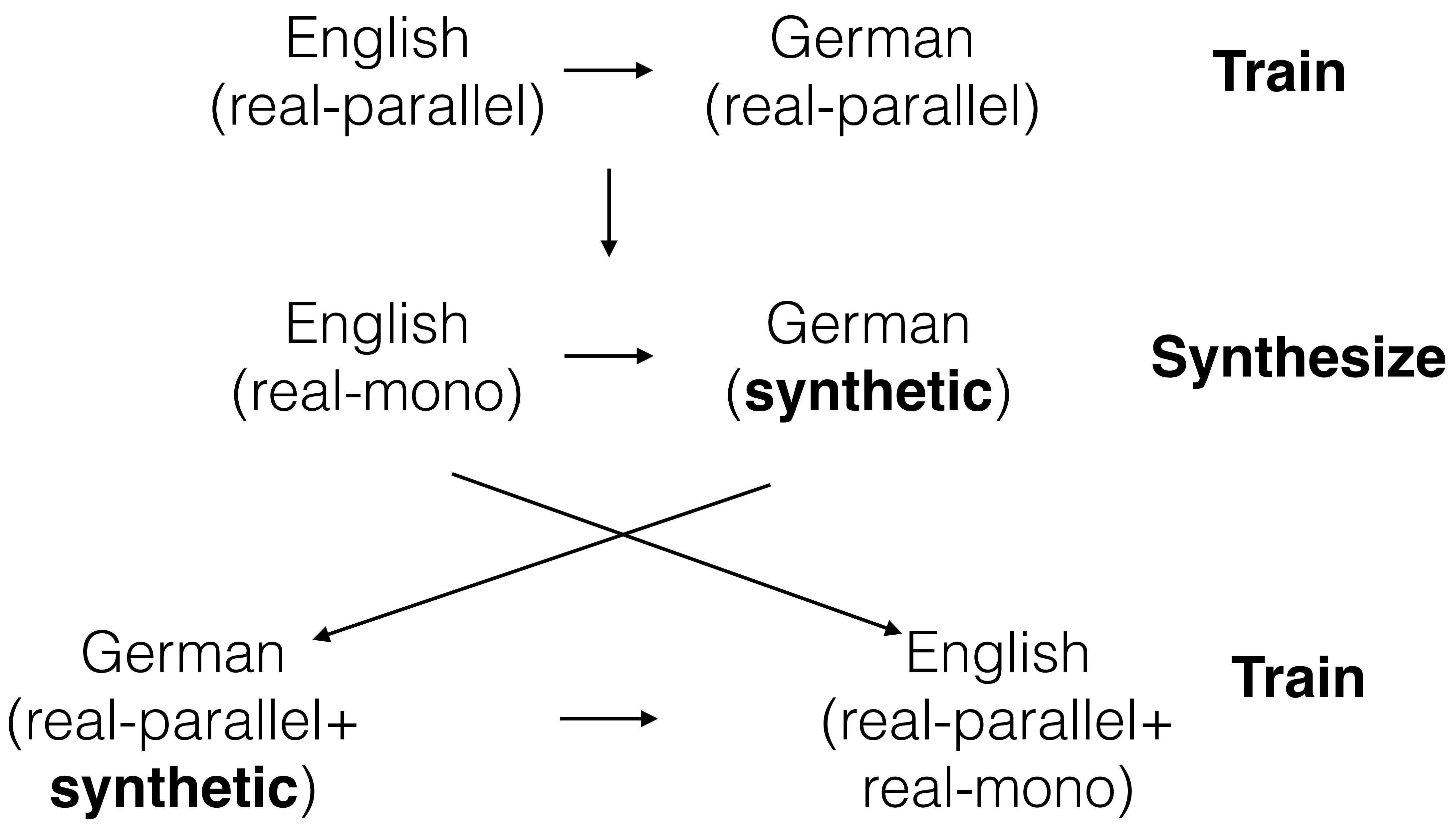
Back-Translation

- Sennrich et al. 2016
- A simpler approach - **synthesize** parallel data from monolingual data:
  - Train a “reverse” model with the available parallel data
  - Translate the monolingual data using the reverse model

English (real-parallel)	⟶	German (real-parallel)	**Train**
	↓		
English (real-mono)	⟶	German (**synthetic**)	**Synthesize**

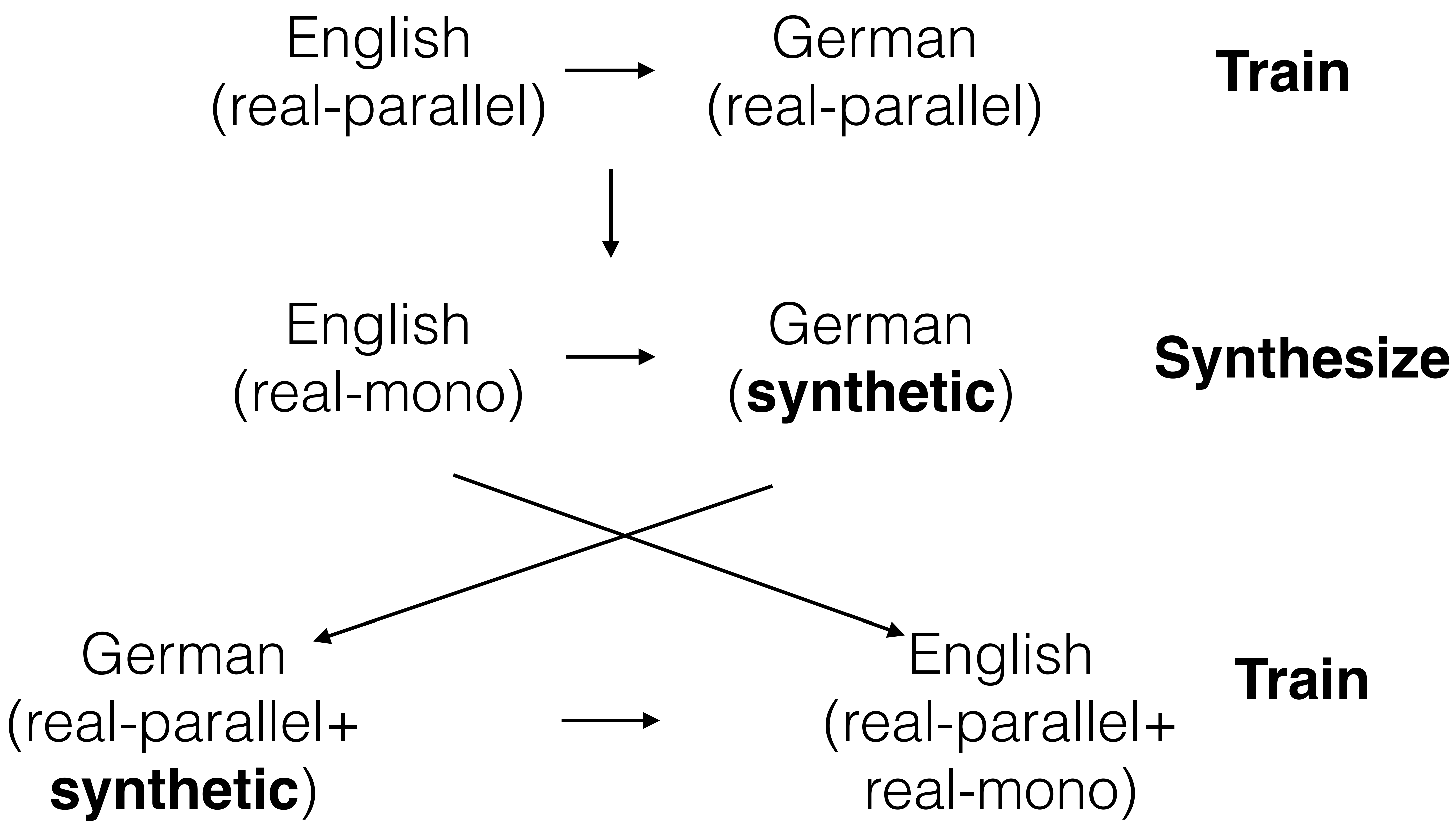
Back-Translation

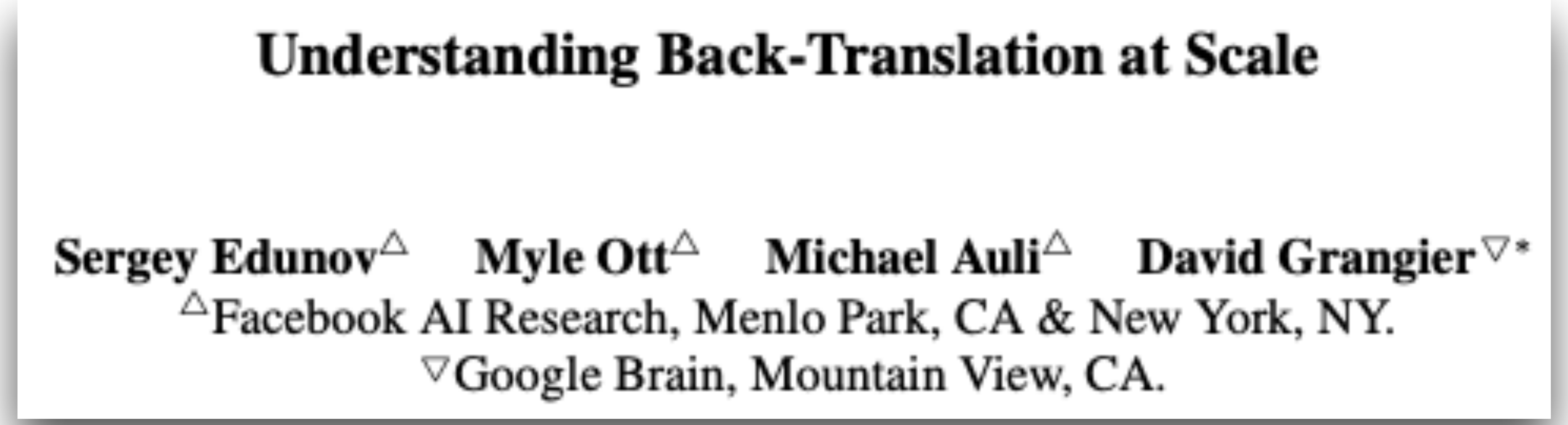
- Sennrich et al. 2016
- A simpler approach - **synthesize** parallel data from monolingual data:
  - Train a “reverse” model with the available parallel data
  - Translate the monolingual data using the reverse model
  - Train a model using the “real” parallel data and the synthetic parallel data



Back-Translation

- Sennrich et al. 2016
- A simpler approach - **synthesize** parallel data from monolingual data:
    - Train a “reverse” model with the available parallel data
    - Translate the monolingual data using the reverse model
    - Train a model using the “real” parallel data and the synthetic parallel data
- The driving force of todays state-of-the-art systems

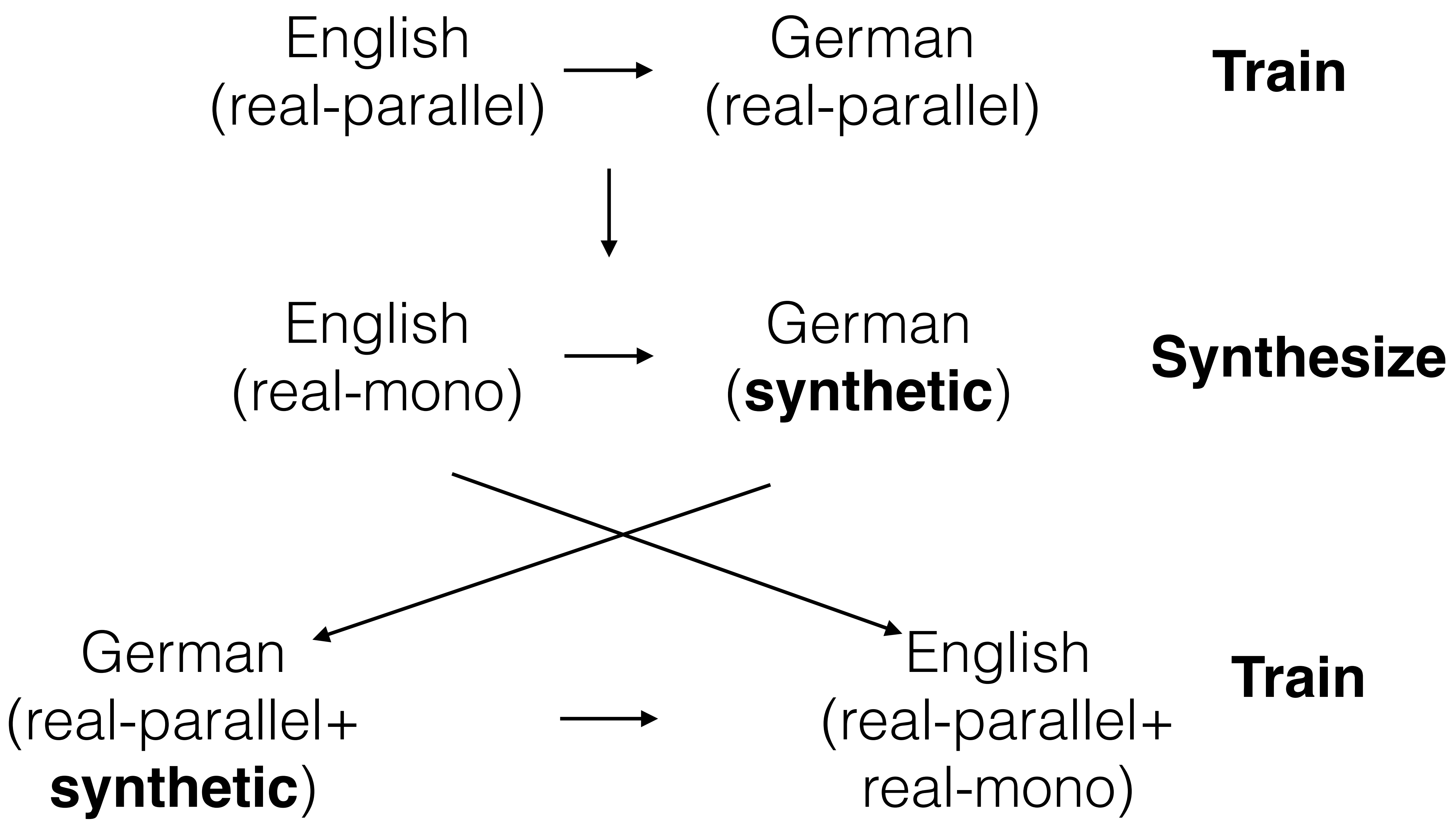


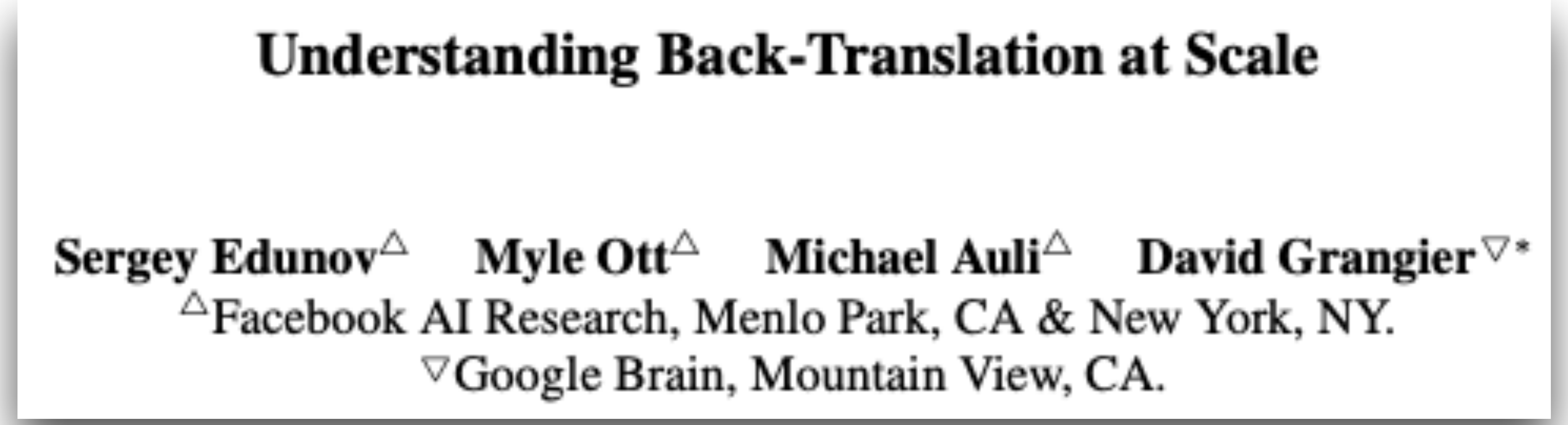

Understanding Back-Translation at Scale

Sergey Edunov[△] **Myle Ott**[△] **Michael Auli**[△] **David Grangier**[▽*]
[△]Facebook AI Research, Menlo Park, CA & New York, NY.
[▽]Google Brain, Mountain View, CA.

Back-Translation

- <u>Sennrich et al. 2016</u>
- A simpler approach - **synthesize** parallel data from monolingual data:
  - Train a “reverse” model with the available parallel data
  - Translate the monolingual data using the reverse model
  - Train a model using the “real” parallel data and the synthetic parallel data
- The driving force of todays state-of-the-art systems
  - To “fix” the noise of synthetic data, usually followed by fine-tuning on “clean” data



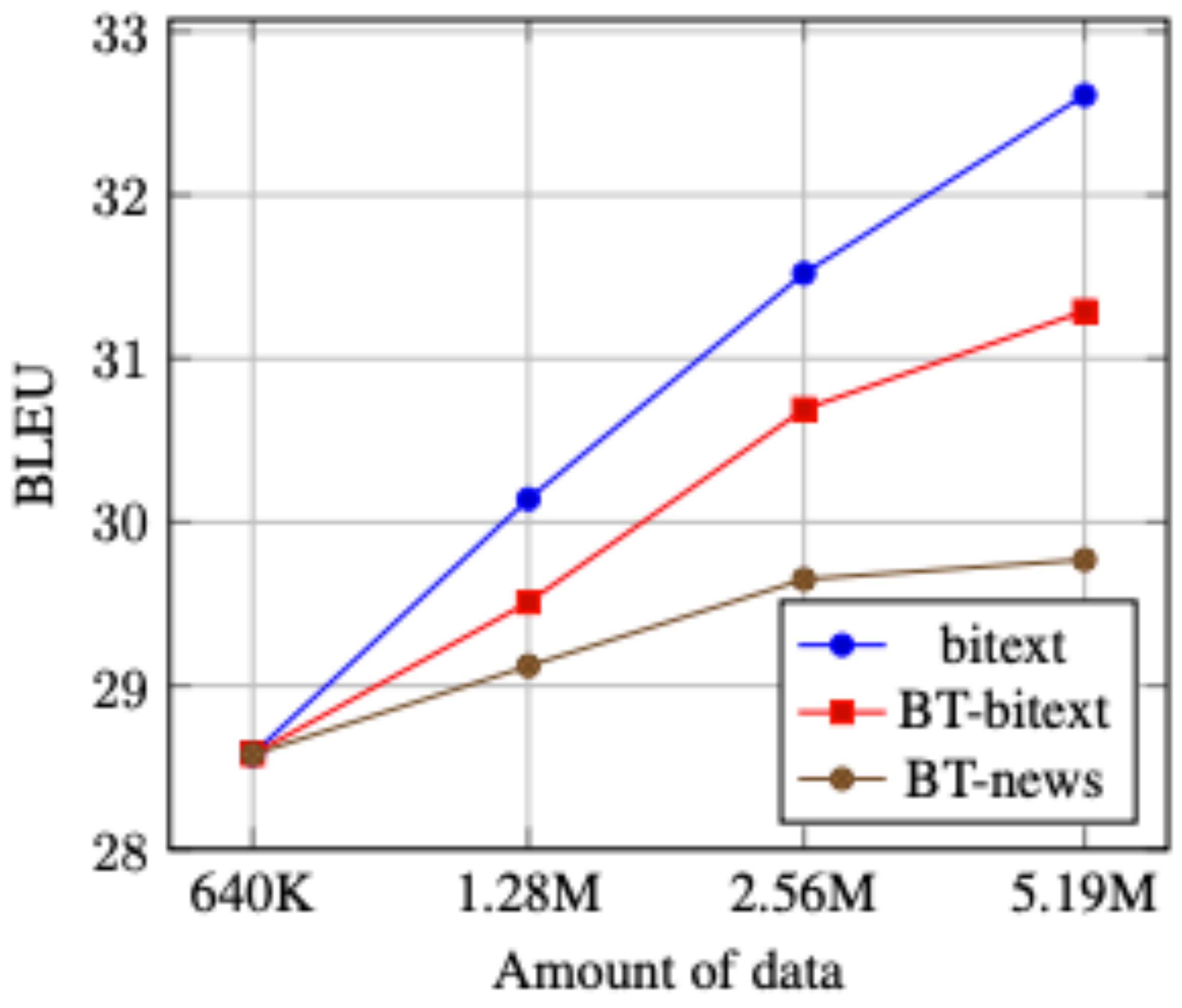

Understanding Back-Translation at Scale

Sergey Edunov[△] **Myle Ott**[△] **Michael Auli**[△] **David Grangier**[▽*]
[△]Facebook AI Research, Menlo Park, CA & New York, NY.
[▽]Google Brain, Mountain View, CA.

