CSE 493s/599s
Lecture 1.
Tokenization for LLMs

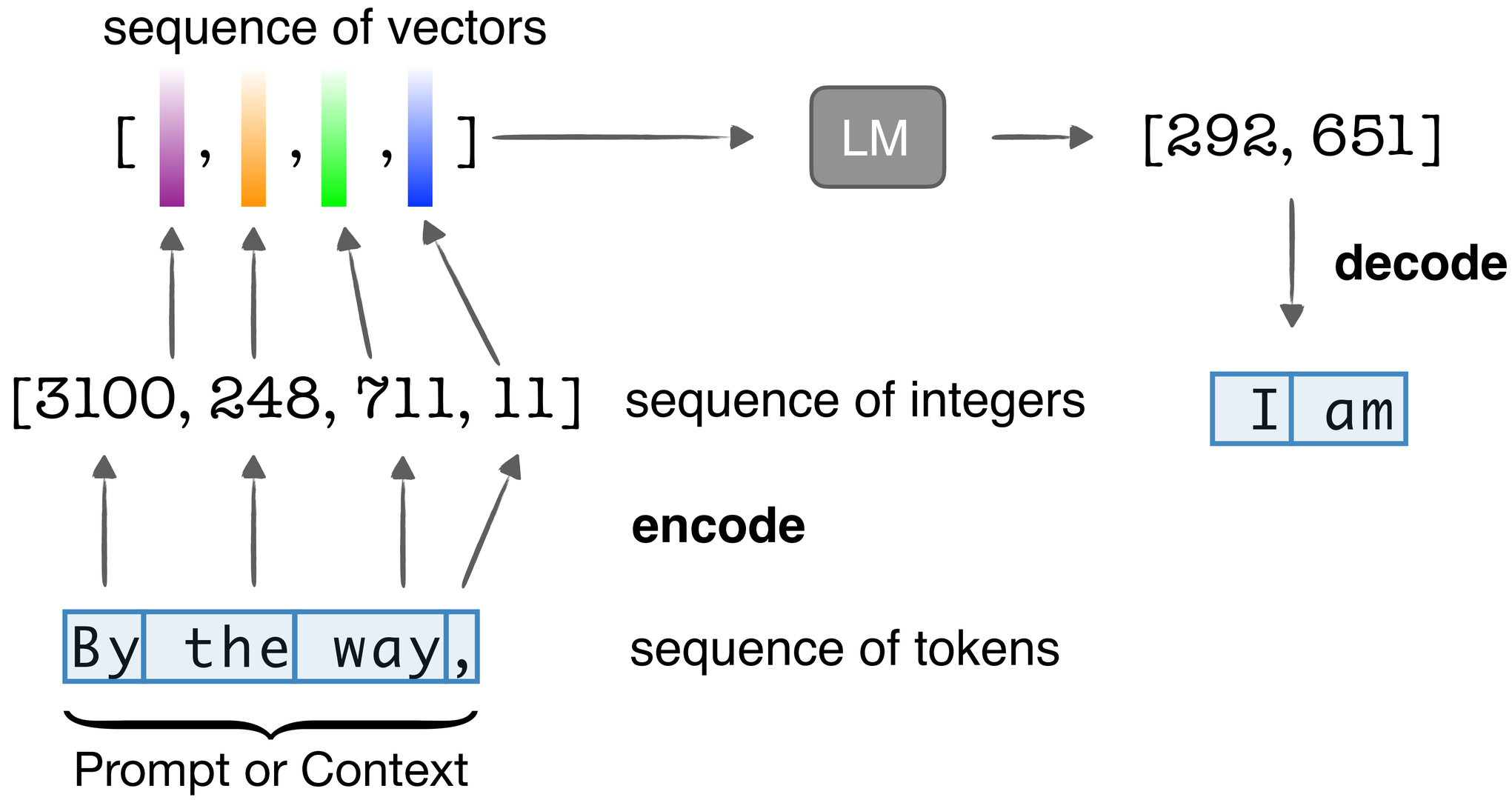
Sewoong Oh

First half of CSE493S

- Tokenizers
- Language models
- Architecture
  - Transformers
  - Mixture-of-experts
- Inference
  - Speculative decoding
  - In-context learning
  - Chain-of-thought prompting
  - Test-time compute
- Post-training
  - Parameter Efficient fine-tuning
  - Alignment