Transfer Learning

Transfer Learning

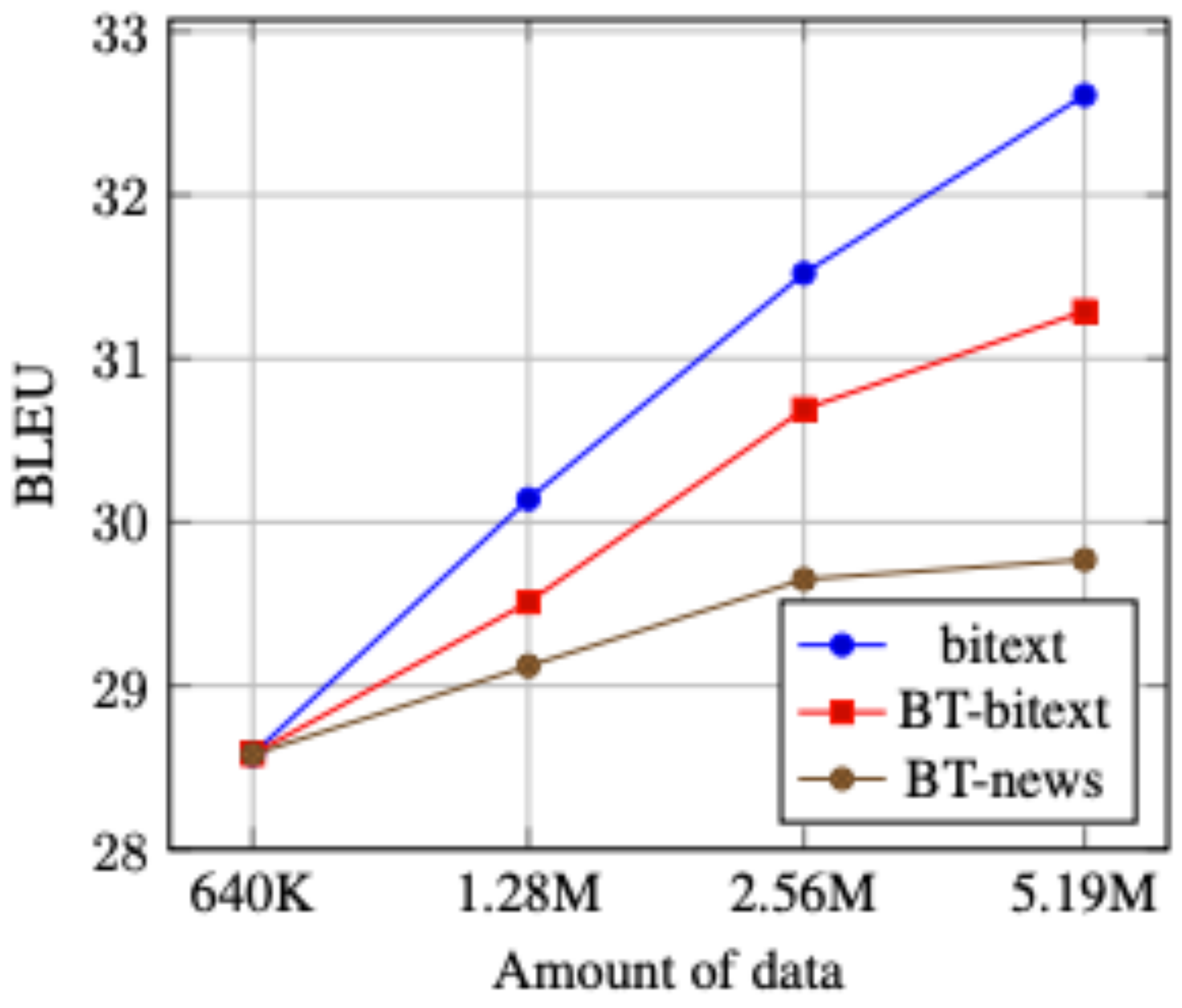
- Back-Translation gives nice improvements, but monolingual data is not as good as parallel data



<u>From Edunov et al 2018</u>

Transfer Learning

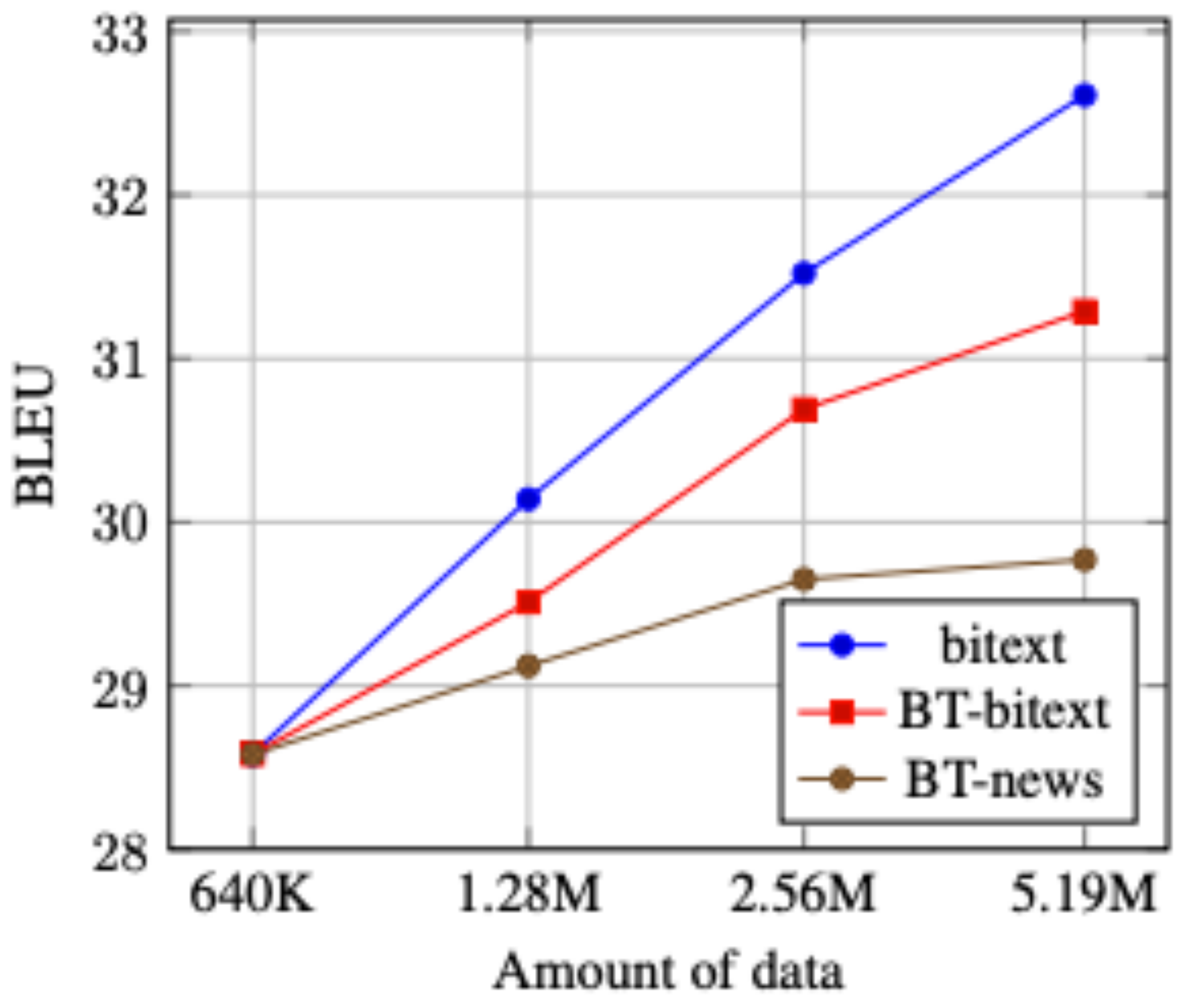
- Back-Translation gives nice improvements, but monolingual data is not as good as parallel data
- Can we use parallel data from other language pairs?



From Edunov et al 2018

Transfer Learning

- Back-Translation gives nice improvements, but monolingual data is not as good as parallel data
- Can we use parallel data from other language pairs?
- "Transfer Learning for Low-Resource Neural Machine Translation", Zoph et al. (2016)



From Edunov et al 2018

Transfer Learning

- Back-Translation gives nice improvements, but monolingual data is not as good as parallel data
- Can we use parallel data from other language pairs?
- "Transfer Learning for Low-Resource Neural Machine Translation", Zoph et al. (2016)
- Idea - Train a high-resource "parent" model (French-English) and **fine-tune** it for a low-resource "child" pair (Uzbek-English)

Language Pair	Parent	Train Size	BLEU ↑	PPL ↓
Uzbek–English	None	1.8m	10.7	22.4
	French–English	1.8m	**15.0 (+4.3)**	**13.9**
French′–English	None	1.8m	13.3	28.2
	French–English	1.8m	**20.0 (+6.7)**	**10.9**

Transfer Learning

- Back-Translation gives nice improvements, but monolingual data is not as good as parallel data
- Can we use parallel data from other language pairs?
- "Transfer Learning for Low-Resource Neural Machine Translation", Zoph et al. (2016)
- Idea - Train a high-resource "parent" model (French-English) and **fine-tune** it for a low-resource "child" pair (Uzbek-English)
- **Freezing** some parts of the network helps - avoids **"catastrophic forgetting"**

Language Pair	Parent	Train Size	BLEU ↑	PPL ↓
Uzbek–English	None	1.8m	10.7	22.4
	French–English	1.8m	**15.0 (+4.3)**	**13.9**
French′–English	None	1.8m	13.3	28.2
	French–English	1.8m	**20.0 (+6.7)**	**10.9**

Source Embeddings	Source RNN	Target RNN	Attention	Target Input Embeddings	Target Output Embeddings	Dev BLEU ↑	Dev PPL ↓
🔒	🔒	🔒	🔒	🔒	🔒	0.0	112.6
🔓	🔒	🔒	🔒	🔒	🔒	7.7	24.7
🔓	🔓	🔒	🔒	🔒	🔒	11.8	17.0
🔓	🔓	🔓	🔒	🔒	🔒	14.2	14.5
🔓	🔓	🔓	🔓	🔒	🔒	**15.0**	13.9
🔓	🔓	🔓	🔓	🔓	🔒	14.7	**13.8**
🔓	🔓	🔓	🔓	🔓	🔓	13.7	14.4

Table 7: Starting with the parent French–English model (BLEU =24.4, PPL=6.2), we randomly assign Uzbek word types to French word embeddings, freeze various parameters of the neural network model (🔒), and allow Uzbek–English (child model) training to modify other parts (🔓). The table shows how Uzbek–English BLEU and perplexity vary as we allow more parameters to be re-trained.

Multilingual NMT

Multilingual NMT

- If transfer learning helps, let’s train multiple language pairs **together**

Multilingual NMT

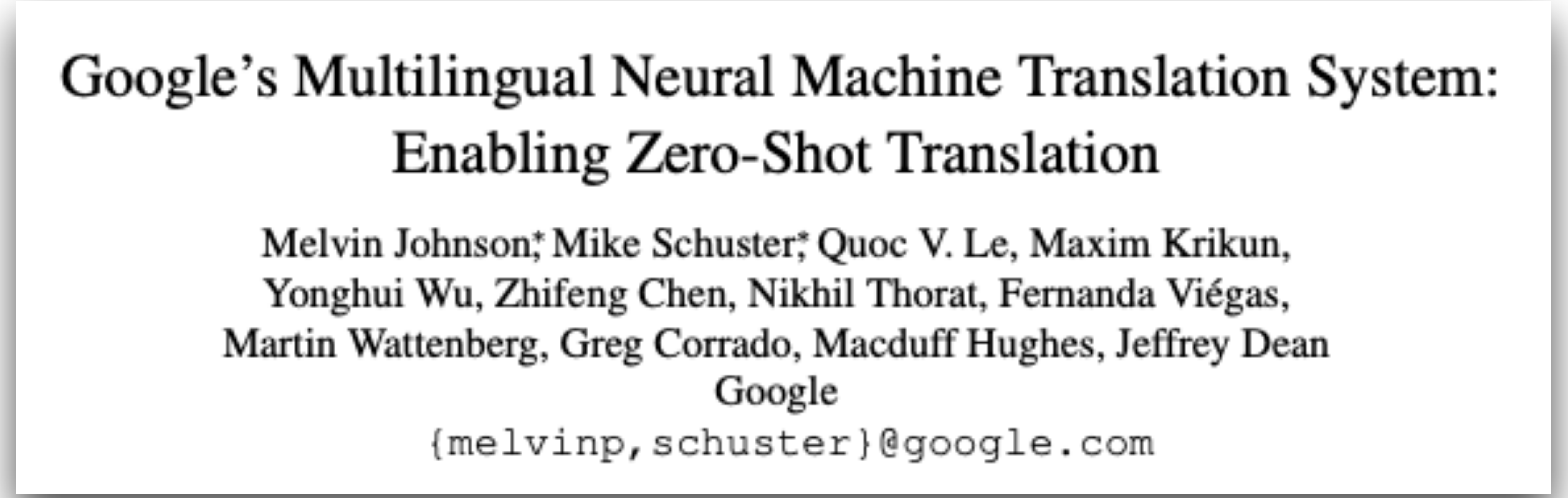
- If transfer learning helps, let's train multiple language pairs **together**
- Encourage more knowledge transfer between language pairs - better quality

Multilingual NMT

- If transfer learning helps, let's train multiple language pairs **together**
- Encourage more knowledge transfer between language pairs - better quality
- Reduce hardware requirements - one model, one server, many language pairs

Multilingual NMT

- If transfer learning helps, let’s train multiple language pairs **together**
- Encourage more knowledge transfer between language pairs - better quality
- Reduce hardware requirements - one model, one server, many language pairs



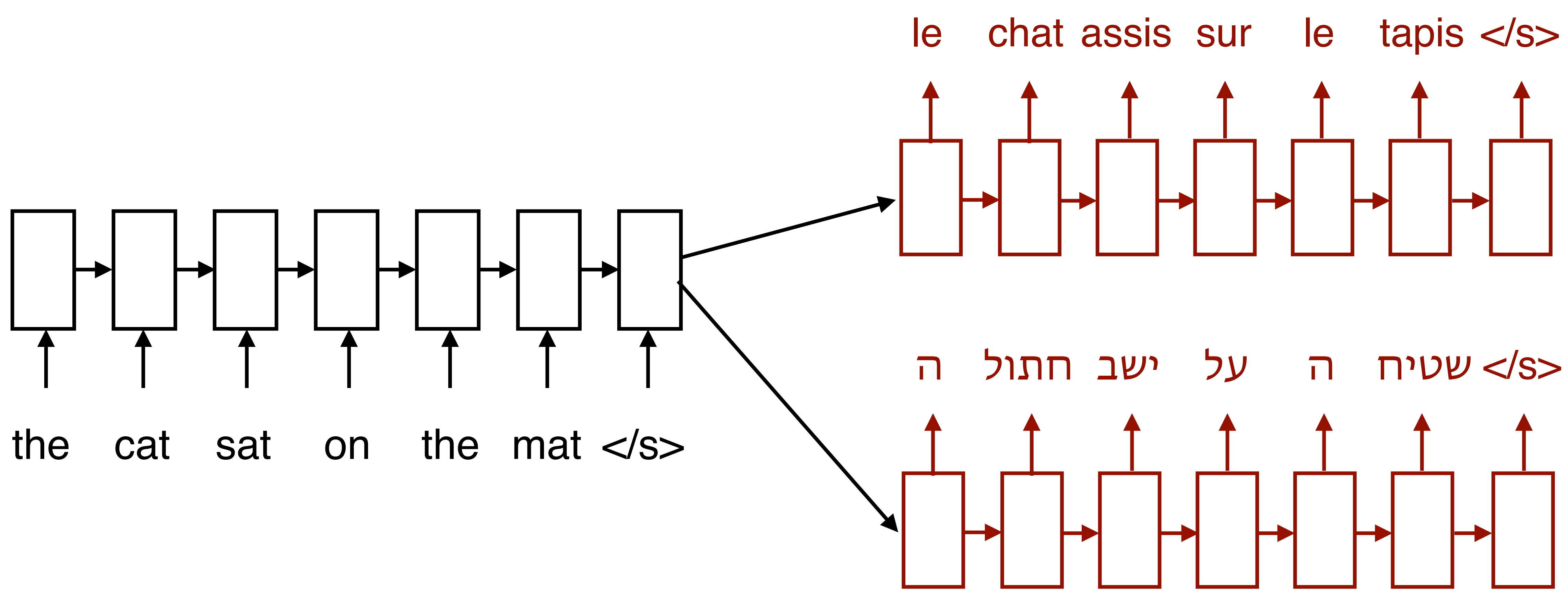
Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation

Melvin Johnson,* Mike Schuster,* Quoc V. Le, Maxim Krikun,
Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas,
Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean
Google
{melvinp,schuster}@google.com

Multilingual NMT

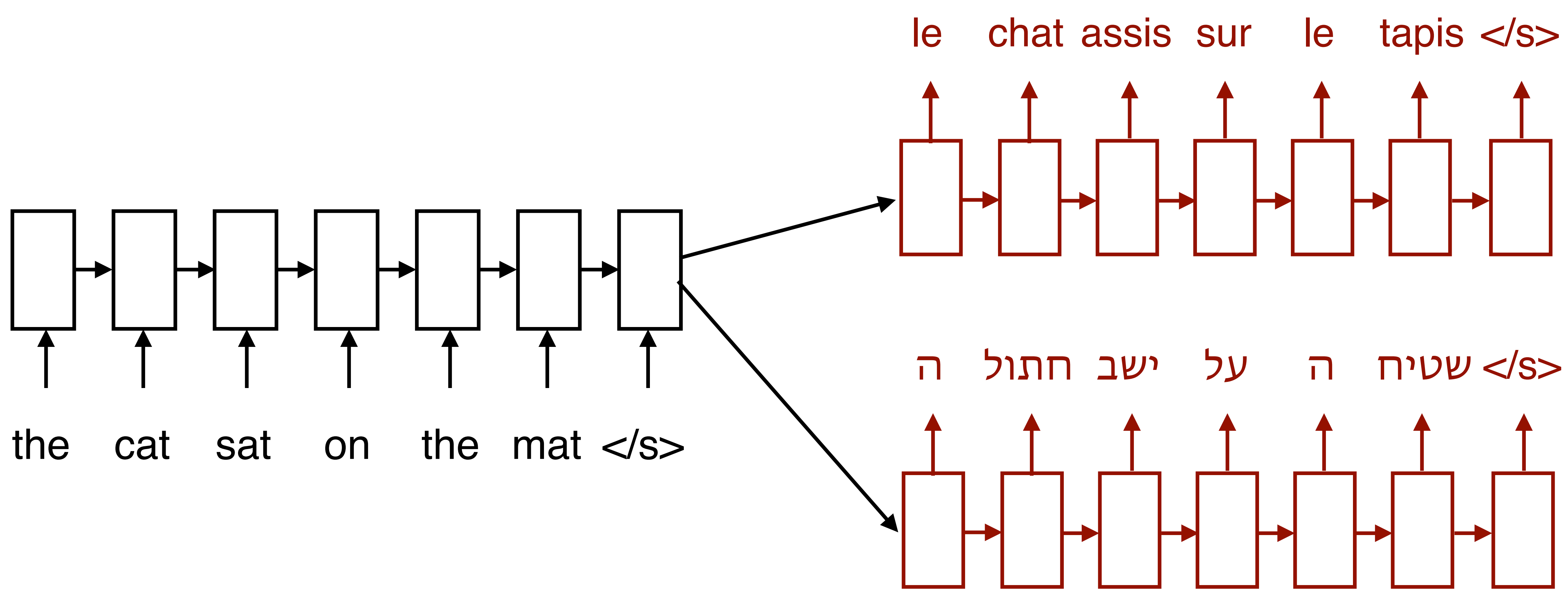
Multilingual NMT

- One approach: separate encoder/decoder per language (Dong et al. 2015, Firat et al. 2016)



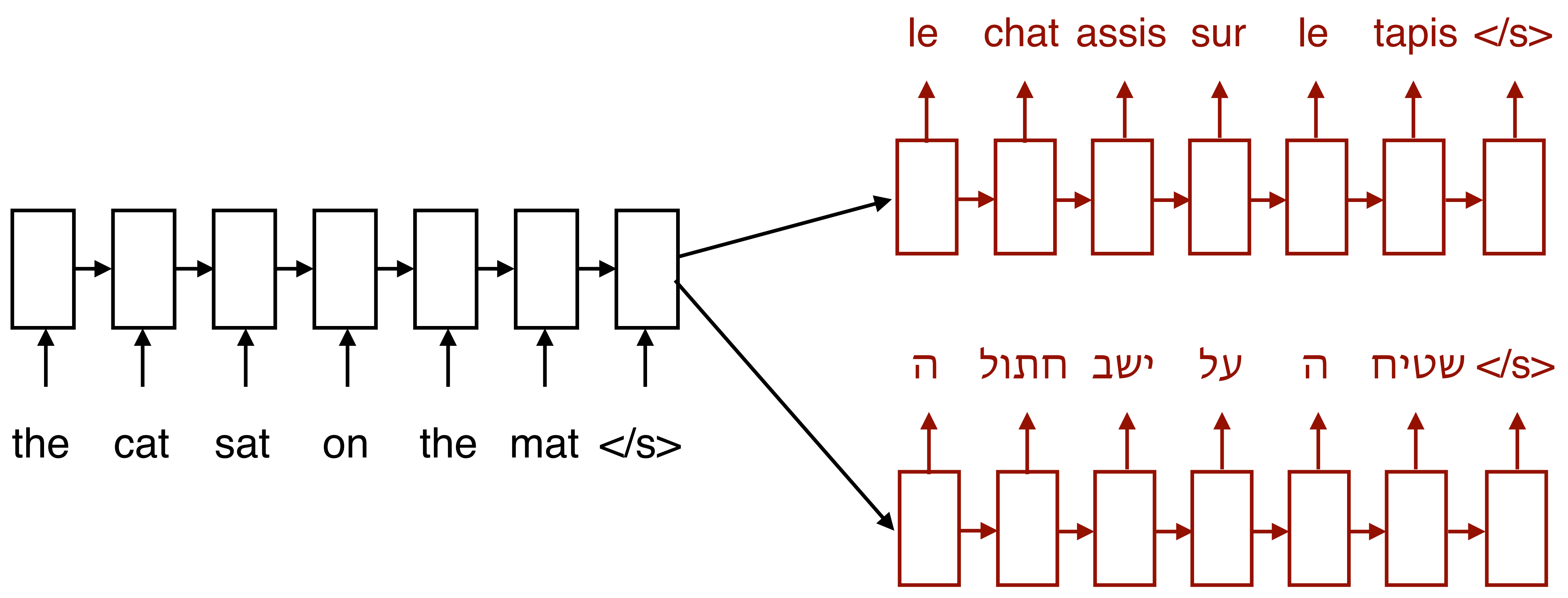
Multilingual NMT

- One approach: separate encoder/decoder per language ([Dong et al. 2015](), [Firat et al. 2016]())
    - Pros - each language has its own parameters, no interference