Tokenizer for LLMs

Tokens are sequences of characters used by LMs to understand text, bridging the text (based on characters) and LM (processes numbers)



Modern transformer-based LMs use subword tokenization

- Character-level:

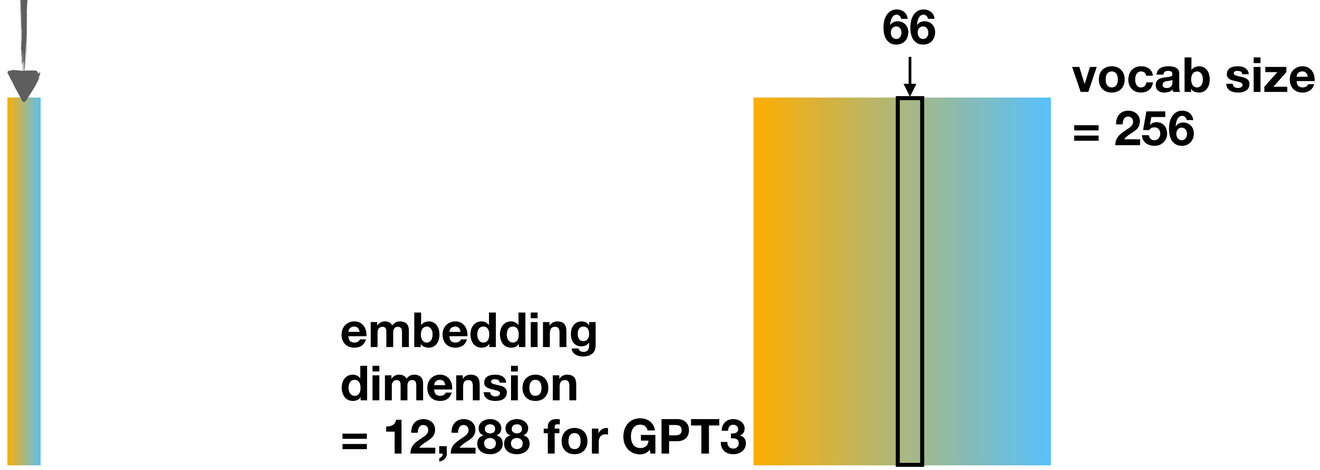
Character-level tokenization

- Step 1: the text is broken into a sequence of characters

- Step 2: **Look-up table** (for example, ASCII code) converts characters into unique integer IDs (in this case Bytes)

[66, 121, 32, 116, 104, 101, 32, 119, 97, 121, 44, 32, 73, 32, 97, 109, 32, 97, 32, 102, 97, 110, 32, ...

- Step 3: **Learned embedding table** converts IDs into embedding representations



Character-level tokenization

- Step 1: the text is broken into a sequence of characters

- What is wrong with **character-level tokenization**?

Character-level tokenization

- Step 1: the text is broken into a sequence of characters

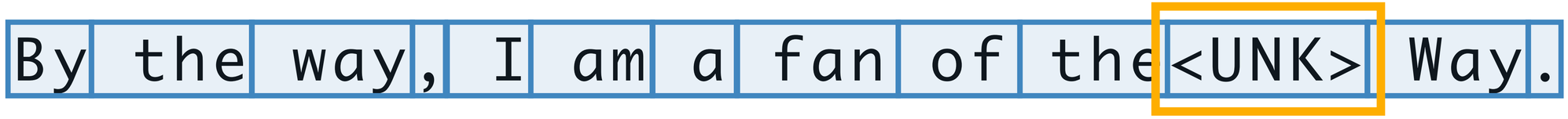
B	y		t	h	e		w	a	y	,		I		a	m		a		f	a	n		o	f		t	h	e		M	i	l	k	y		w	a	y	.