Multilingual NMT

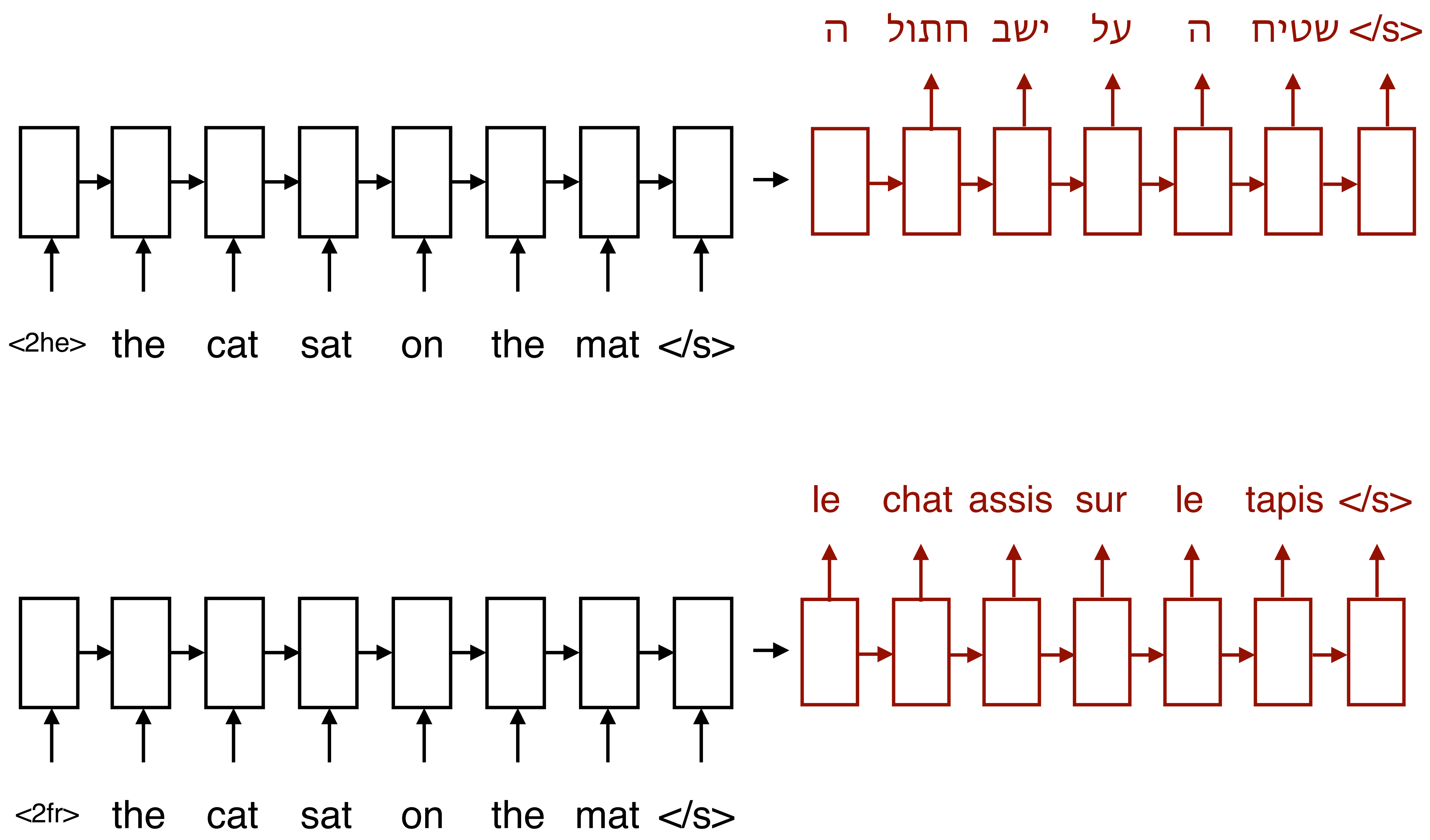
- One approach: separate encoder/decoder per language (Dong et al. 2015, Firat et al. 2016)
  - Pros - each language has its own parameters, no interference
  - Cons - complex architecture, less parameter sharing for transfer



Multilingual NMT

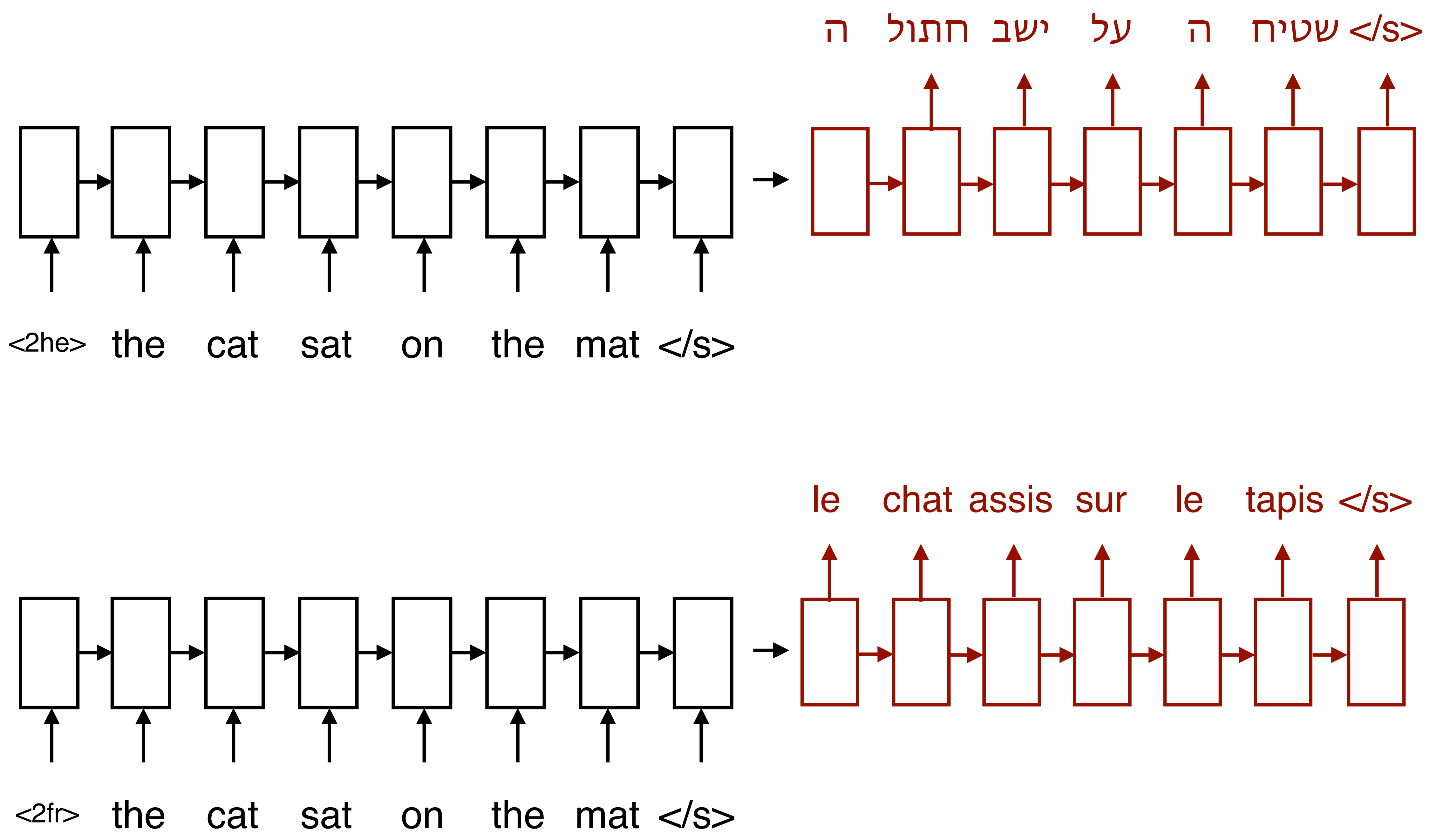
Multilingual NMT

- Another approach - **share all parameters** (Johnson et al. 2016, Ha et al. 2016)



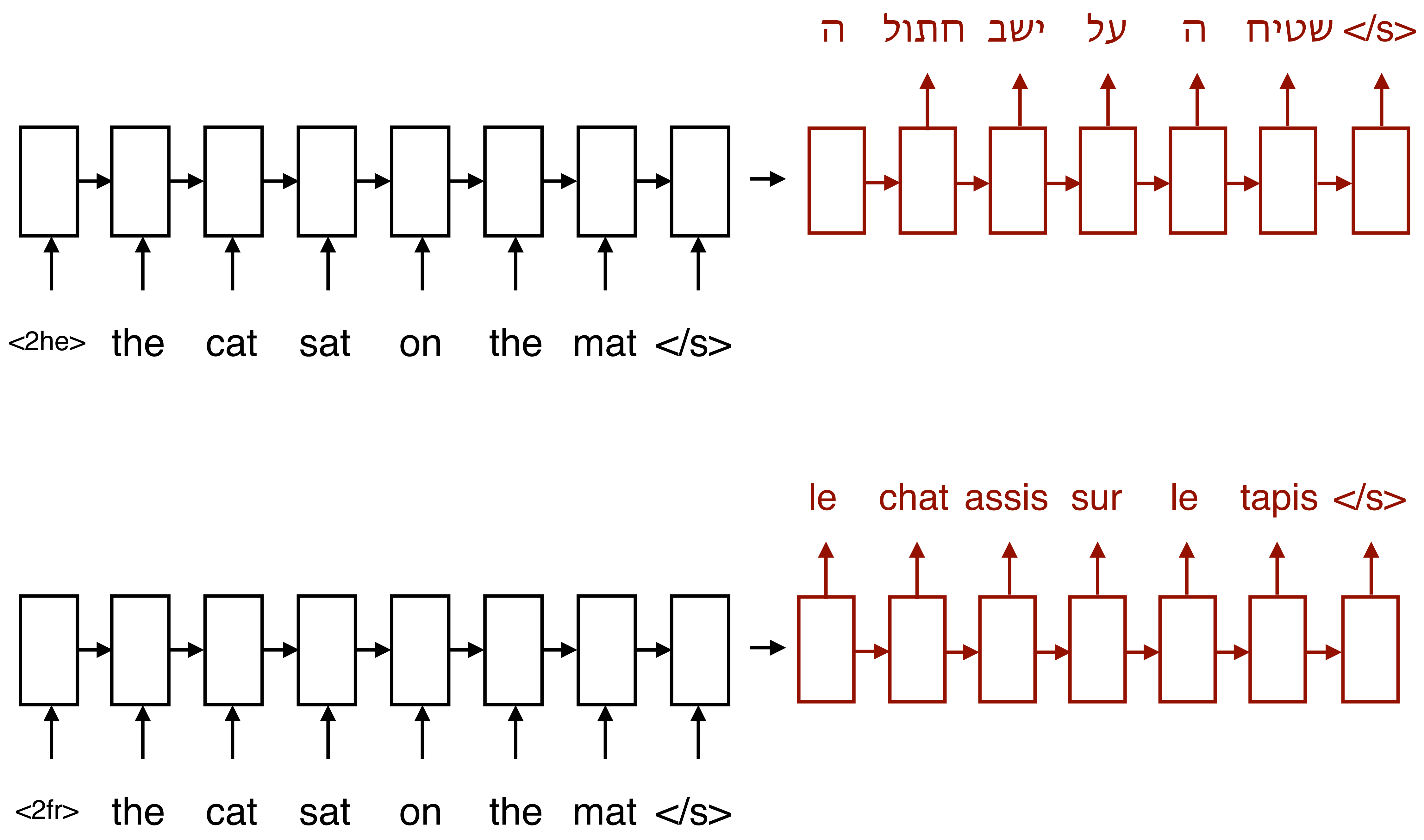
Multilingual NMT

- Another approach - **share all parameters** (Johnson et al. 2016, Ha et al. 2016)
    - Use a special language token to control the target language



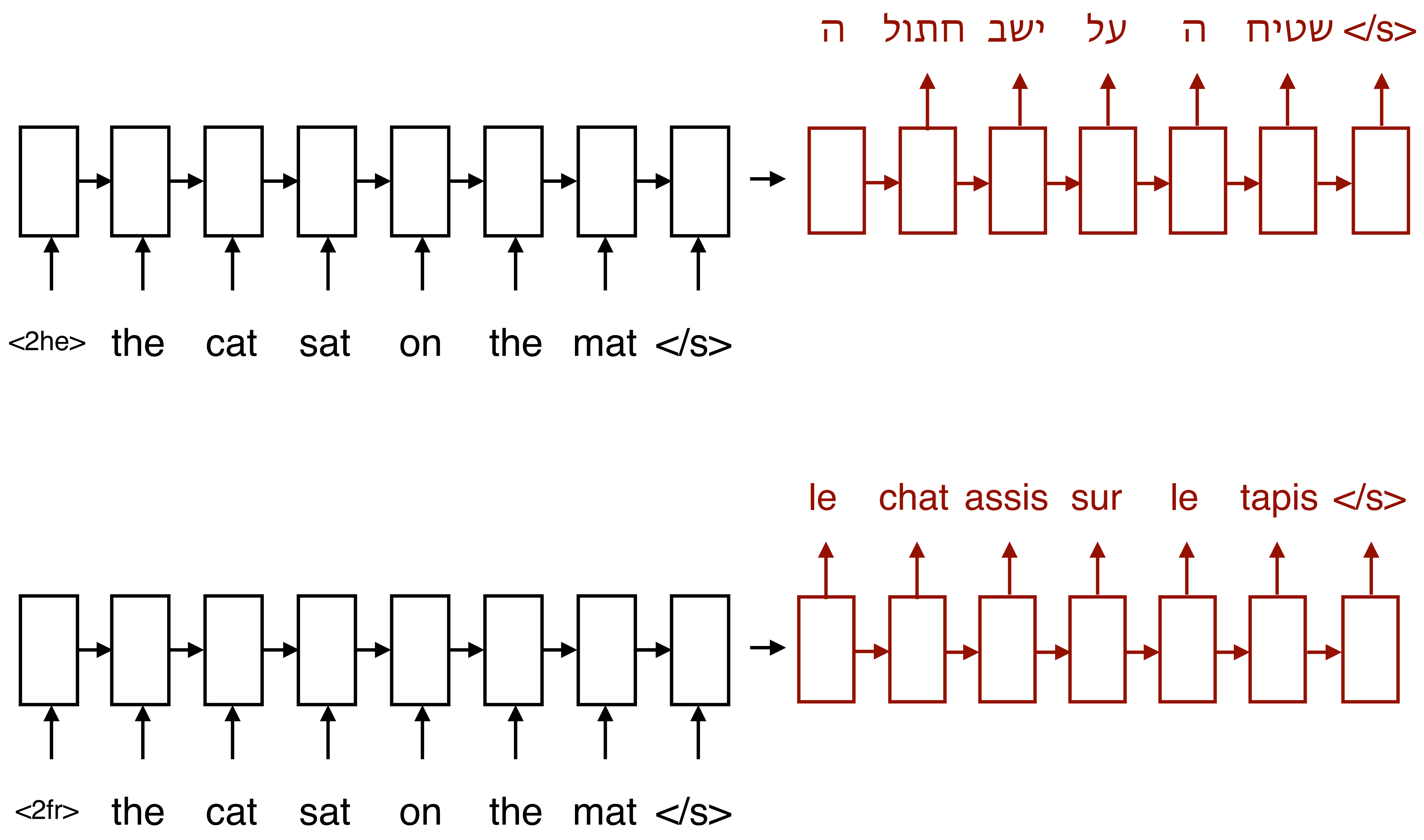
Multilingual NMT

- Another approach - **share all parameters** (Johnson et al. 2016, Ha et al. 2016)
  - Use a special language token to control the target language
  - Pros - Full parameter sharing, no architecture changes



Multilingual NMT

- Another approach - **share all parameters** (Johnson et al. 2016, Ha et al. 2016)
  - Use a special language token to control the target language
  - Pros - Full parameter sharing, no architecture changes
  - Con - languages may “interfere” each other



Multilingual NMT

Multilingual NMT

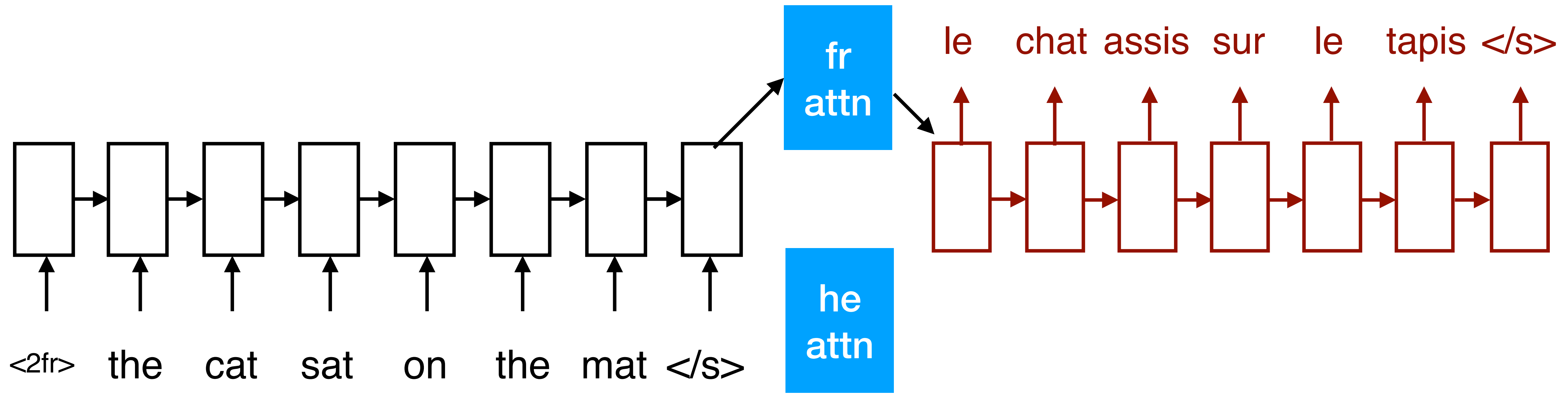
- A third approach - “in between”

Multilingual NMT

- A third approach - “in between”
- Share only some of the parameters

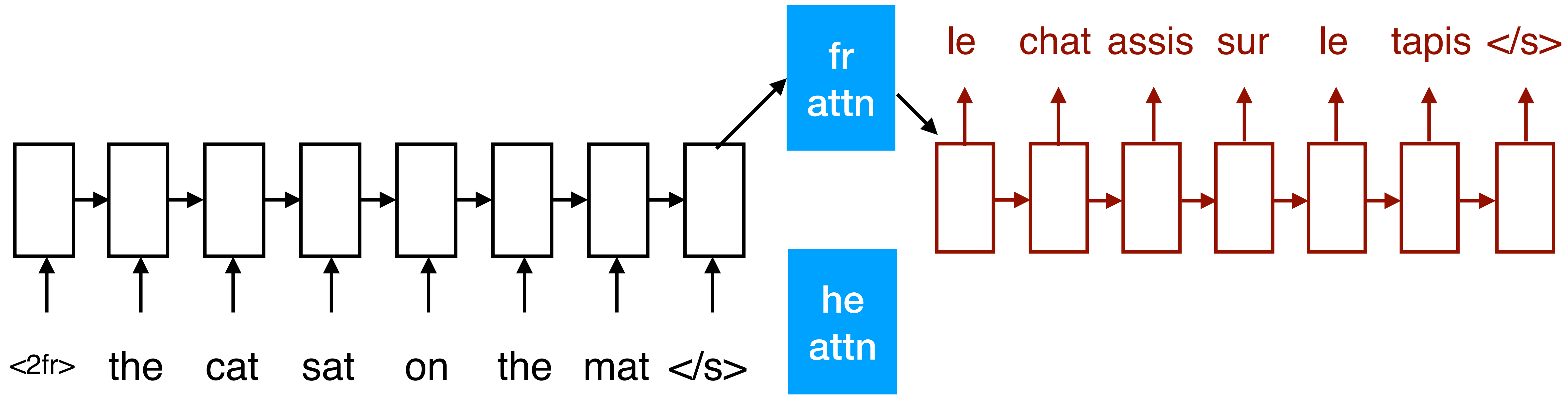
Multilingual NMT

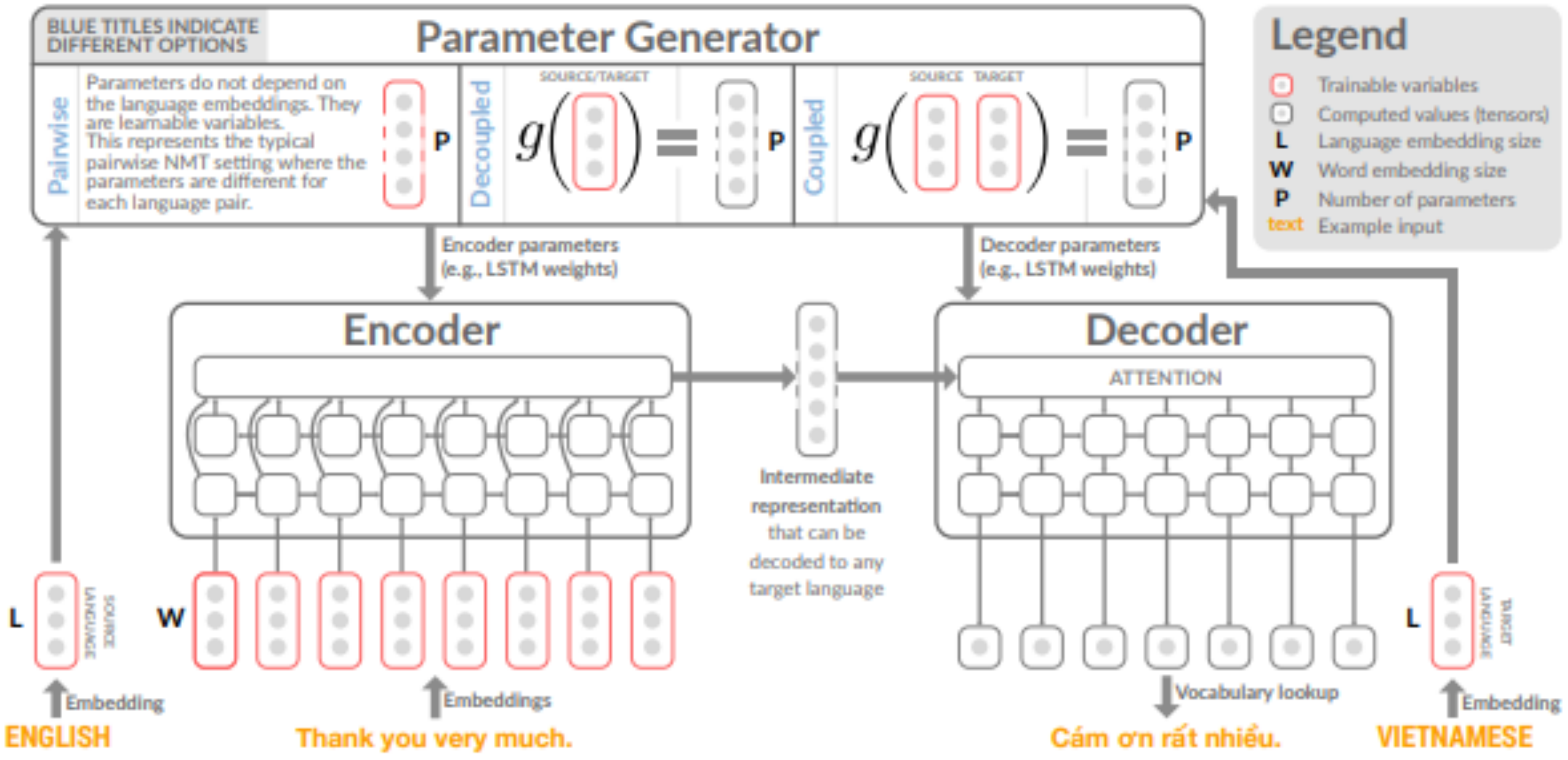
- A third approach - “in between”
- Share only some of the parameters
    - Blackwood et al (2018) - all but the attention



Multilingual NMT

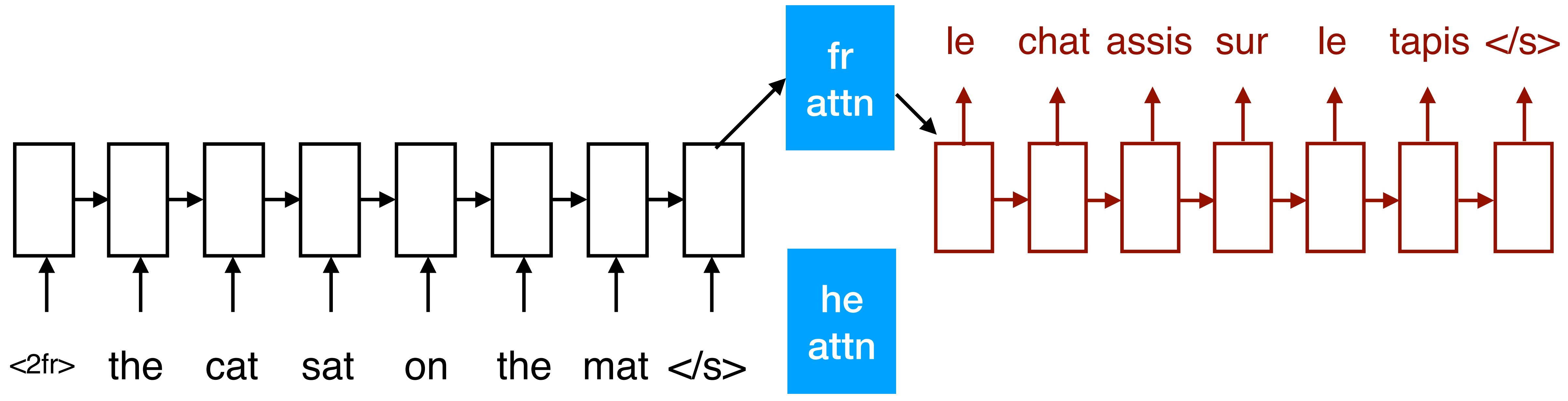
- A third approach - "in between"
- Share only some of the parameters
  - Blackwood et al (2018) - all but the attention
  - Platanios et al (2018) - learn what to share

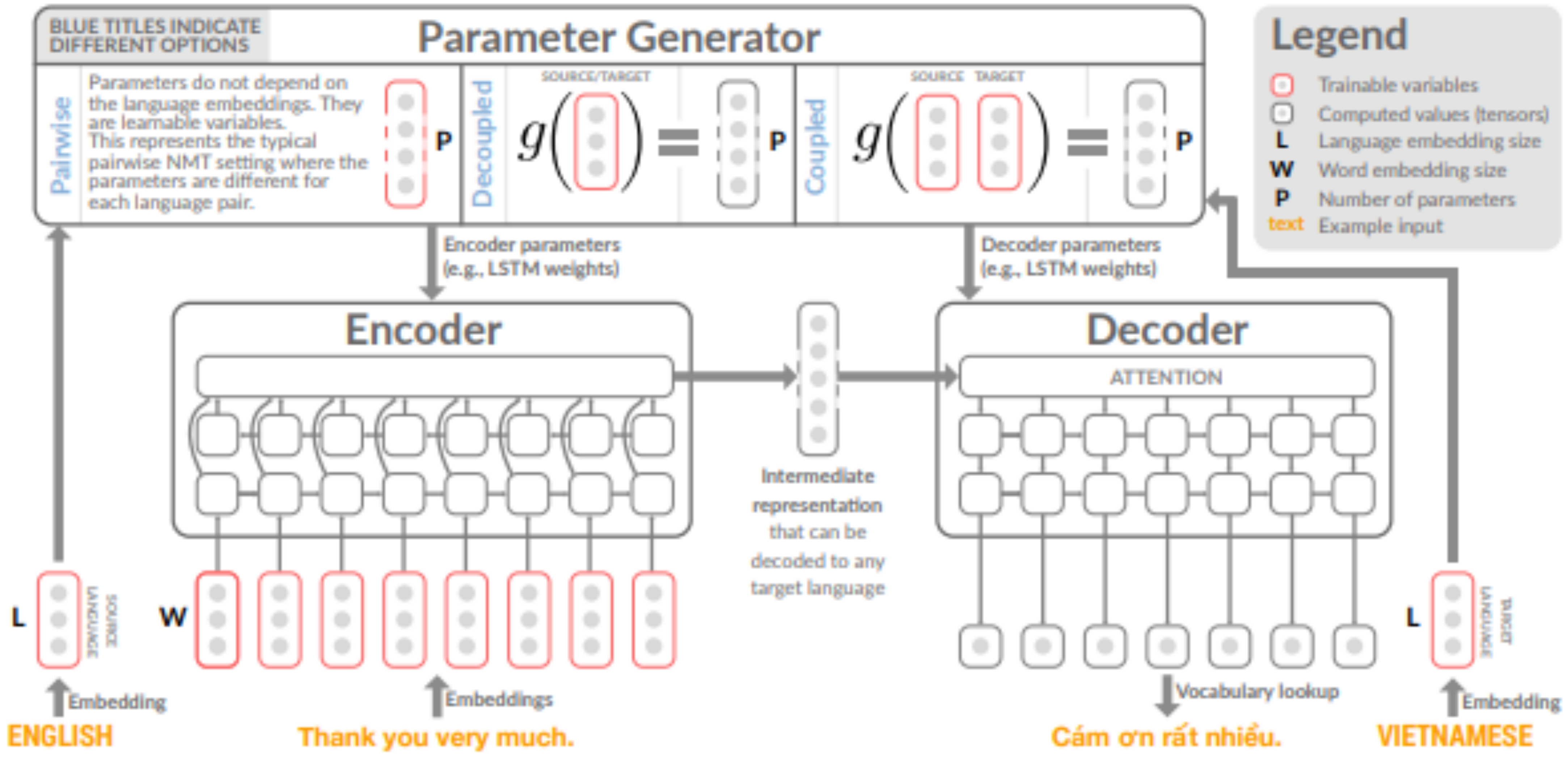




Multilingual NMT

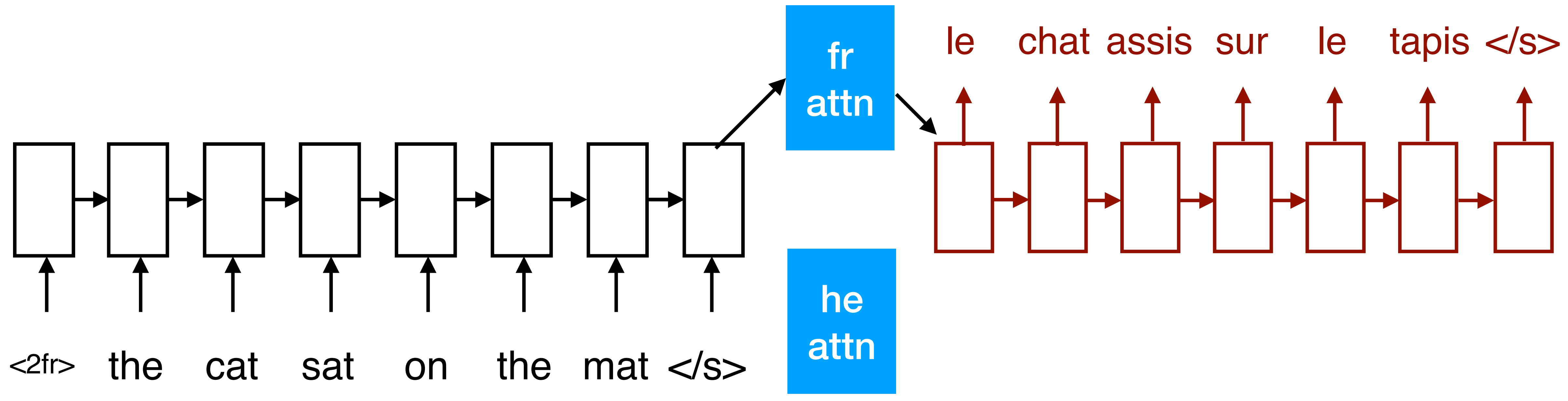
- A third approach - “in between”
- Share only some of the parameters
  - Blackwood et al (2018) - all but the attention
  - Platanios et al (2018) - learn what to share
- Can reduce interference

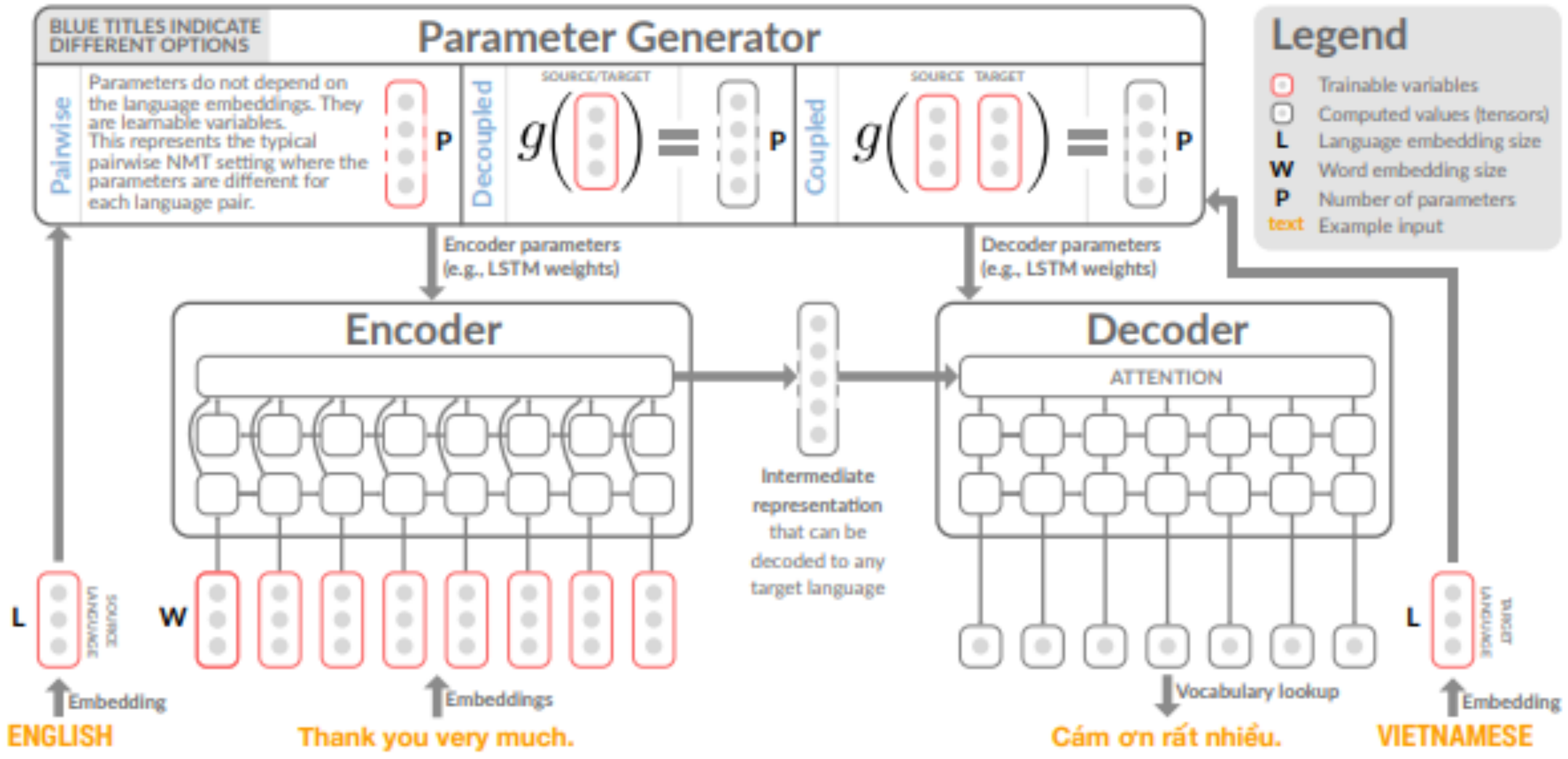




Multilingual NMT

- A third approach - “in between”
- Share only some of the parameters
  - Blackwood et al (2018) - all but the attention
  - Platanios et al (2018) - learn what to share
- Can reduce interference
- More complex models