- What is wrong with **character-level tokenization**?
  - **Efficiency**: the number of tokens needed to represent text is quite large, which increases the input dimension of the model
    (run-time of a language model is quadratic in the context length)
  - **Language diversity**: only handles English
    - A variable length code of UTF-8 is used to handle world language (together with BPE to be explained later)

Modern transformer-based LMs use subword tokenization

- Character-level:

- Word-level:

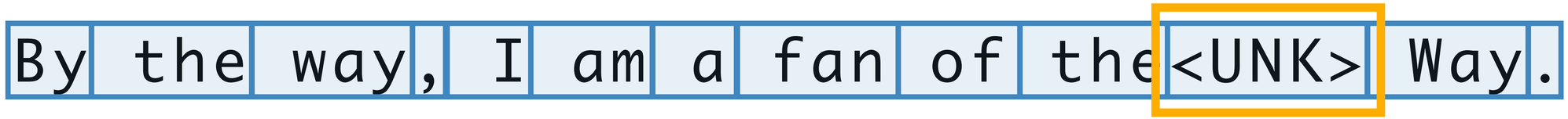


  - Much more efficient:
    - about 5 characters per English word on average $\Longrightarrow$ 5x compression rate for context
  - Typical vocabulary size ~= 170,000 words (e.g., Oxford English Dictionary)
    - but can encounter new words that is not in the vocab, due to morphology, names, numerals, and misspellings, which is represented by a special token **<UNK>,** since there are many more uncommon words

Modern transformer-based LMs use subword tokenization

- Character-level:

- Word-level:



- **Subword-level:**

Byte Pair Encoding (BPE)

- Universal method for learning subword tokenizers today
- Introduced by Sennrich et al. 2016 and popularized by GPT-2 (2019)
- Main idea: build vocabulary of tokens bottom-up by repeatedly merging frequent pair of tokens
- **Colab** for playing with BPE courtesy of Tayyib Ul Hassan Gondal

Sennrich, Haddow, Birch, “Neural Machine Translation of Rare Words with Subword Units”, ACL 2016
Radford et al., “Language Models are Unsupervised Multitask Learners”, 2019

Byte Pair Encoding (BPE)

Training Data

```
Proof of the Milky Way
consisting of many stars
came in 1610 when Galileo
Galilei used a telescope to
study the Milky Way and
discovered that it is
composed of a huge number
of faint stars.
```

Given **training data** D

Training Data

{Proof, _of, _the, _Milky, _Way, _consisting, _of, _many, _stars, _came, _in, _1610, _when, _Galileo, _Galilei, _used, _a, _telescope, _to, _study, _the, _Milky, _Way, _and, _discovered, _that, _it, _is, _composed, _of, _a, _huge, _number, _of, _faint, _stars.}

Because BPE is a sub-word tokenizer, you never need to look beyond the white space:
Pretokenize D by splitting on whitespace

comma indicates pretokenizer separation

Training Data

_ P r o o f, _ o f, _ t h
e, _ M i l k y, _ W a y, _
c o n s i s t i n g, _ o f,
_ m a n y, _ s t a r s, _ c
a m e, _ i n, _ 1 6 1 0, _
w h e n, _ G a l i l e o, _
G a l i l e i, _ u s e d, _
a, _ t e l e s c o p e, _ t
o, _ s t u d y, _ t h e, _
M i l k y, _ W a y, _ a n
d, _ d i s c o v e r e d, _
t h a t, _ i t, _ i s, _ c
o m p o s e d, _ o f, _ a,
_ h u g e, _ n u m b e r, _
o f, _ f a i n t, _ s t a r
s .

Split pretokenized D into sequences of **bytes=characters**

comma indicates pretokenizer separation

Training Data

_ P r o o f, _ o f, _ t h e, _ M i l k y, _ W a y, _ c o n s i s t i n g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ i n, _ 1 6 1 0, _ w h e n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _ a, _ t e l e s c o p e, _ t o, _ s t u d y, _ t h e, _ M i l k y, _ W a y, _ a n d, _ d i s c o v e r e d, _ t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _ a, _ h u g e, _ n u m b e r, _ o f, _ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

comma indicates pretokenizer separation

Training Data

_ P r o o f, _ o f, _ t h e, _ M i l k y, _ W a y, _ c o n s i s t i n g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ i n, _ 1 6 1 0, _ w h e n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _ a, _ t e l e s c o p e, _ t o, _ s t u d y, _ t h e, _ M i l k y, _ W a y, _ a n d, _ d i s c o v e r e d, _ t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _ a, _ h u g e, _ n u m b e r, _ o f, _ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

Training Data

_ P r o o f, _ o f, _ t h e, _ M i l k y, _ W a y, _ c o n s i s t i n g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ i n, _ 1 6 1 0, _ w h e n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _ a, _ t e l e s c o p e, _ t o, _ s t u d y, _ t h e, _ M i l k y, _ W a y, _ a n d, _ d i s c o v e r e d, _ t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _ a, _ h u g e, _ n u m b e r, _ o f, _ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