Multilingual NMT - Data Settings

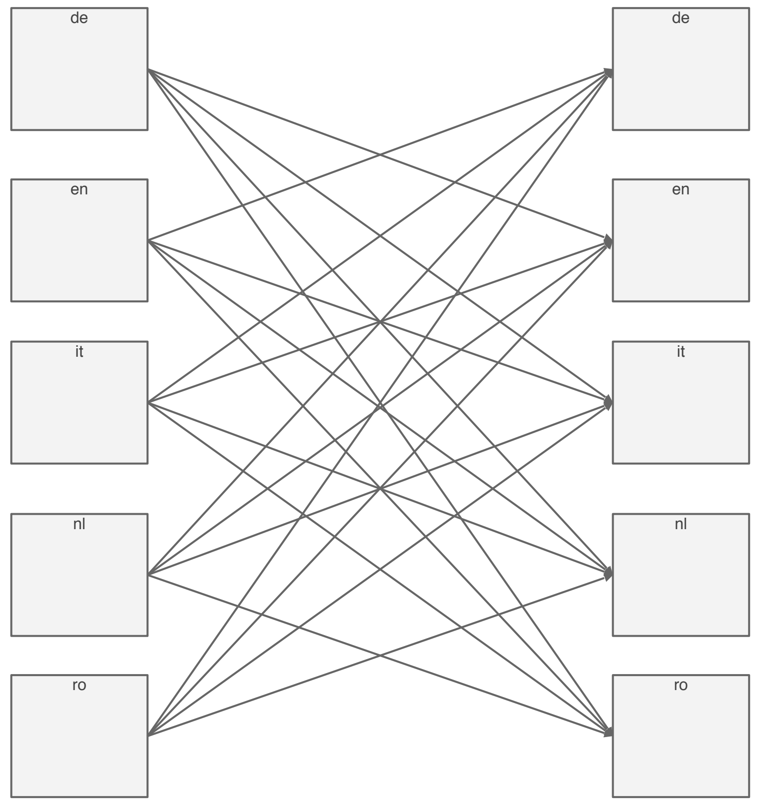
Multilingual NMT - Data Settings

Many-to-Many

Multilingual NMT - Data Settings

Many-to-Many

Fully-Supervised

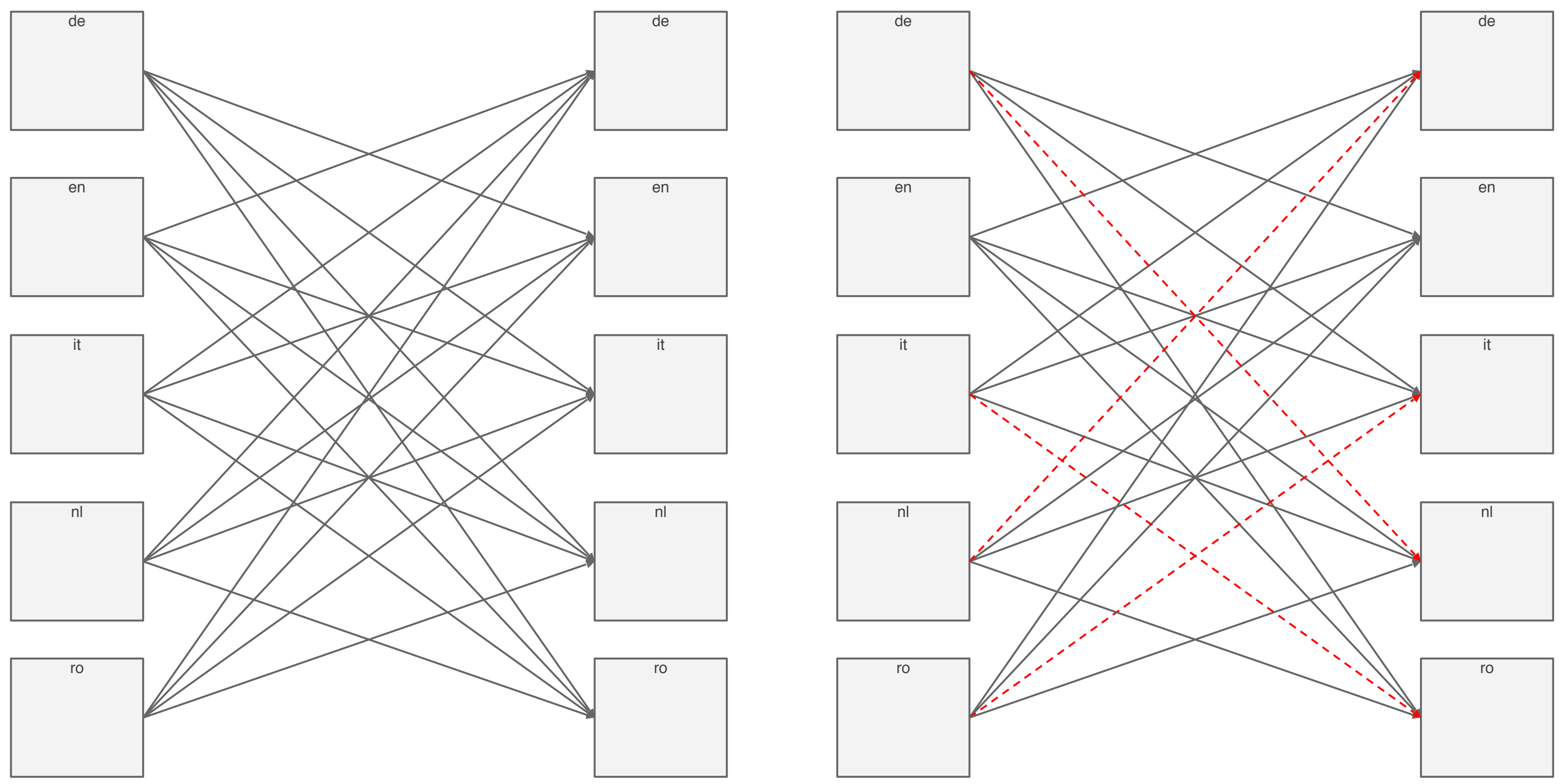


Multilingual NMT - Data Settings

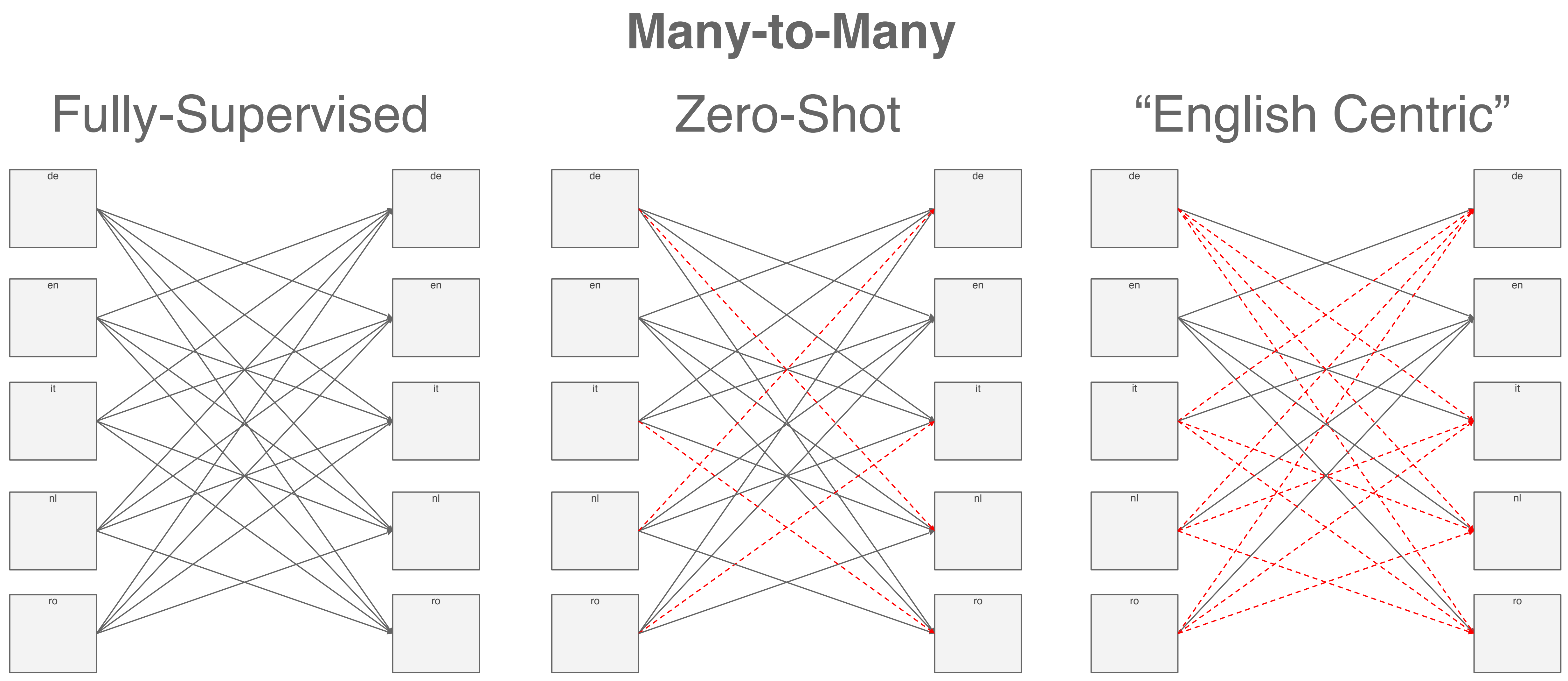
Many-to-Many

Fully-Supervised

Zero-Shot



Multilingual NMT - Data Settings



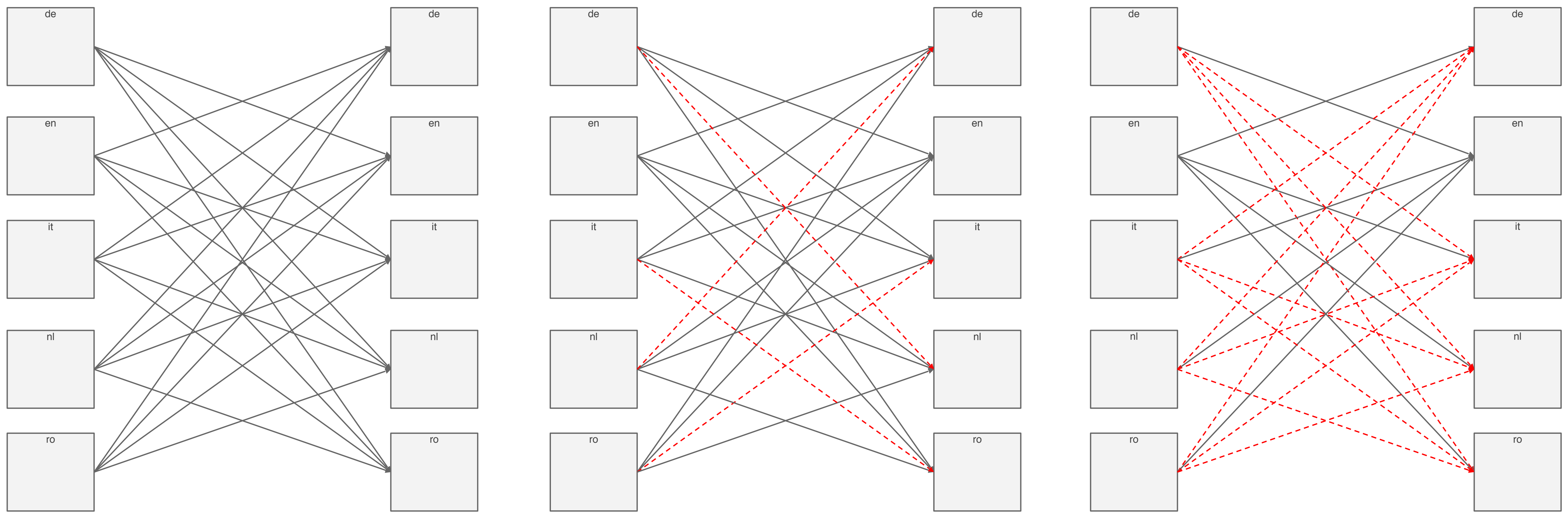
Multilingual NMT - Data Settings

Many-to-Many

Fully-Supervised

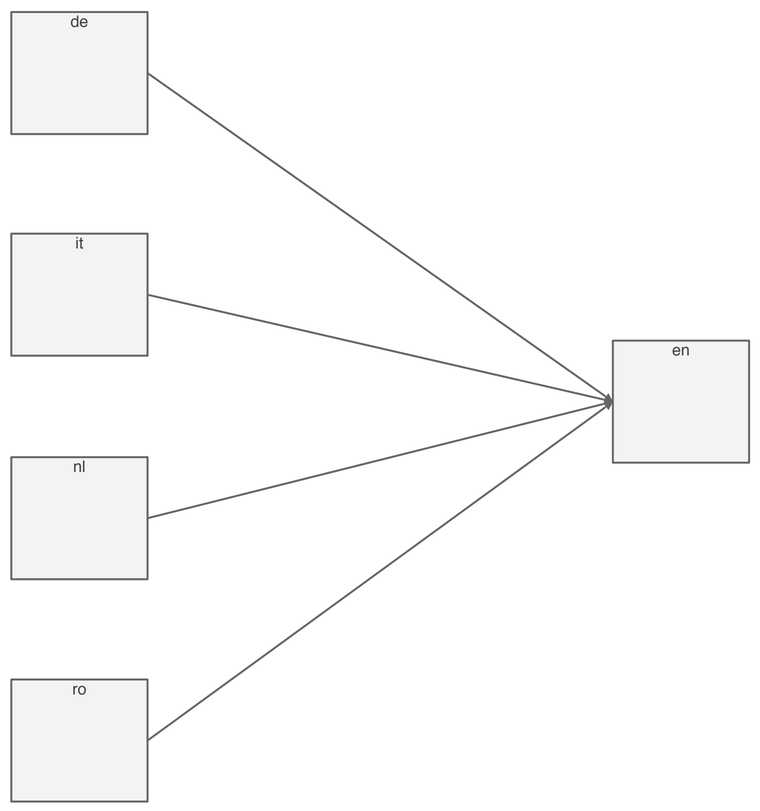
Zero-Shot

“English Centric”



Many-to-One

all-en



Multilingual NMT - Data Settings

Many-to-Many

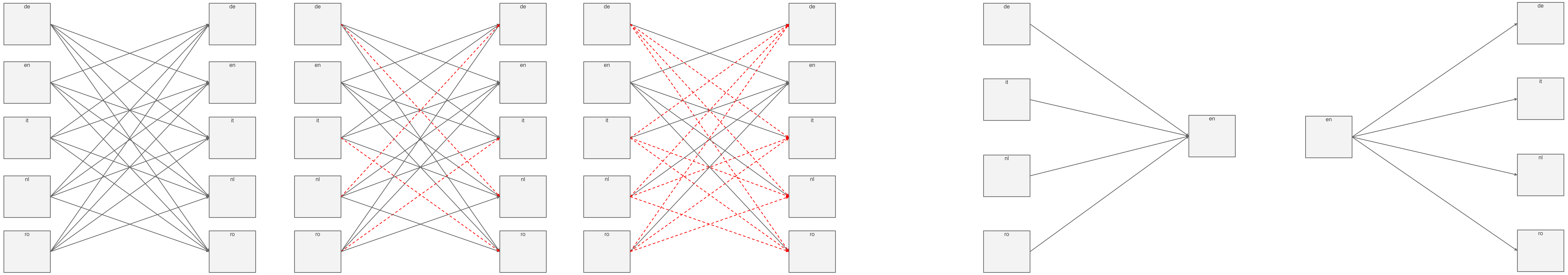
Fully-Supervised | Zero-Shot | "English Centric"

Many-to-One

all-en

One-to-Many

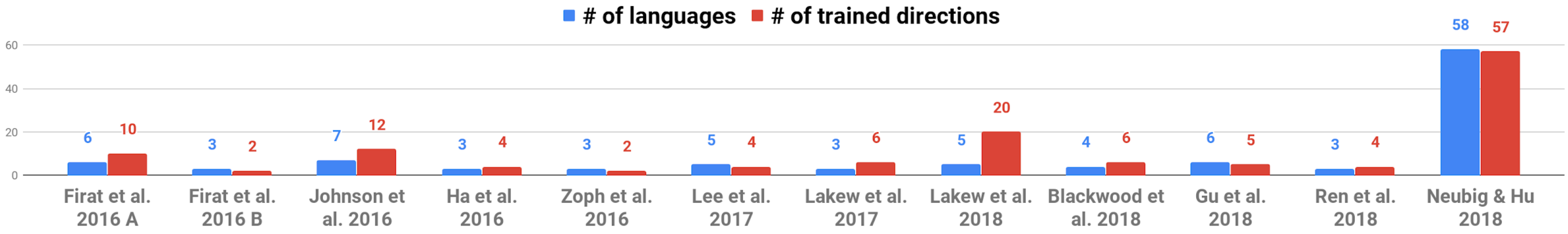
en-all



Massively Multilingual NMT

Massively Multilingual NMT

- Most works until 2018 - up to 5 languages, 20 translation directions (one outlier)



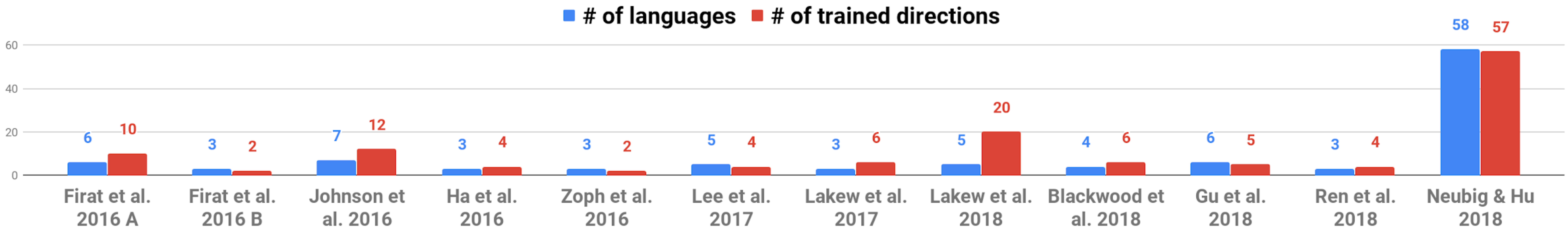
Massively Multilingual NMT

- Most works until 2018 - up to 5 languages, 20 translation directions (one outlier)
- Why stop here?

Massively Multilingual Neural Machine Translation

Roee Aharoni*
Bar Ilan University
Ramat-Gan
Israel
roee.aharoni@gmail.com

Melvin Johnson and **Orhan Firat**
Google AI
Mountain View
California
melvinp,orhanf@google.com



Massively Multilingual NMT

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset

TEDBlog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)

TEDBlog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols
  - Many-to-Many (English-Centric), Many-to-One, One-to-Many, One-to-One

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols
  - Many-to-Many (English-Centric), Many-to-One, One-to-Many, One-to-One
  - Joint Multilingual models

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com

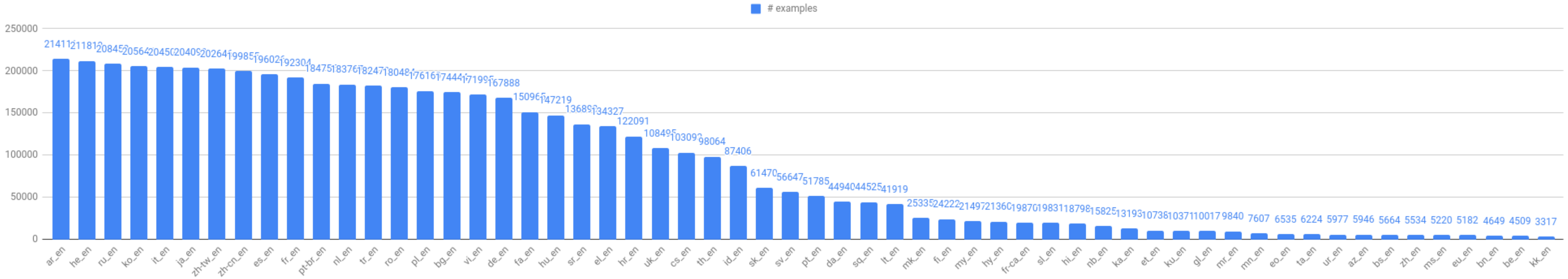
Massively Multilingual NMT

- Low resource experiments: The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols
  - Many-to-Many (English-Centric), Many-to-One, One-to-Many, One-to-One
  - Joint Multilingual models

TED Blog

Uncategorized

TED's Open Translation Project brings subtitles in 40+ languages to TED.com



Massively Multilingual NMT

Massively Multilingual NMT

- Multilingual models significantly outperform baselines

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

Massively Multilingual NMT

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

Massively Multilingual NMT

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models
- Similar result in language pairs with more data (baselines stronger here)

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

	Ar-En	De-En	He-En	It-En	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	27.84	30.5	**34.37**	33.64	31.59
many-to-one	25.93	28.87	30.19	32.42	29.35
many-to-many	**28.32**	**32.97**	33.18	**35.14**	**32.4**

Massively Multilingual NMT

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models
- Similar result in language pairs with more data (baselines stronger here)
- Why? many-to-many is "harder" 🤔

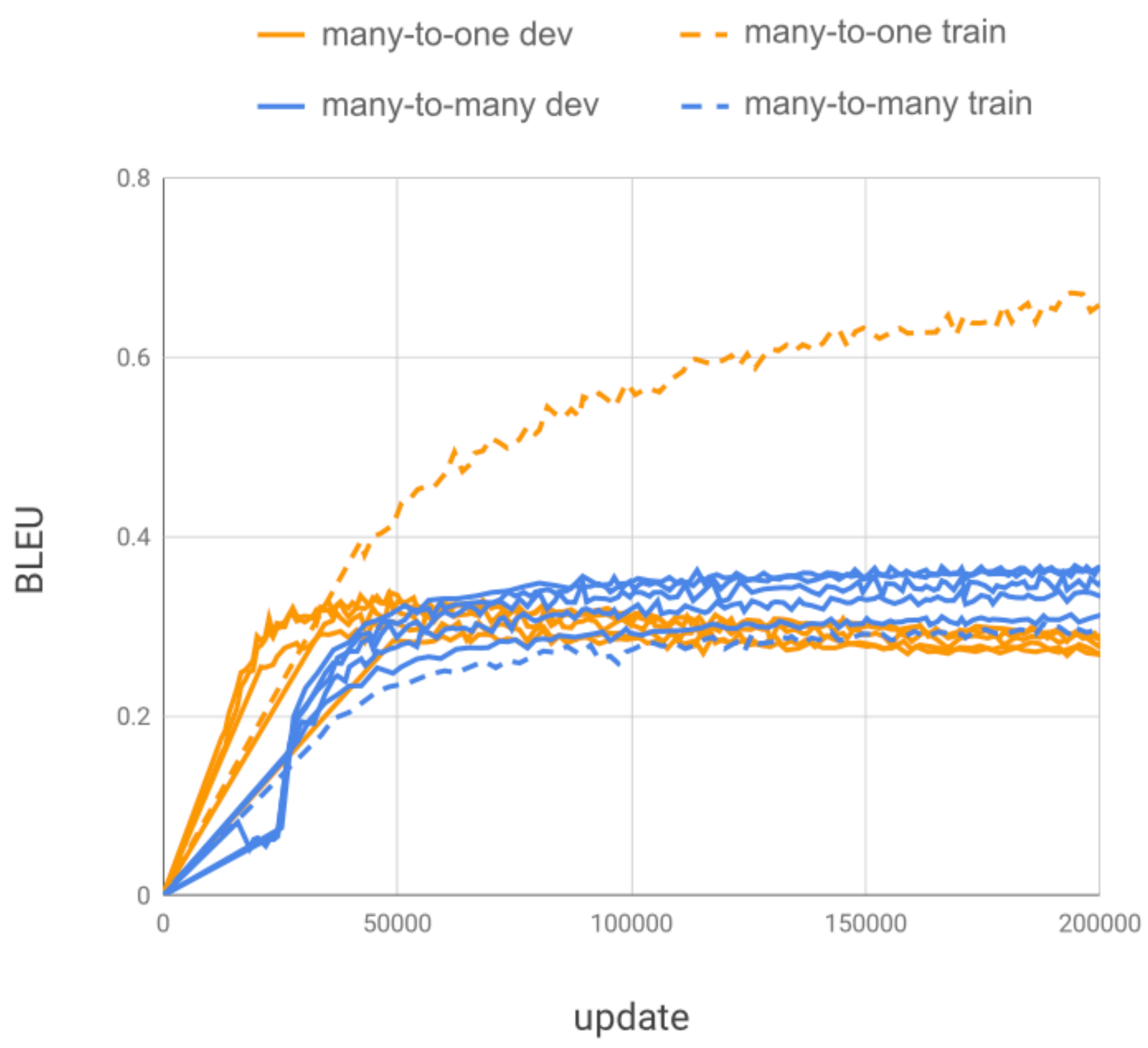
	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

	Ar-En	De-En	He-En	It-En	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	27.84	30.5	**34.37**	33.64	31.59
many-to-one	25.93	28.87	30.19	32.42	29.35
many-to-many	**28.32**	**32.97**	33.18	**35.14**	**32.4**

Multilinguality as Regularization

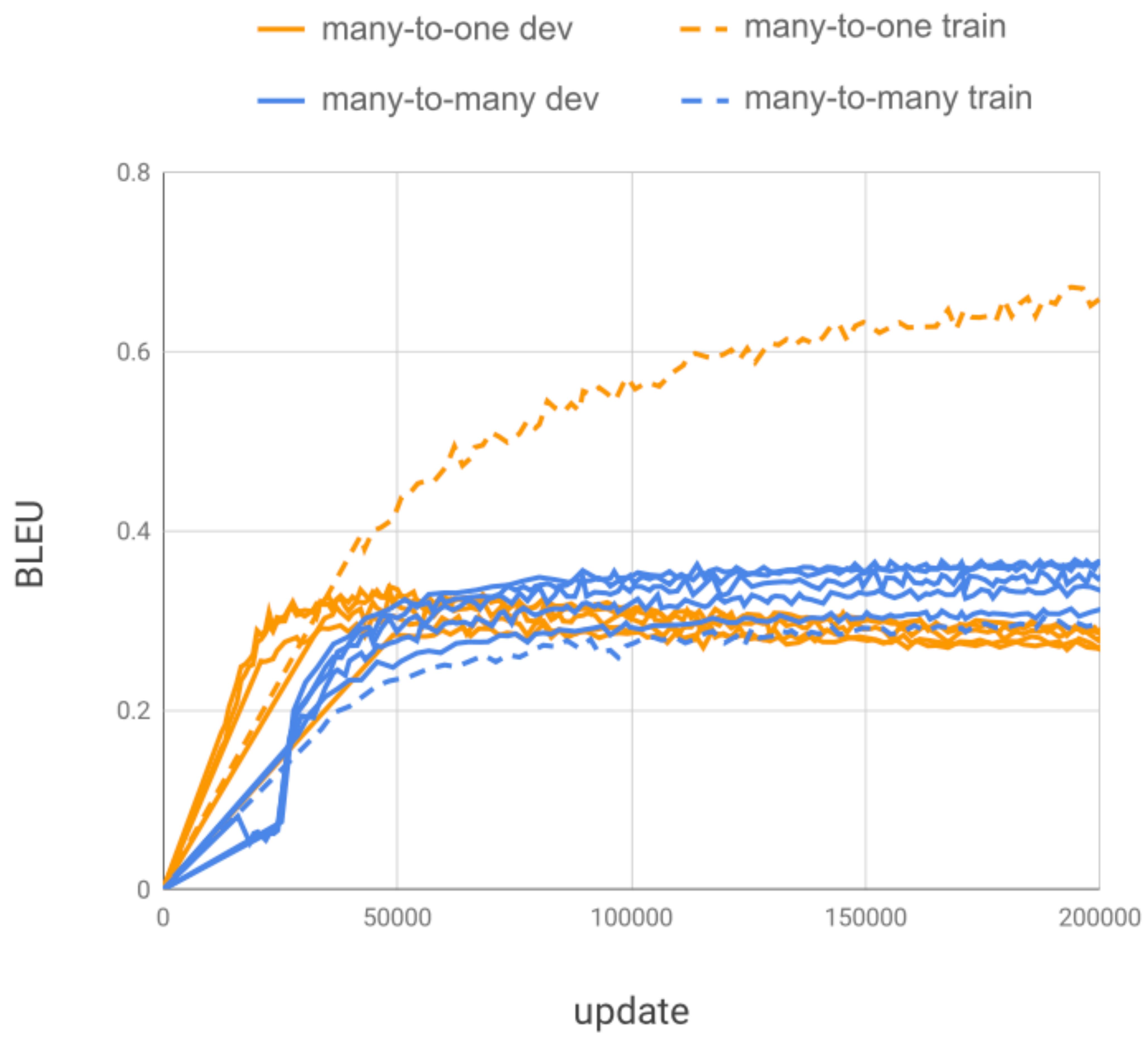
Multilinguality as Regularization

- The models we used are very large - prone to overfitting on the small datasets



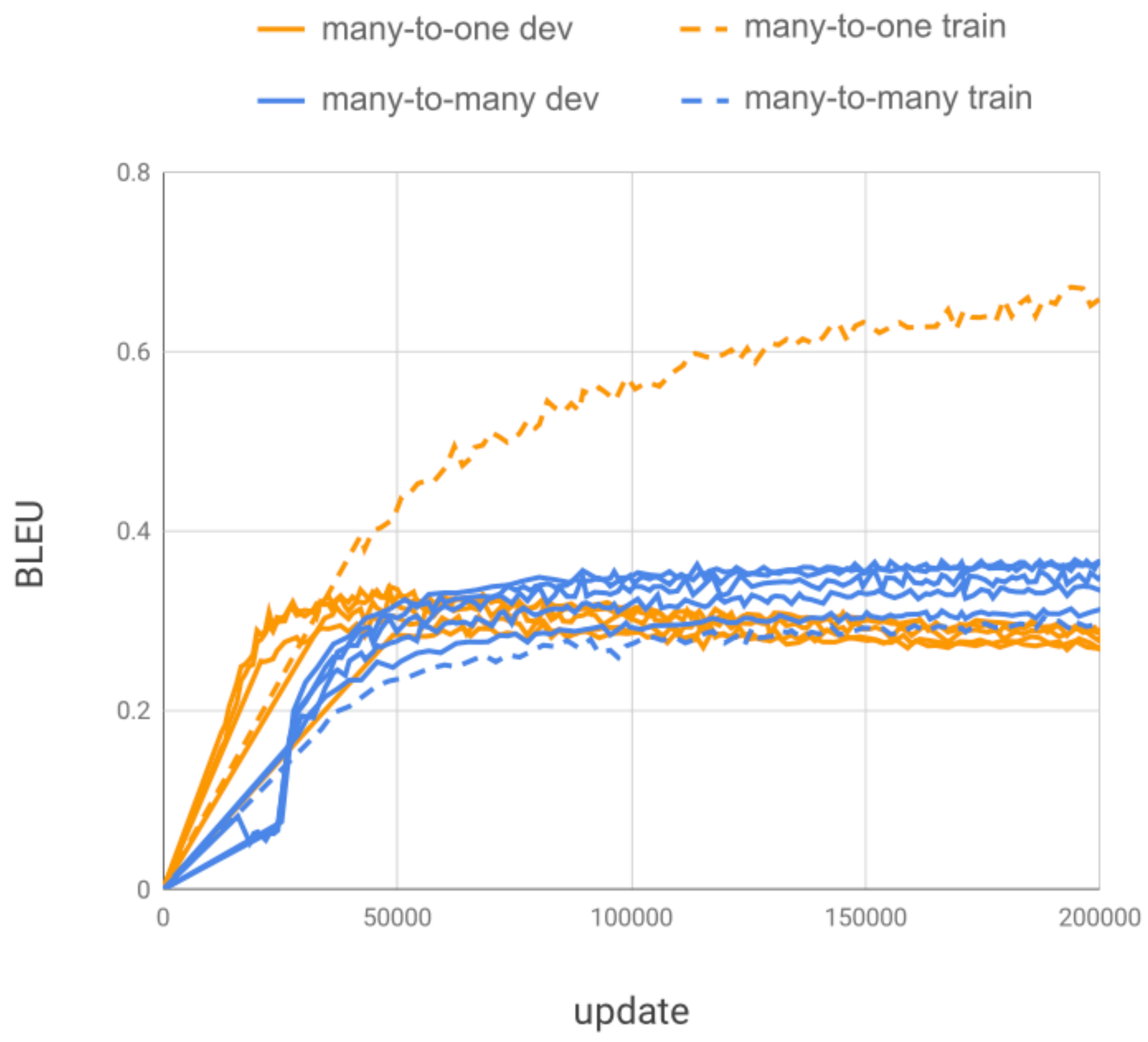
Multilinguality as Regularization

- The models we used are very large - prone to overfitting on the small datasets
- Having many target languages makes it harder to memorize, even with small data



Multilinguality as Regularization

- The models we used are very large - prone to overfitting on the small datasets
- Having many target languages makes it harder to memorize, even with small data
- Also easy to memorize since multi-way parallel



Evaluating out of English

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines
- Many-to-Many models are biased towards English in the target

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines
- Many-to-Many models are biased towards English in the target
- When English memorization is not an issue, better to train on fewer directions

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Experiments - High Resource

Experiments - High Resource

- We saw that:

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)
- Does this hold:

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)
- Does this hold:
  - With even more languages?

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)
- Does this hold:
  - With even more languages?
  - With larger, balanced, “real-world” datasets?

Experiments - High Resource

Experiments - High Resource

- Transformer Big(ger) models

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)
  - ~1M examples per language pair (balanced)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)
  - ~1M examples per language pair (balanced)
  - Not multi-way parallel

Results - Into English

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
  - When the data is large enough and not multi-way-parallel, memorization is not an issue and “less is more”

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
    - When the data is large enough and not multi-way-parallel, memorization is not an issue and “less is more”
- German and Italian outliers - due to interference

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
  - When the data is large enough and not multi-way-parallel, memorization is not an issue and "less is more"
- German and Italian outliers - due to interference
  - Many-to-one reached 38 BLEU when evaluated using German only dev-set, but degraded

Results - Out of English

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases
- Up to 6-8 BLEU improvement over baseline (Slovak, German)

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases
- Up to 6-8 BLEU improvement over baseline (Slovak, German)
- Less burden, not biased towards English

Analysis

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?
  - Both supervised and Zero-Shot

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?
  - Both supervised and Zero-Shot
- Keep model fixed, measure performance on 5 languages while varying the number of additional languages

Analysis - Supervised Directions

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

- Clear trade-off between number of languages and model accuracy

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

- Clear trade-off between number of languages and model accuracy
- Maybe we need even bigger models? 1M examples per language pair is not very large… (in MT scale)

Analysis - Zero-Shot Directions

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization

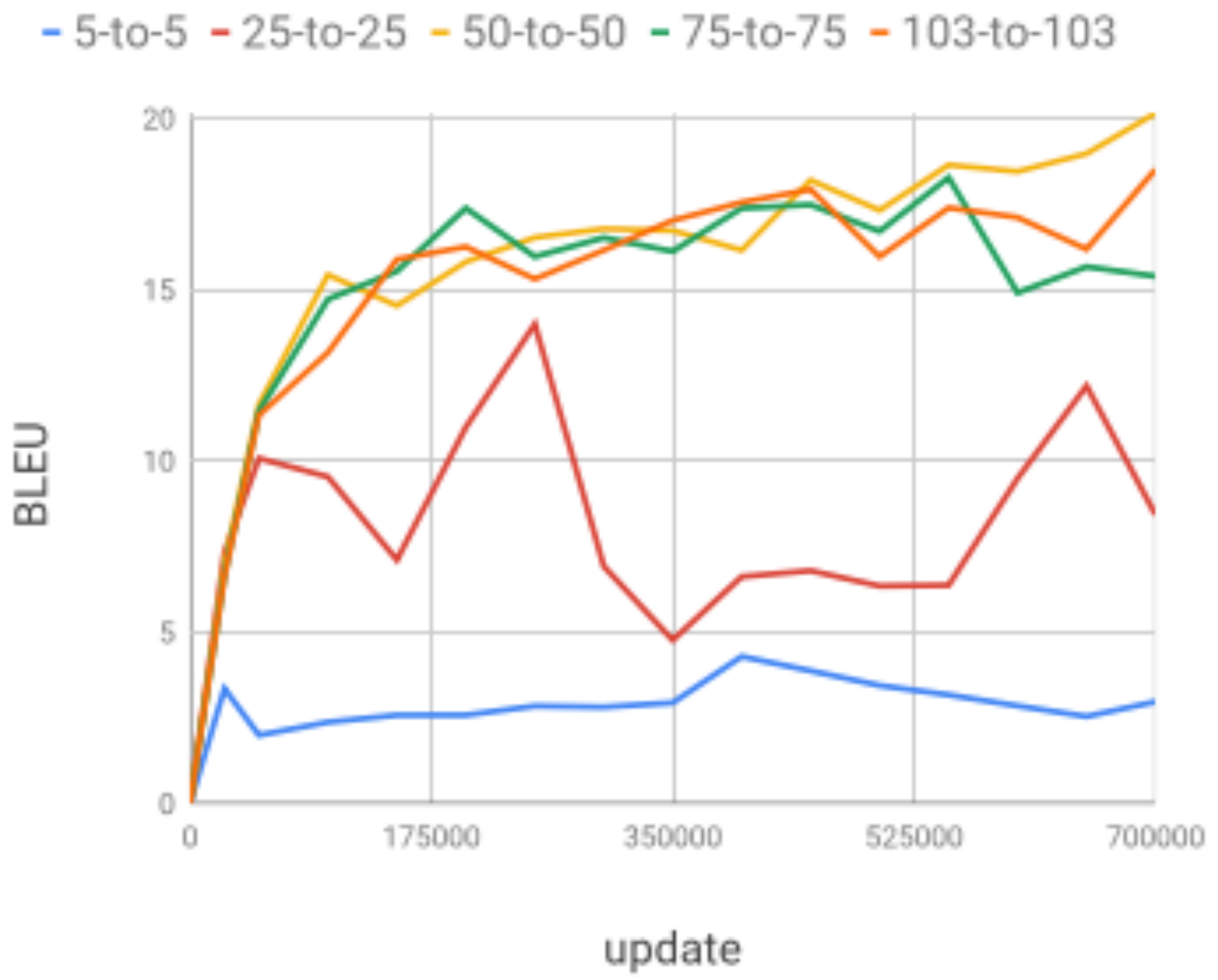
- Similar languages are much easier

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization
- Similar languages are much easier
- General trend - more languages, more generalization (interlingua?)

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17



Analysis - Internal Representations

Analysis - Internal Representations

- Kudugunta et al. 2019 investigated the representations learned by massively multilingual models

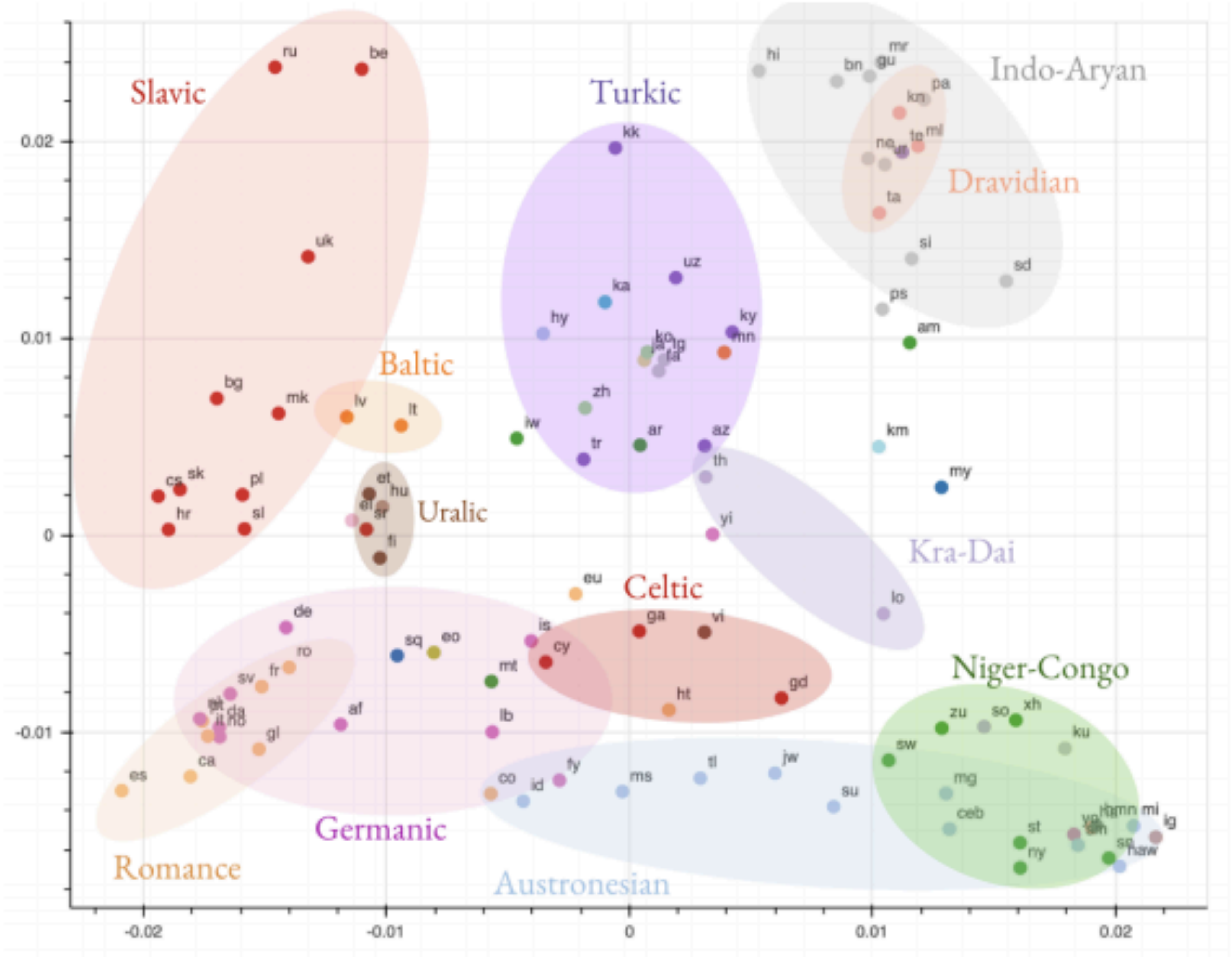


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Analysis - Internal Representations

- Kudugunta et al. 2019 investigated the representations learned by massively multilingual models
- Encoder representations of different languages cluster based on linguistic similarity

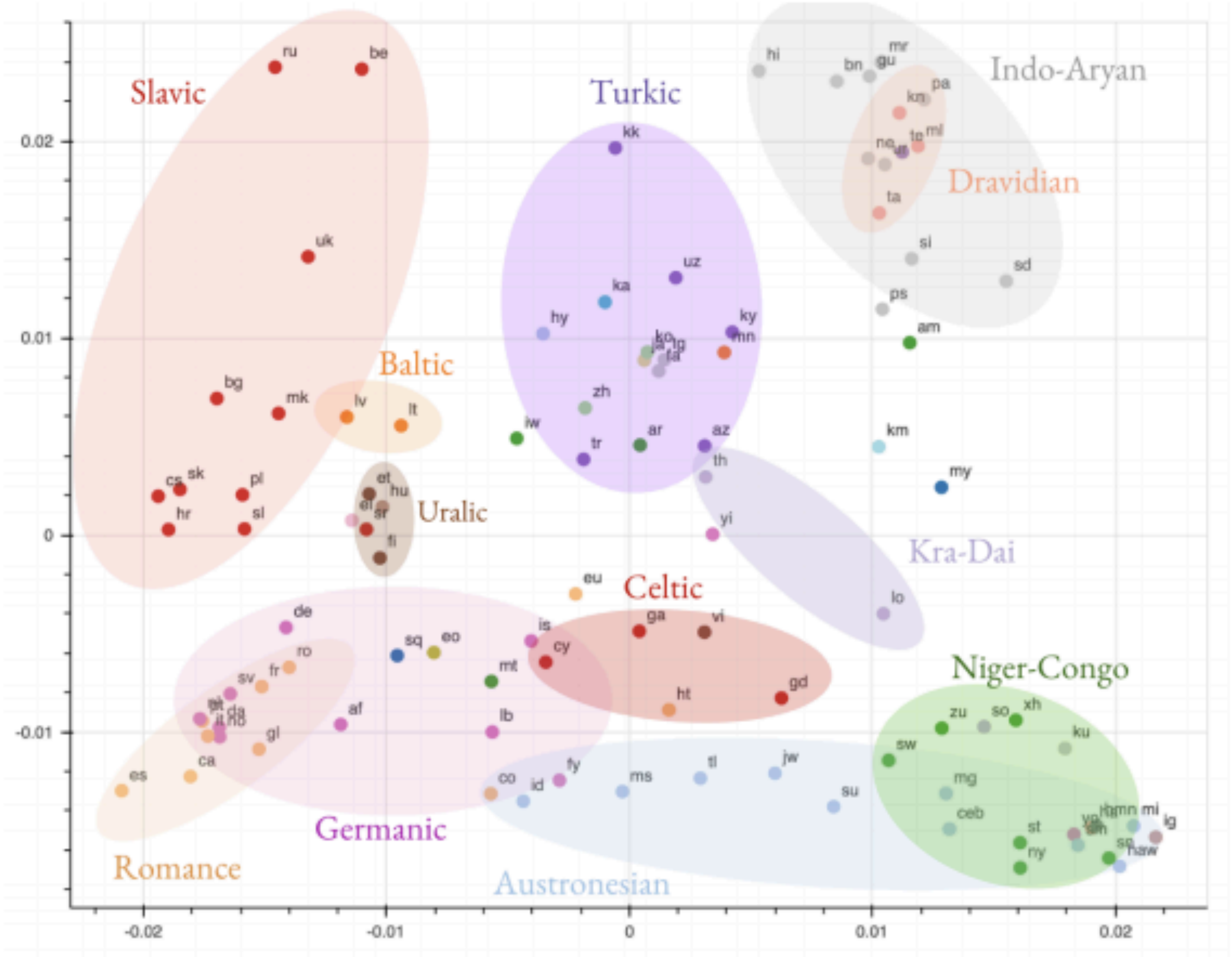


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Analysis - Internal Representations

- Kudugunta et al. 2019 investigated the representations learned by massively multilingual models
- Encoder representations of different languages cluster based on linguistic similarity
- Representations of a source language learned by the encoder are dependent on the target language, and vice-versa

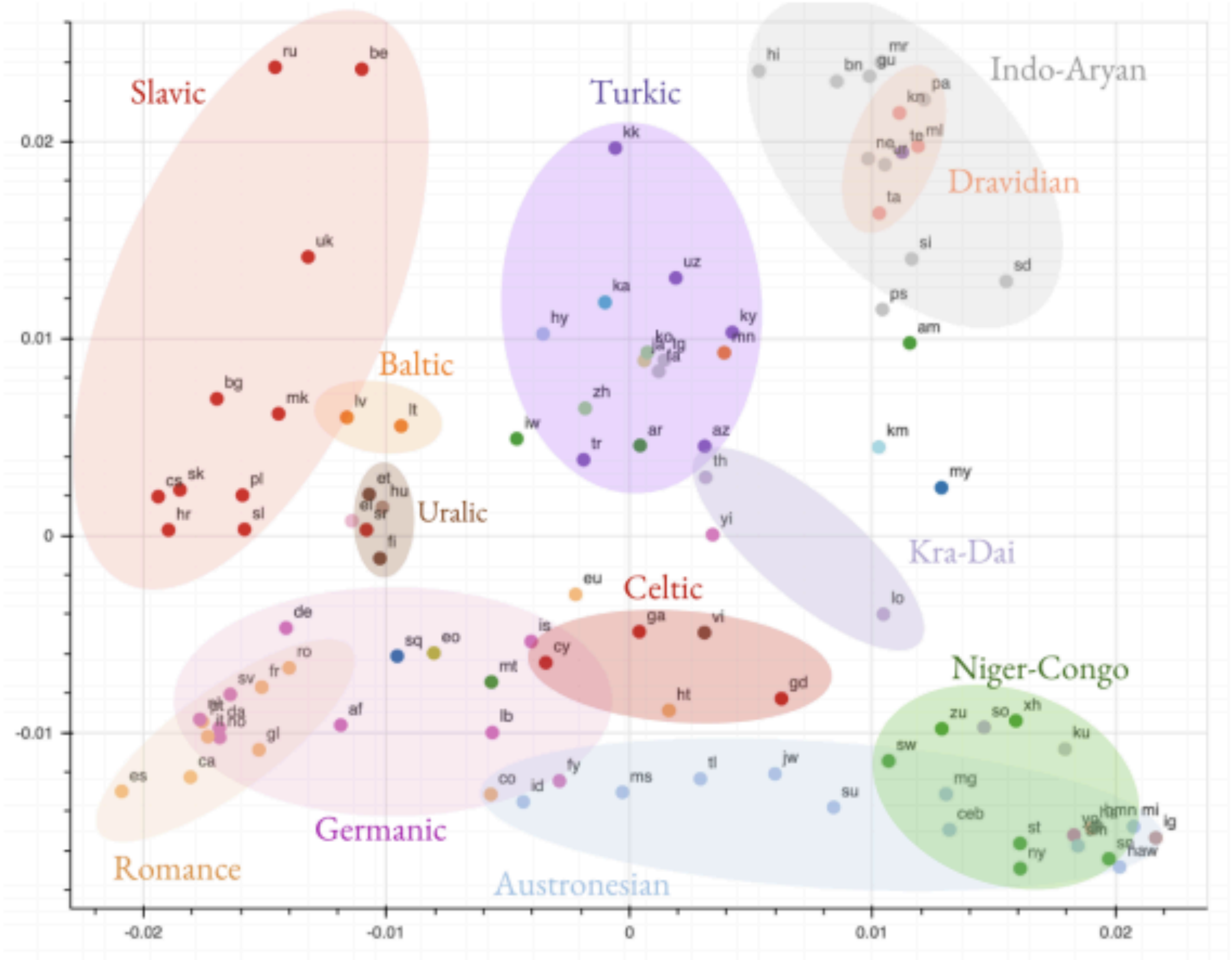


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Analysis - Internal Representations

- Kudugunta et al. 2019 investigated the representations learned by massively multilingual models
- Encoder representations of different languages cluster based on linguistic similarity
- Representations of a source language learned by the encoder are dependent on the target language, and vice-versa
- Representations of high resource and/or linguistically similar languages are more robust when fine-tuning on an arbitrary language pair