Training Data

_ P r o o f, _ o f, _ t h e, _ M i l k y, _ W a y, _ c o n s i s t i n g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ i n, _ 1 6 1 0, _ w h e n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _ a, _ t e l e s c o p e, _ t o, _ s t u d y, _ t h e, _ M i l k y, _ W a y, _ a n d, _ d i s c o v e r e d, _ t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _ a, _ h u g e, _ n u m b e r, _ o f, _ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

Training Data

_ P r o o f , _ o f , _ t h e , _ M i l k y , _ W a y , _ c o n s i s t i n g , _ o f , _ m a n y , _ s t a r s , _ c a m e , _ i n , _ 1 6 1 0 , _ w h e n , _ G a l i l e o , _ G a l i l e i , _ u s e d , _ a , _ t e l e s c o p e , _ t o , _ s t u d y , _ t h e , _ M i l k y , _ W a y , _ a n d , _ d i s c o v e r e d , _ t h a t , _ i t , _ i s , _ c o m p o s e d , _ o f , _ a , _ h u g e , _ n u m b e r , _ o f , _ f a i n t , _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

Training Data

_ P r o o f, _ o f, _ t h e, _ M i l k y, _ W a y, _ c o n s i s t i n g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ i n, _ 1 6 1 0, _ w h e n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _ a, _ t e l e s c o p e, _ t o, _ s t u d y, _ t h e, _ M i l k y, _ W a y, _ a n d, _ d i s c o v e r e d, _ t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _ a, _ h u g e, _ n u m b e r, _ o f, _ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

Training Data

_ P r o o f , _ o f , _ t h e , _ M i l k y , _ W a y , _ c o n s i s t i n g , _ o f , _ m a n y , _ s t a r s , _ c a m e , _ i n , _ 1 6 1 0 , _ w h e n , _ G a l i l e o , _ G a l i l e i , _ u s e d , _ a , _ t e l e s c o p e , _ t o , _ s t u d y , _ t h e , _ M i l k y , _ W a y , _ a n d , _ d i s c o v e r e d , _ t h a t , _ i t , _ i s , _ c o m p o s e d , _ o f , _ a , _ h u g e , _ n u m b e r , _ o f , _ f a i n t , _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ t	12335282
t h	10067390
_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
_t h	7897058
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
_t h	7897058
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
_t h	7897058
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783

Vocabulary

_t

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _ a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _ a n d, _ d
i s c o v e r e d, _t h a
t, _ i t, _ i s, _ c o m p
o s e d, _ o f, _ a, _ h u
g e, _ n u m b e r, _ o f,
_ f a i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
_t h	7897058
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783

Vocabulary

_t

_a

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _a n d, _ d i
s c o v e r e d, _t h a t,
_ i t, _ i s, _ c o m p o s
e d, _ o f, _a, _ h u g e,
_ n u m b e r, _ o f, _ f a
i n t, _ s t a r s .

Pair counts

_ a	9319062
h e	8771183
i n	8024060
_t h	7897058
e r	6517430
a n	6315205
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783

Vocabulary

_t

_a

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _a n d, _ d i
s c o v e r e d, _t h a t,
_ i t, _ i s, _ c o m p o s
e d, _ o f, _a, _ h u g e,
_ n u m b e r, _ o f, _ f a
i n t, _ s t a r s .

Pair counts

h e	8771183
i n	8024060
_t h	7897058
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783

Vocabulary

_t
_a

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _a n d, _ d i
s c o v e r e d, _t h a t,
_ i t, _ i s, _ c o m p o s
e d, _ o f, _a, _ h u g e,
_ n u m b e r, _ o f, _ f a
i n t, _ s t a r s .

Pair counts

h e	8771183
i n	8024060
_t h	7897058
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t
_a

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _a n d, _ d i
s c o v e r e d, _t h a t,
_ i t, _ i s, _ c o m p o s
e d, _ o f, _a, _ h u g e,
_ n u m b e r, _ o f, _ f a
i n t, _ s t a r s .

Pair counts

h e	8771183
i n	8024060
_t h	7897058
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t

_a

Training Data

_ P r o o f, _ o f, _t h e,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w h
e n, _ G a l i l e o, _ G a
l i l e i, _ u s e d, _a,
_t e l e s c o p e, _t o, _
s t u d y, _t h e, _ M i l
k y, _ W a y, _a n d, _ d i
s c o v e r e d, _t h a t,
_ i t, _ i s, _ c o m p o s
e d, _ o f, _a, _ h u g e,
_ n u m b e r, _ o f, _ f a
i n t, _ s t a r s .