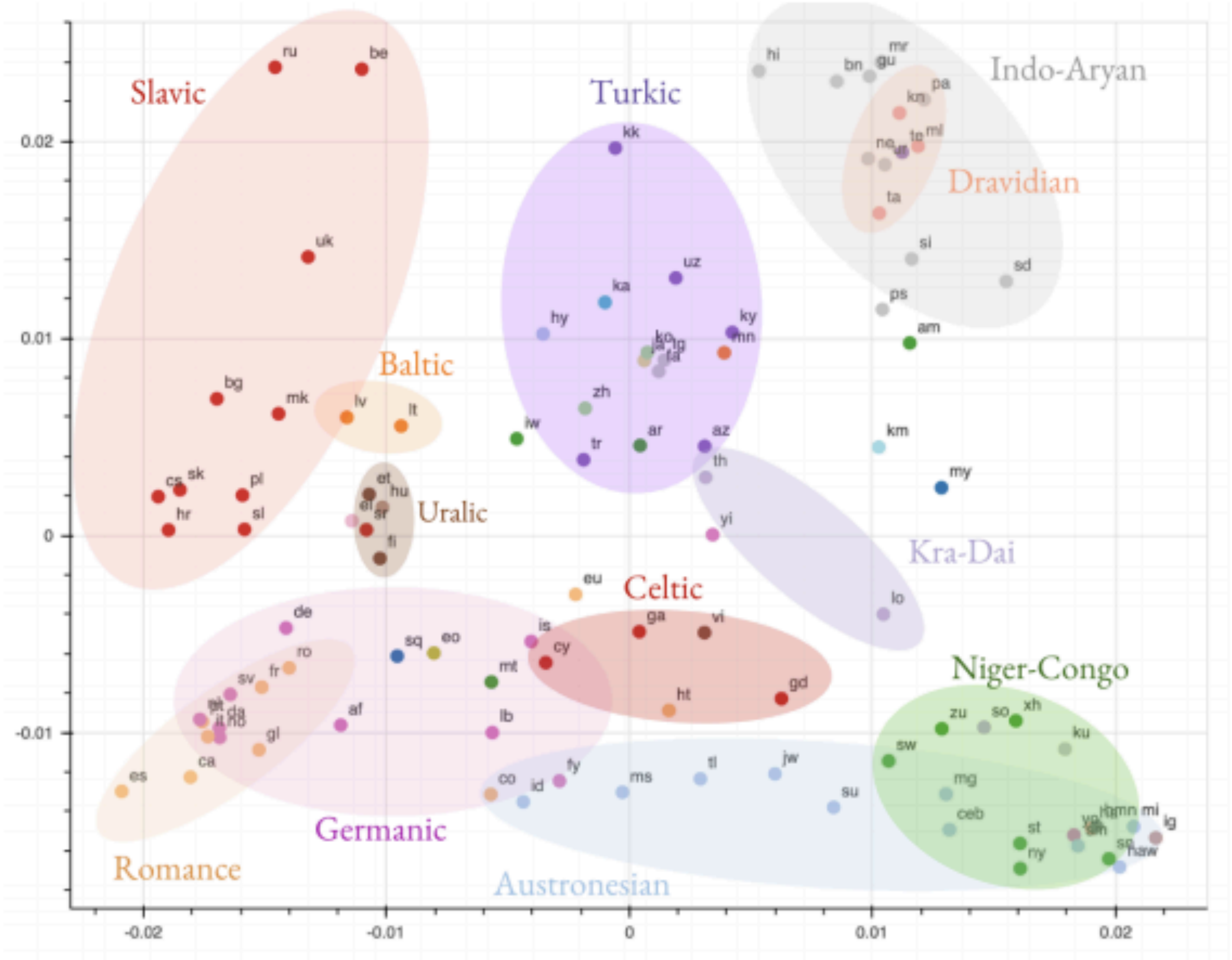


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

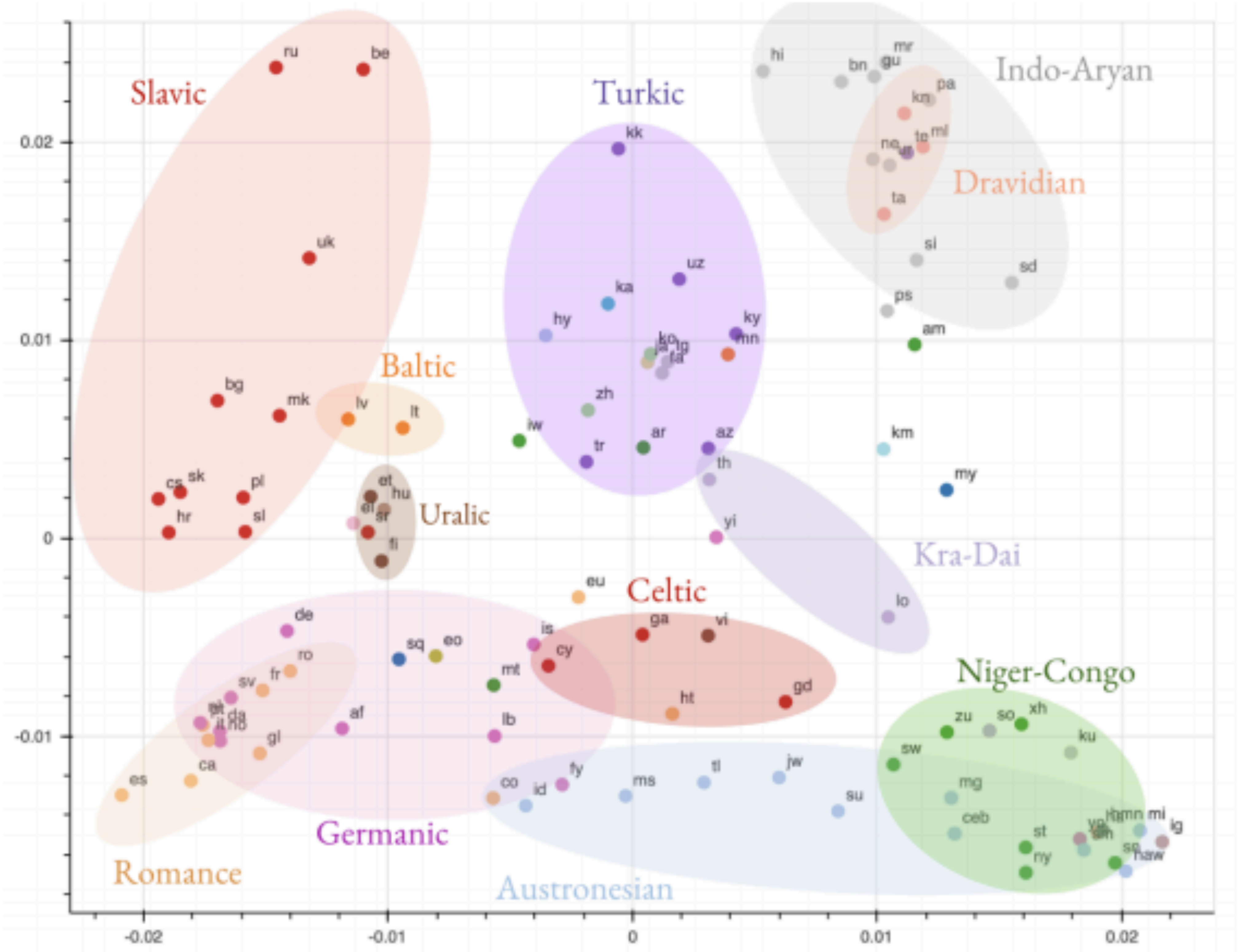


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well

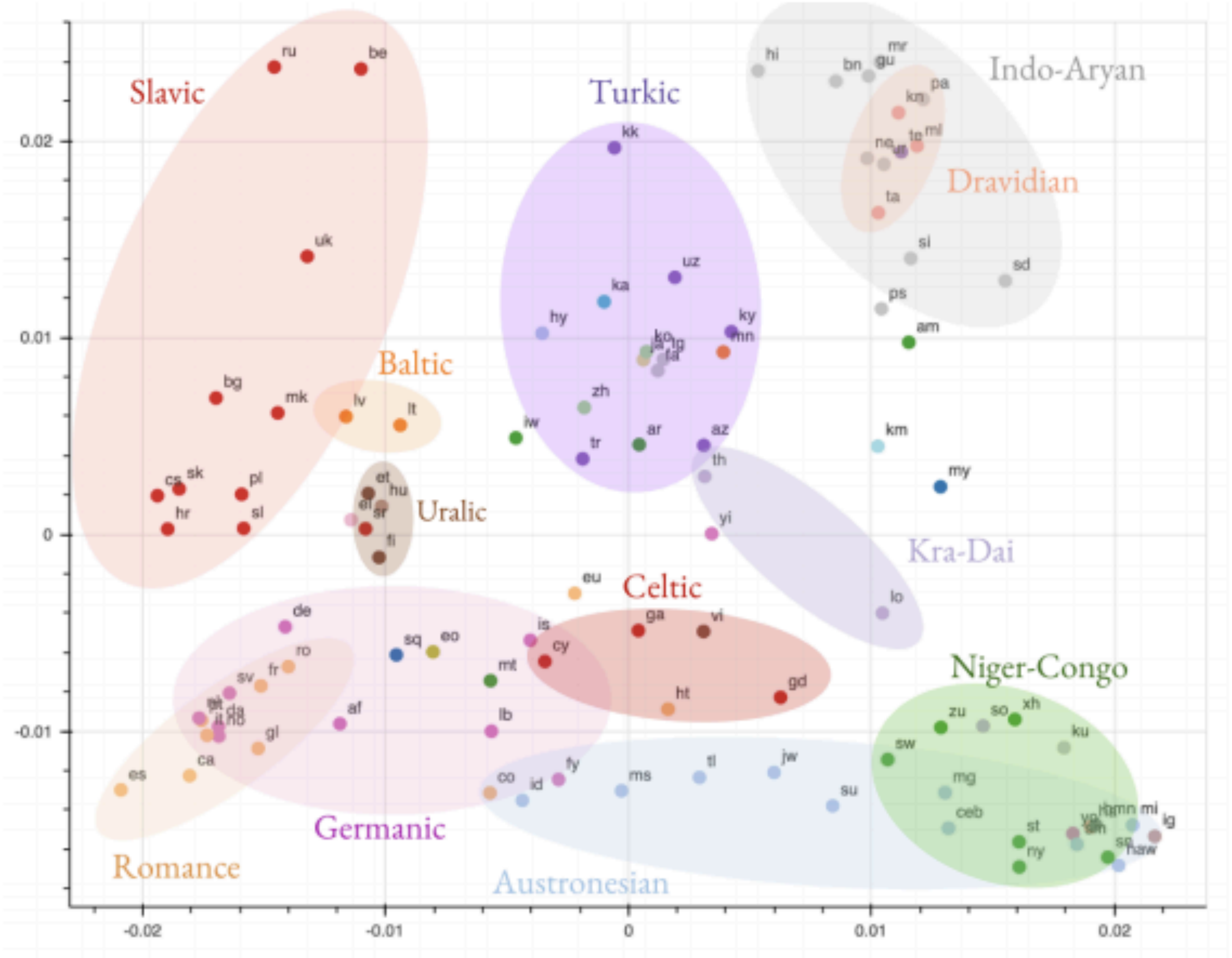


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:

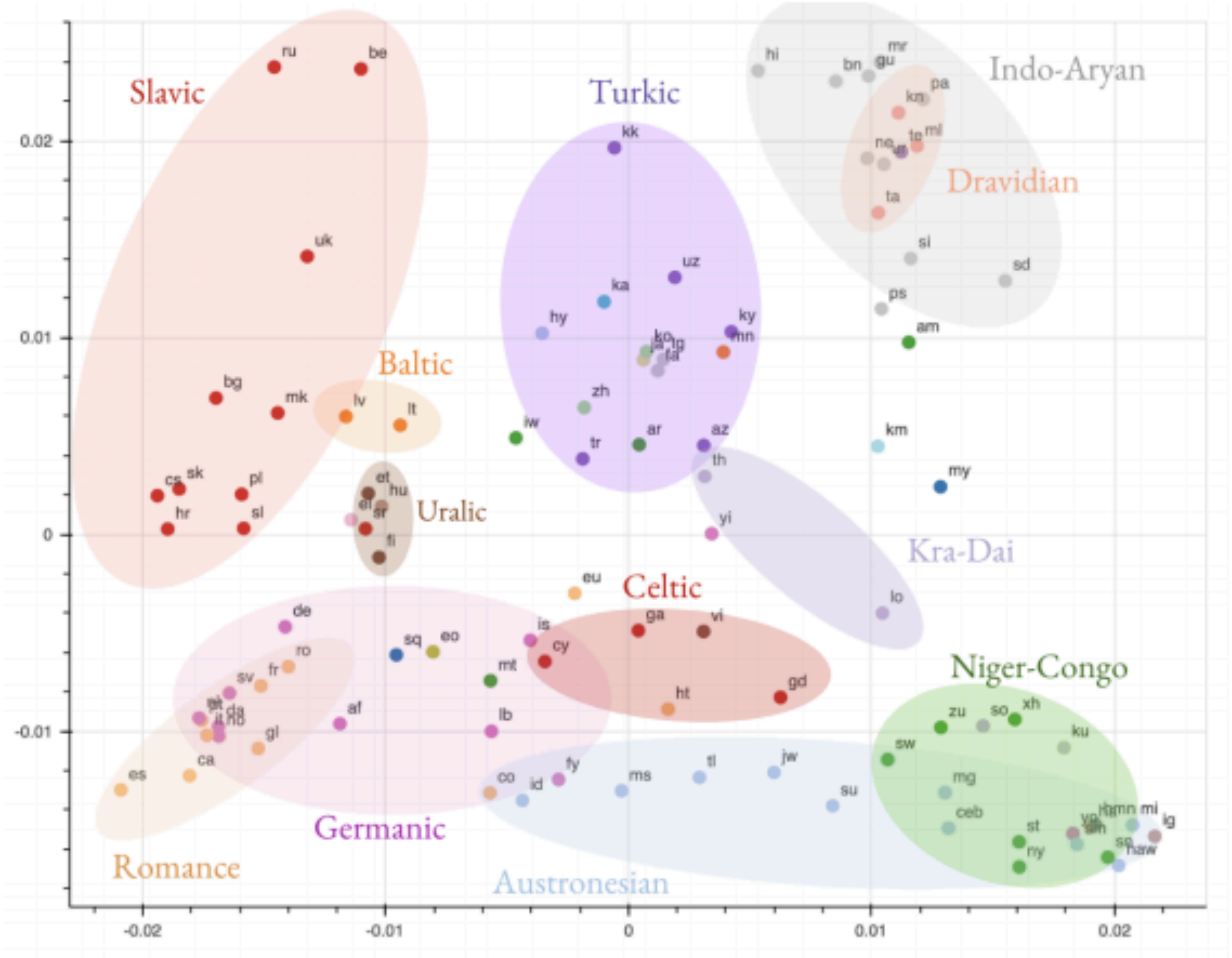


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs

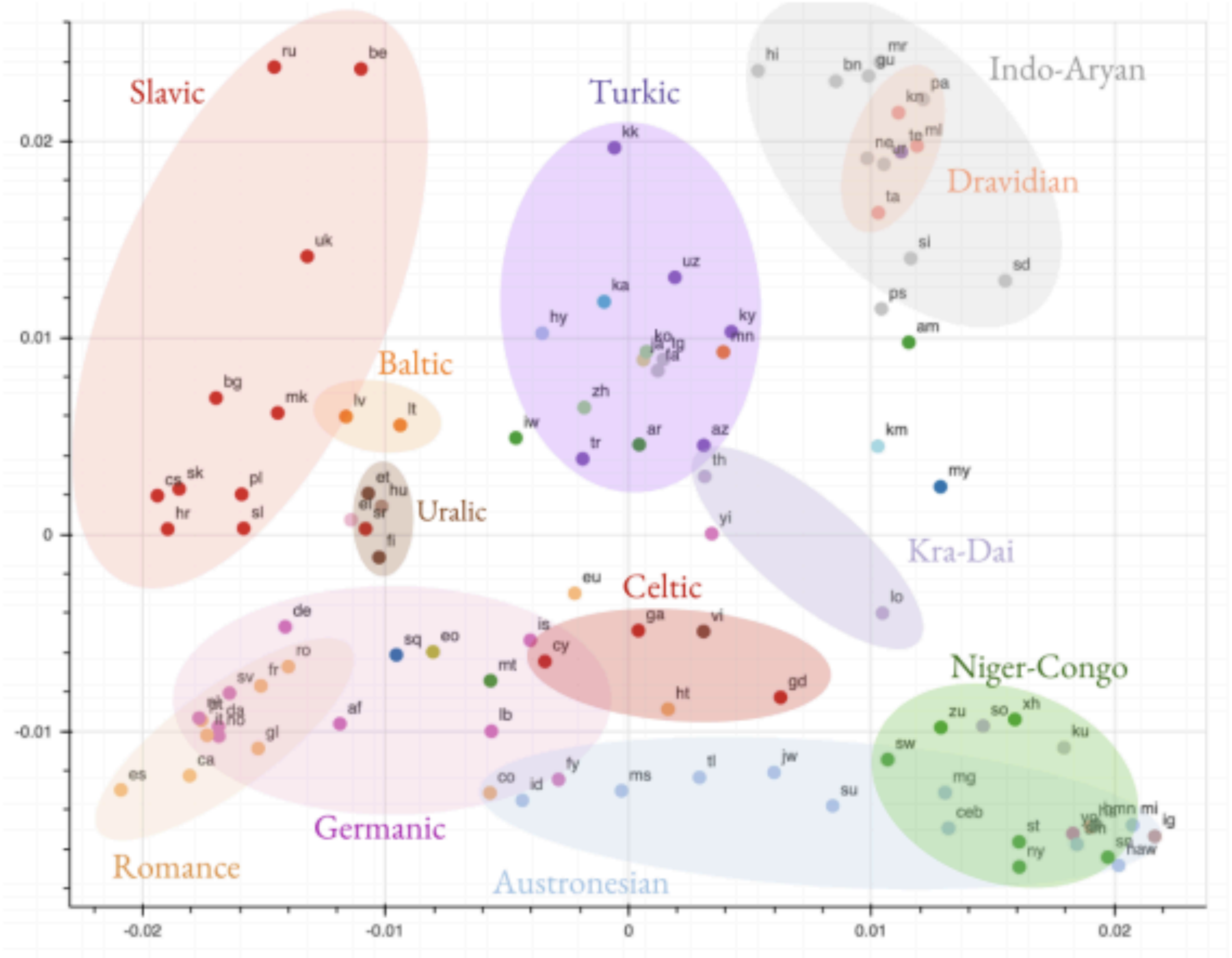


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models

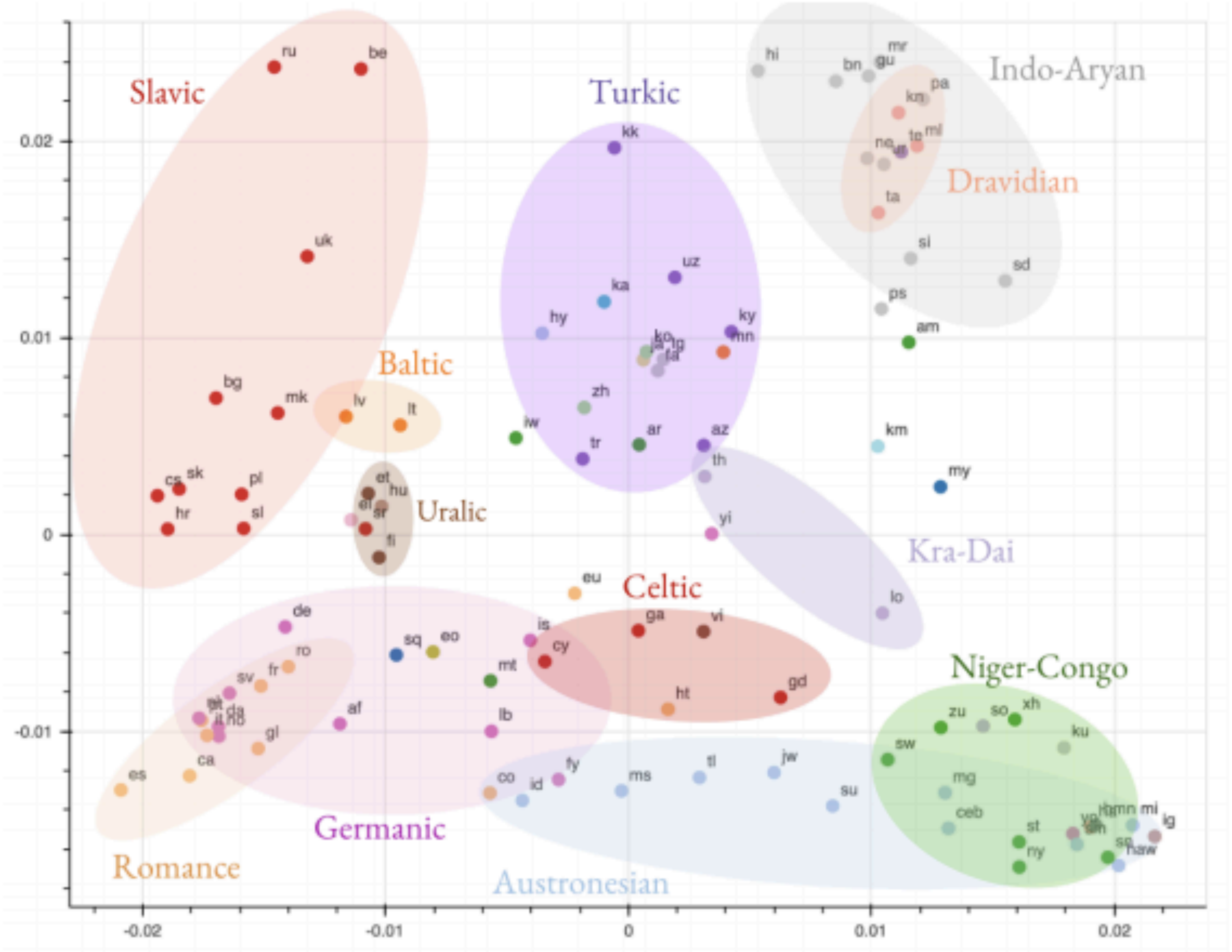


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
    - Word-based models + UNKs
    - Character level models
    - Middle-ground: subword models (BPE)

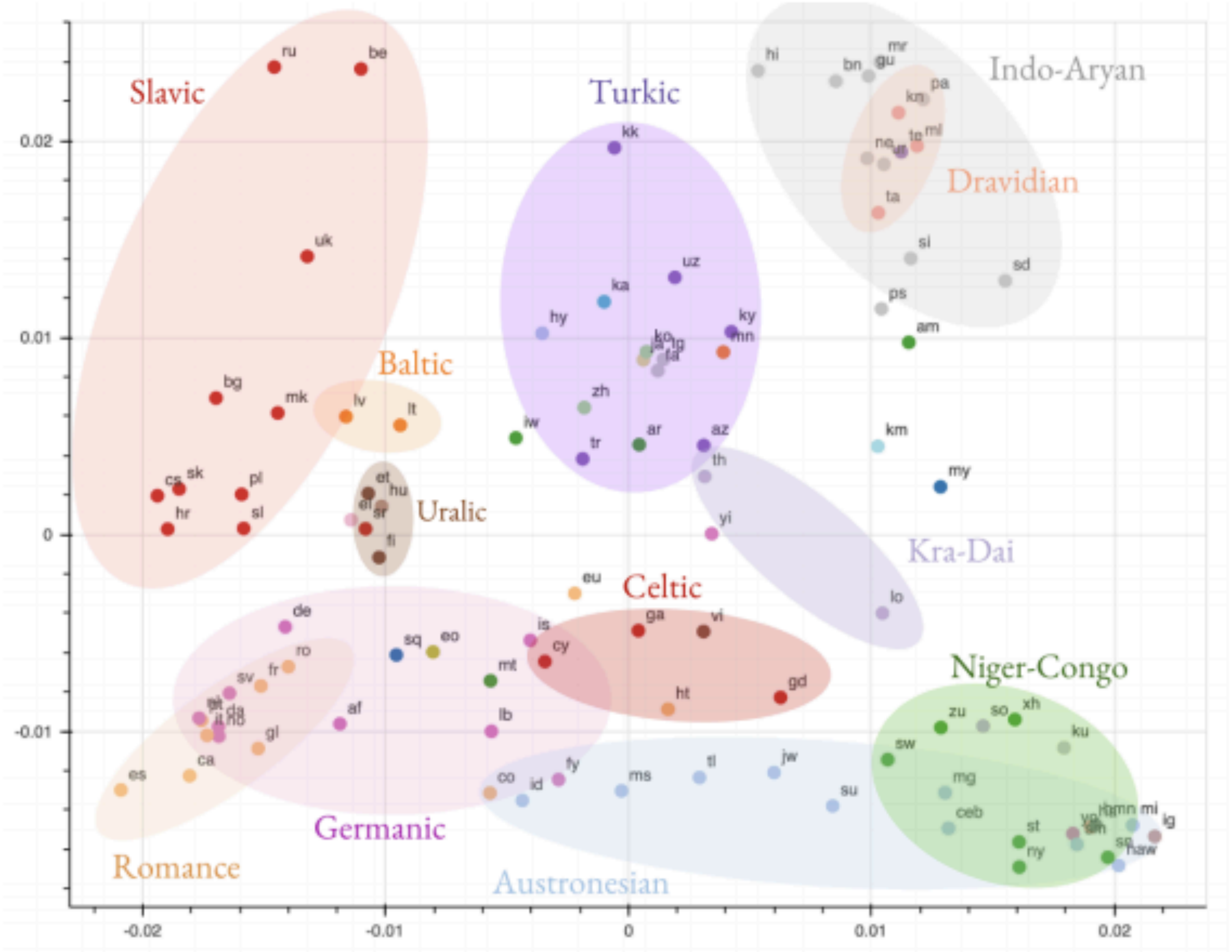


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models
  - Middle-ground: subword models (BPE)
- To use **monolingual data**:

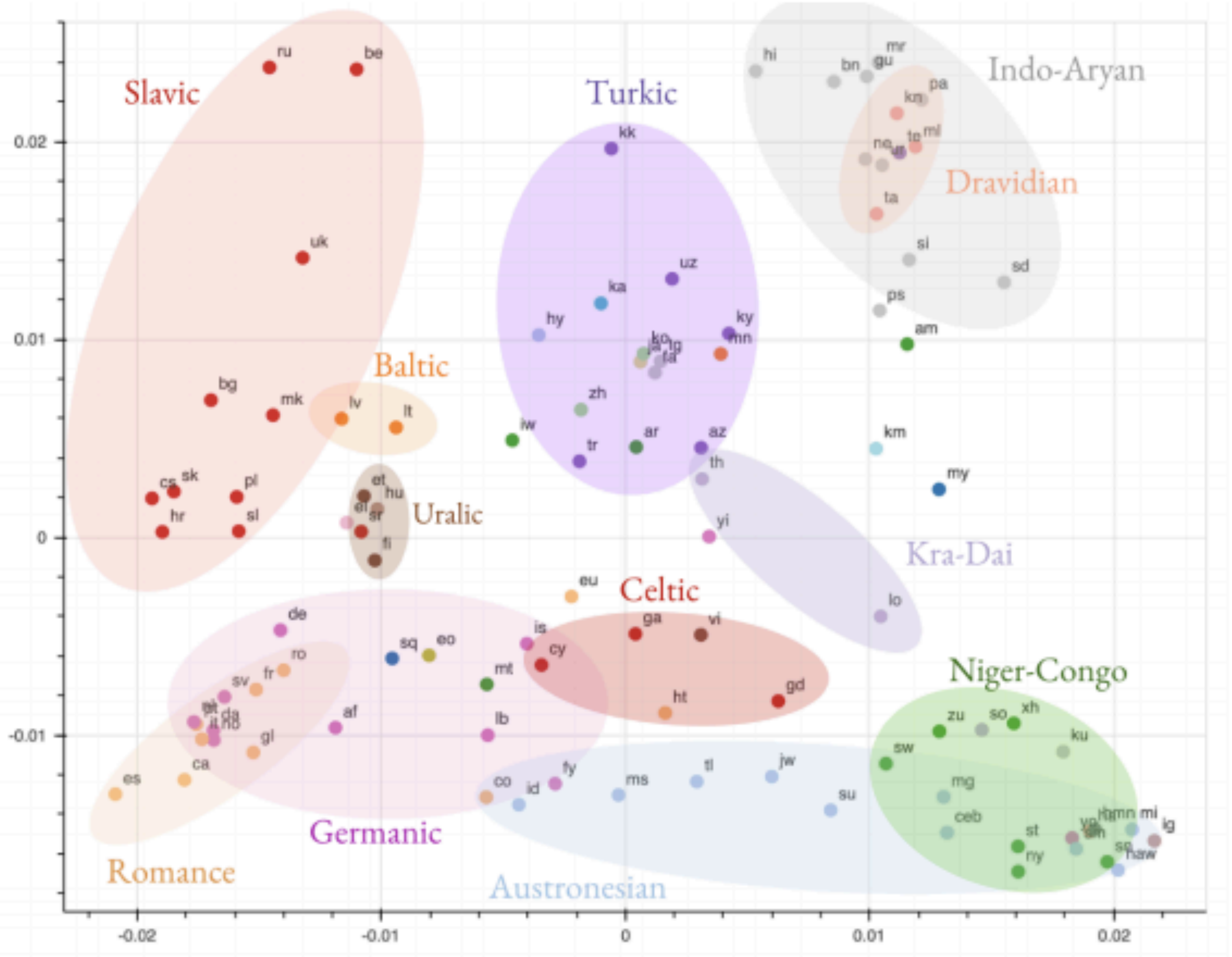


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models
  - Middle-ground: subword models (BPE)
- To use **monolingual data**:
  - Language model fusion

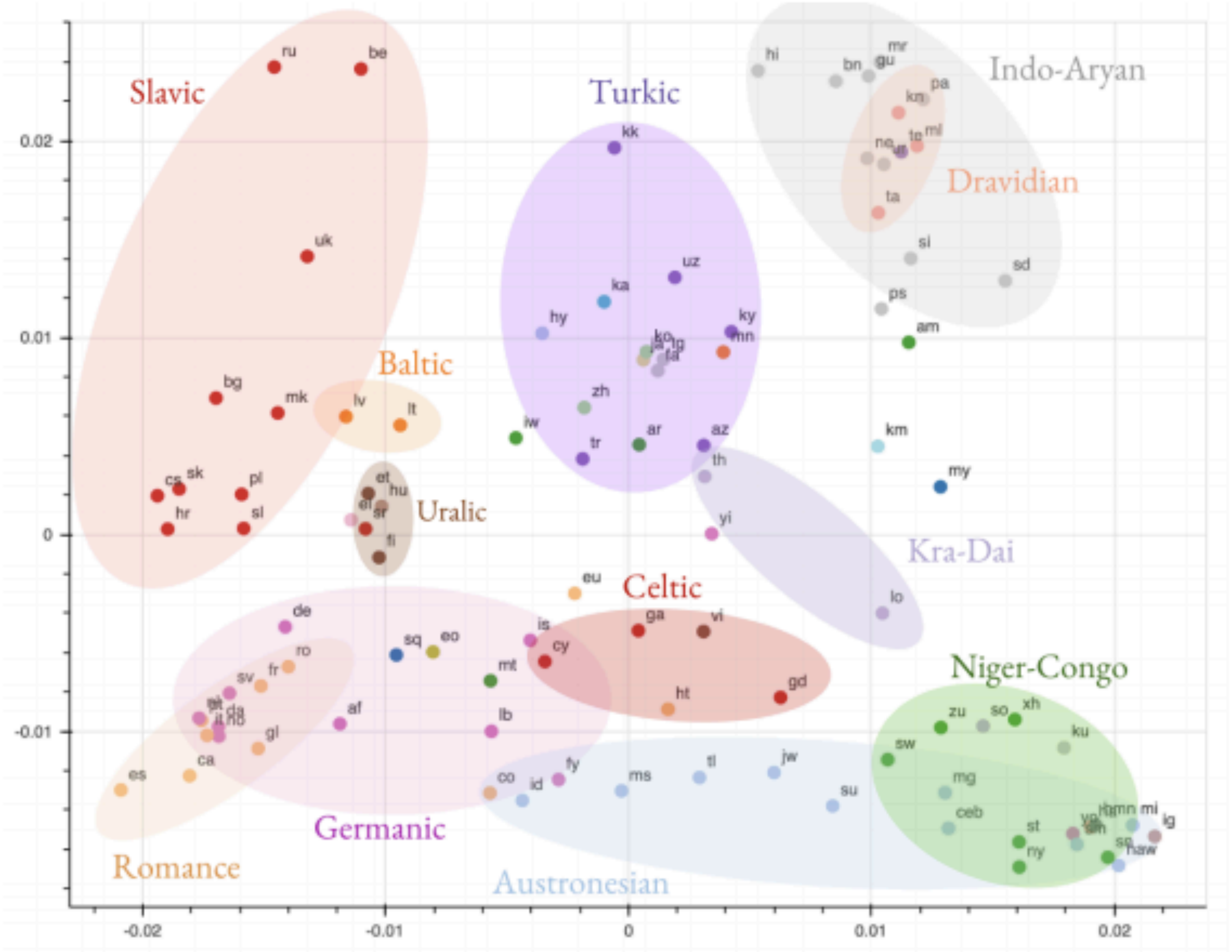


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models
  - Middle-ground: subword models (BPE)
- To use **monolingual data**:
  - Language model fusion
  - Back-translation

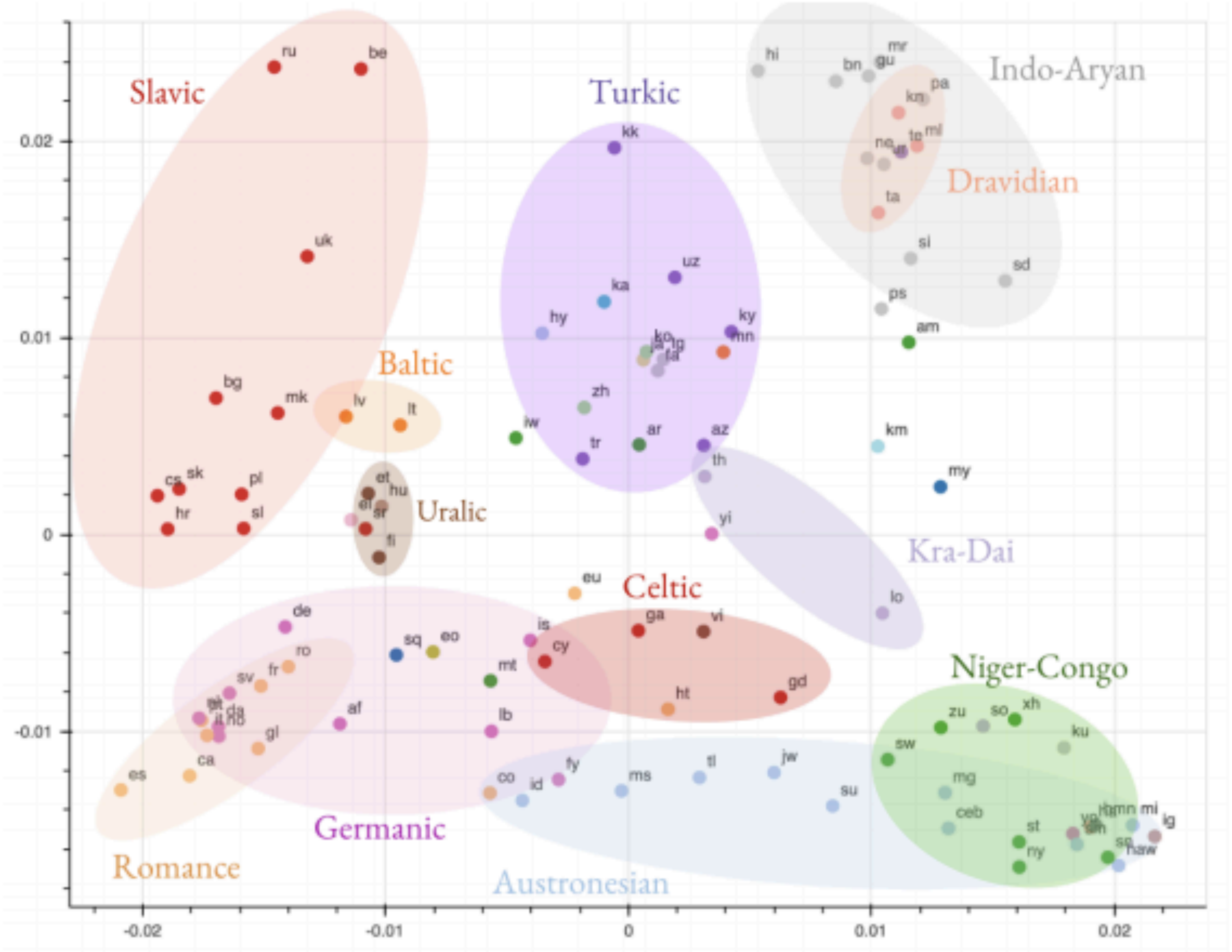


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models
  - Middle-ground: subword models (BPE)
- To use **monolingual data**:
  - Language model fusion
  - Back-translation
  - Transfer learning

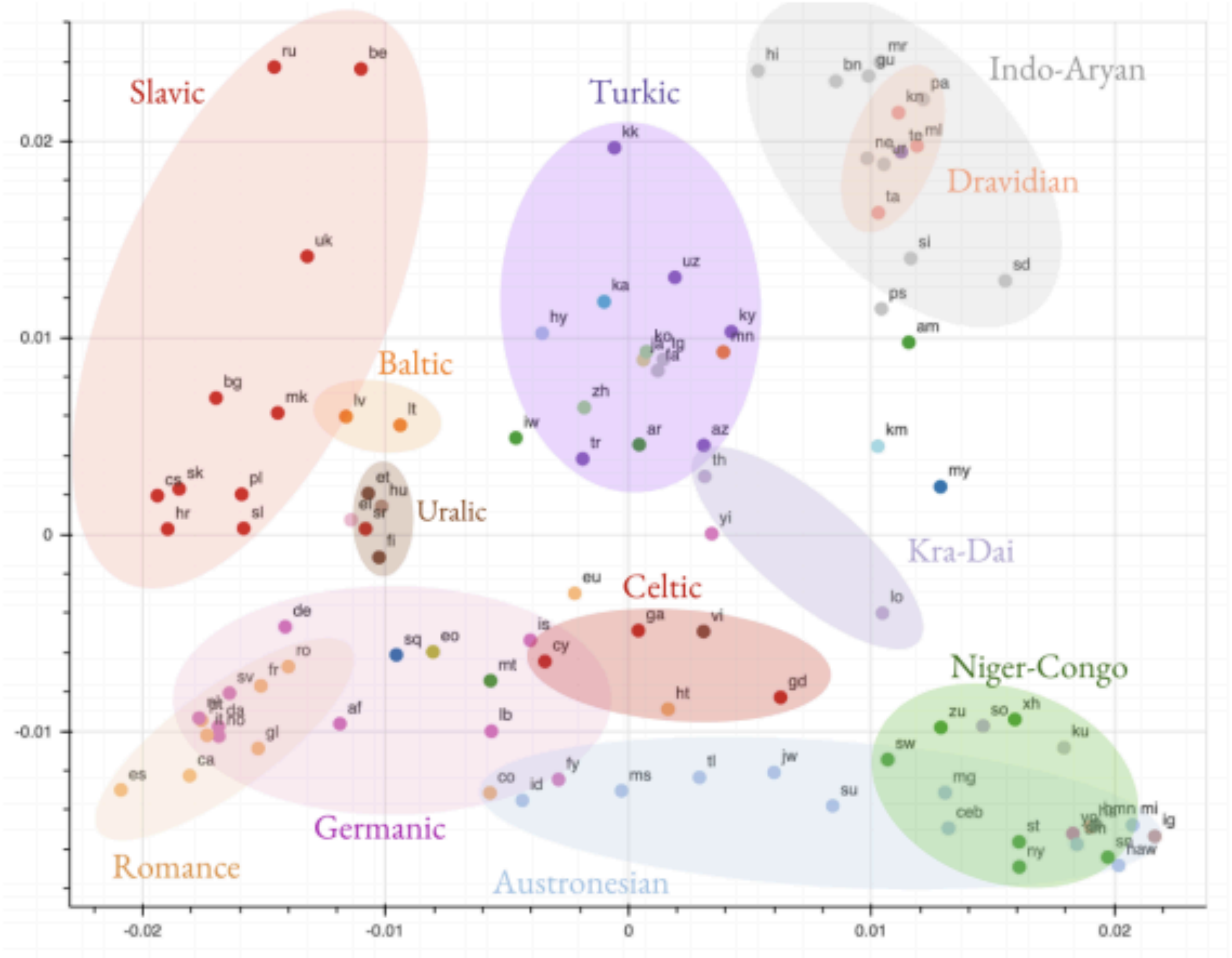


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.

Summary

- NMT is a strong tool, but needs some tweaks to work well
- To handle **large vocabularies**:
  - Word-based models + UNKs
  - Character level models
  - Middle-ground: subword models (BPE)
- To use **monolingual data**:
  - Language model fusion
  - Back-translation
  - Transfer learning
- **Multilingual NMT -** more transfer, practical

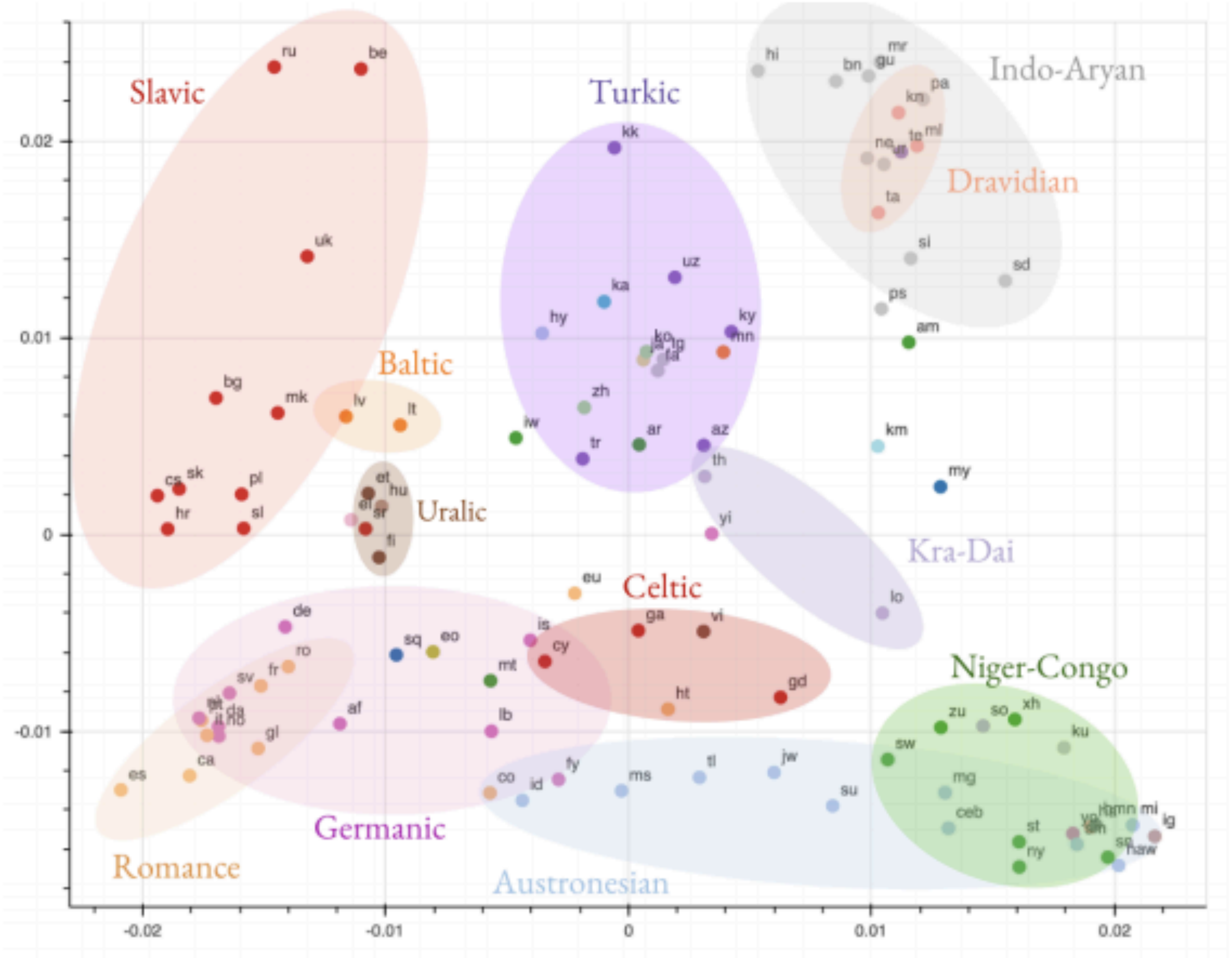
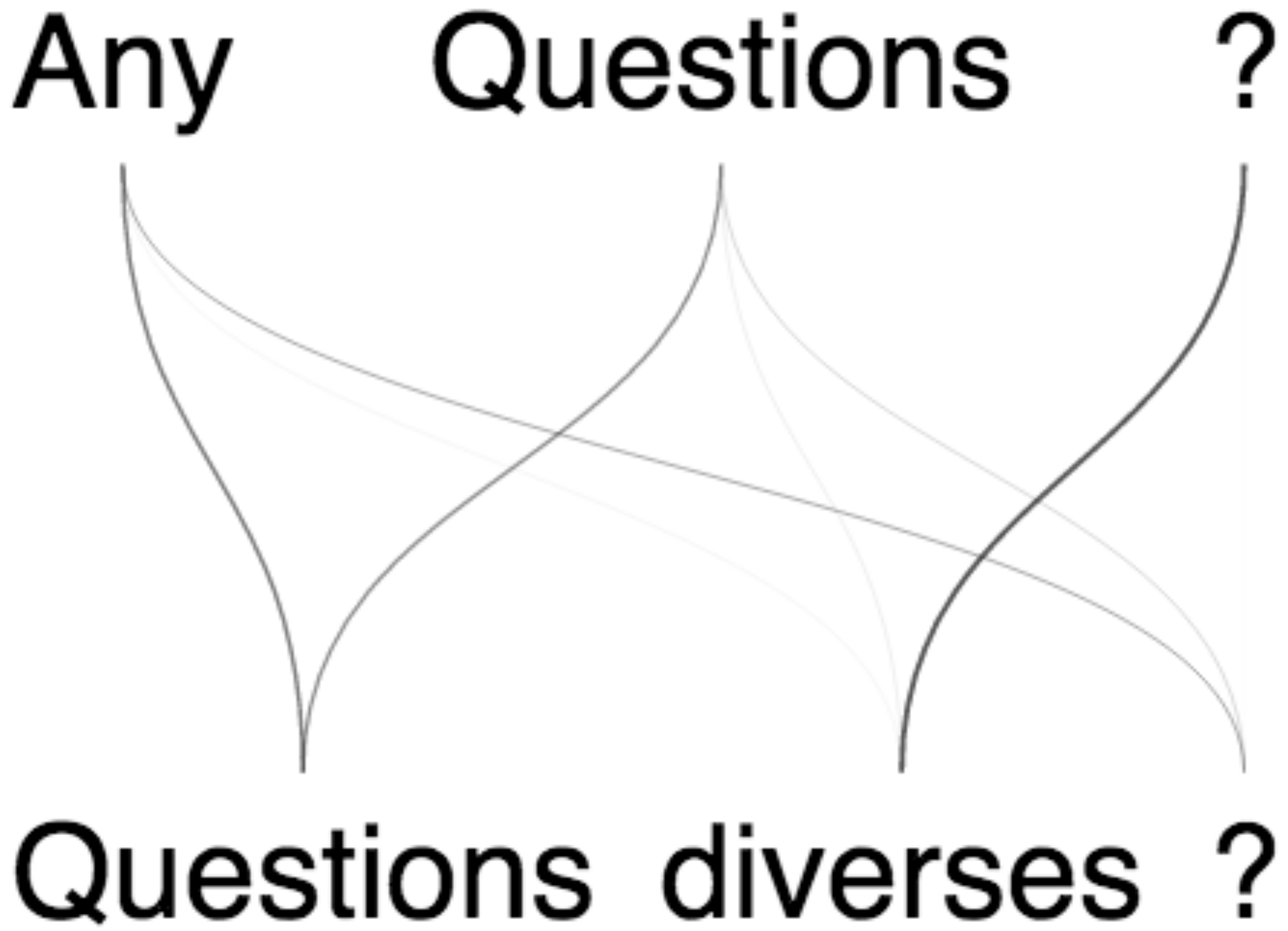


Figure 2: Visualizing clustering of the encoder representations of all languages, based on their SVCCA similarity. Languages are color-coded by their linguistic family. Best viewed in color.


Any Questions ?
Questions diverses ?