Pair counts

h e	8771183
i n	8024060
_t h	7897058
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t

_a

he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

h e	8771183
i n	8024060
_t h	7897058
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t
_a
he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

i n	8024060
e r	6517430
r e	6031043
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t
_a
he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

i n	8024060
r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998

Vocabulary

_t
_a
he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

i n	8024060
r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733

Vocabulary

_t
_a
he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

i n	8024060
r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733

Vocabulary

_t
_a
he

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t i n g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ i n, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a i
n t, _ s t a r s .

Pair counts

i n	8024060
r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733

Vocabulary

_t
_a
he
in

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

i n	8024060
r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ i	5209828
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733

Vocabulary

_t
_a
he
in

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733
o r	4162447
e s	4010515

Vocabulary

_t
_a
he
in

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733
o r	4162447
e s	4010515

Vocabulary

_t
_a
he
in

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e r e d, _t h a t, _
i t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733
o r	4162447
e s	4010515

Vocabulary

_t
_a
he
in
re

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e re d, _t h a t, _ i
t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

r e	6031043
_t he	5605612
e r	5279258
o n	5261131
_ o	5163783
_ s	5035505
_ w	4523998
a t	4424733
o r	4162447
e s	4010515

Vocabulary

_t
_a
he
in
re

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e re d, _t h a t, _ i
t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

_t he	5605612
o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435

Vocabulary

_t
_a
he
in
re

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e re d, _t h a t, _ i
t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

_t he	5605612
o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435

Vocabulary

_t
_a
he
in
re

Training Data

_ P r o o f, _ o f, _t he,
_ M i l k y, _ W a y, _ c o
n s i s t in g, _ o f, _ m
a n y, _ s t a r s, _ c a m
e, _ in, _ 1 6 1 0, _ w he
n, _ G a l i l e o, _ G a l
i l e i, _ u s e d, _a, _t
e l e s c o p e, _t o, _ s
t u d y, _t he, _ M i l k
y, _ W a y, _a n d, _ d i s
c o v e re d, _t h a t, _ i
t, _ i s, _ c o m p o s e
d, _ o f, _a, _ h u g e, _
n u m b e r, _ o f, _ f a
in t, _ s t a r s .

Pair counts

_t he	5605612
o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435

Vocabulary

_t
_a
he
in
re
_the

Training Data

_ P r o o f, _ o f, _the, _ M i l k y, _ W a y, _ c o n s i s t in g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ in, _ 1 6 1 0, _ w he n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _a, _t e l e s c o p e, _t o, _ s t u d y, _the, _ M i l k y, _ W a y, _a n d, _ d i s c o v e re d, _t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _a, _ h u g e, _ n u m b e r, _ o f, _ f a in t, _ s t a r s .

Pair counts

_t he	5605612
o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435

Vocabulary

_t
_a
he
in
re
_the

Training Data

_ P r o o f, _ o f, _the, _ M i l k y, _ W a y, _ c o n s i s t in g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ in, _ 1 6 1 0, _ w he n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _a, _t e l e s c o p e, _t o, _ s t u d y, _the, _ M i l k y, _ W a y, _a n d, _ d i s c o v e re d, _t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _a, _ h u g e, _ n u m b e r, _ o f, _ f a in t, _ s t a r s .

Pair counts

o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435
o r	3661288

Vocabulary

_t
_a
he
in
re
_the

Training Data

_ P r o o f, _ o f, _the, _ M i l k y, _ W a y, _ c o n s i s t in g, _ o f, _ m a n y, _ s t a r s, _ c a m e, _ in, _ 1 6 1 0, _ w he n, _ G a l i l e o, _ G a l i l e i, _ u s e d, _a, _t e l e s c o p e, _t o, _ s t u d y, _the, _ M i l k y, _ W a y, _a n d, _ d i s c o v e re d, _t h a t, _ i t, _ i s, _ c o m p o s e d, _ o f, _a, _ h u g e, _ n u m b e r, _ o f, _ f a in t, _ s t a r s .

Pair counts

o n	5261131
_ o	5163783
_ s	5035505
e r	4754849
_ w	4523998
a t	4424733
o u	3838417
_ c	3831635
n d	3811435
o r	3661288

Vocabulary

_t

_a

he

in

re

_the

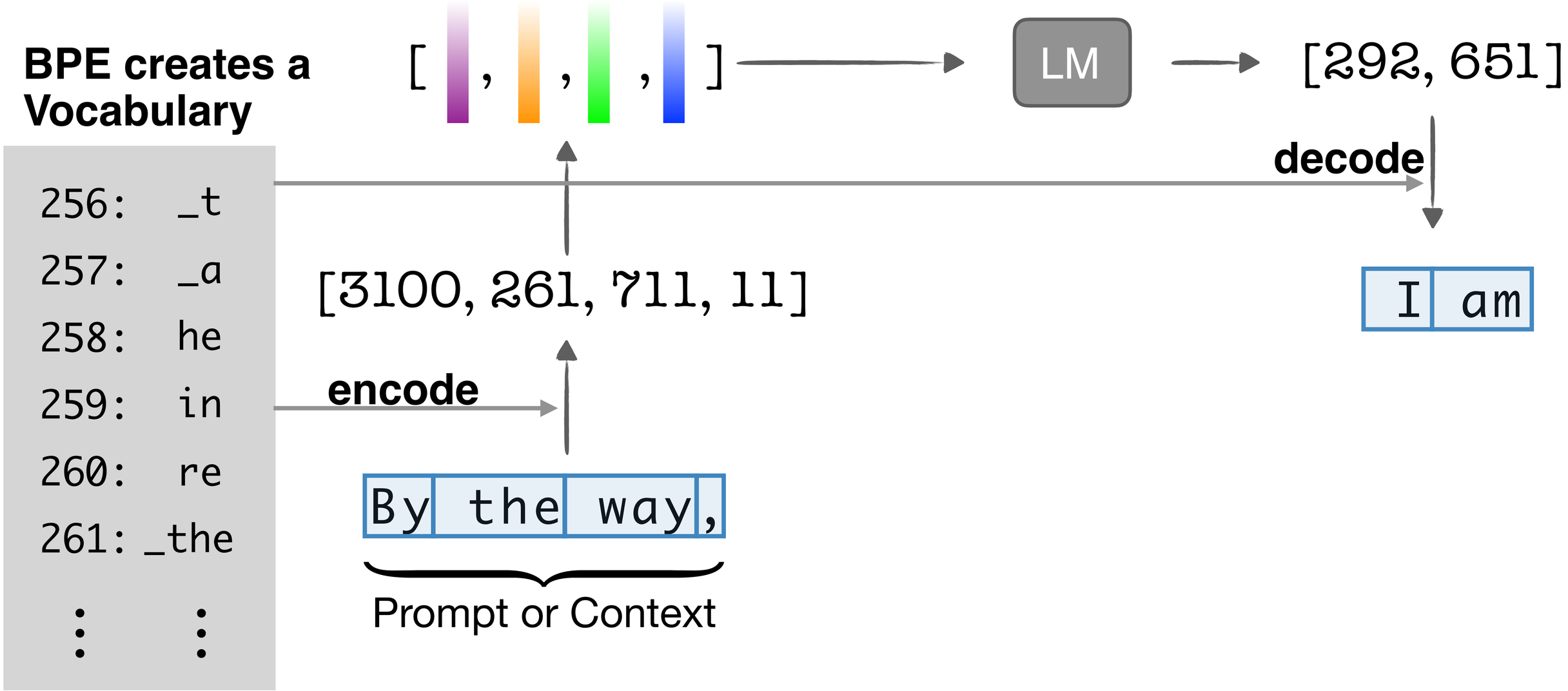
⋮

until we reach desired vocab size T

BPE creates a Vocabulary of (integer ID, pair of tokens)

```
256:  _t
257:  _a
258:  he
259:  in
260:  re
261: _the
  ⋮    ⋮
```

At inference time



Trade-off between vocab size and efficiency

GPT-2 Tokenizer with vocab size **50k** and not trained on coding data

Token count
149

```
def·fizz():\n
····for·i·in·range(1,·101):\n
········if·i·%·5·==·0·and·i·%·3·==·0:\n
············print("fizzbuzz")\n
········elif·i·%·5·==·0:\n
············print("buzz")\n
····elif·i·%·3·==·0:\n
············print("fizz")\n
····else:\n
············print(i)
```

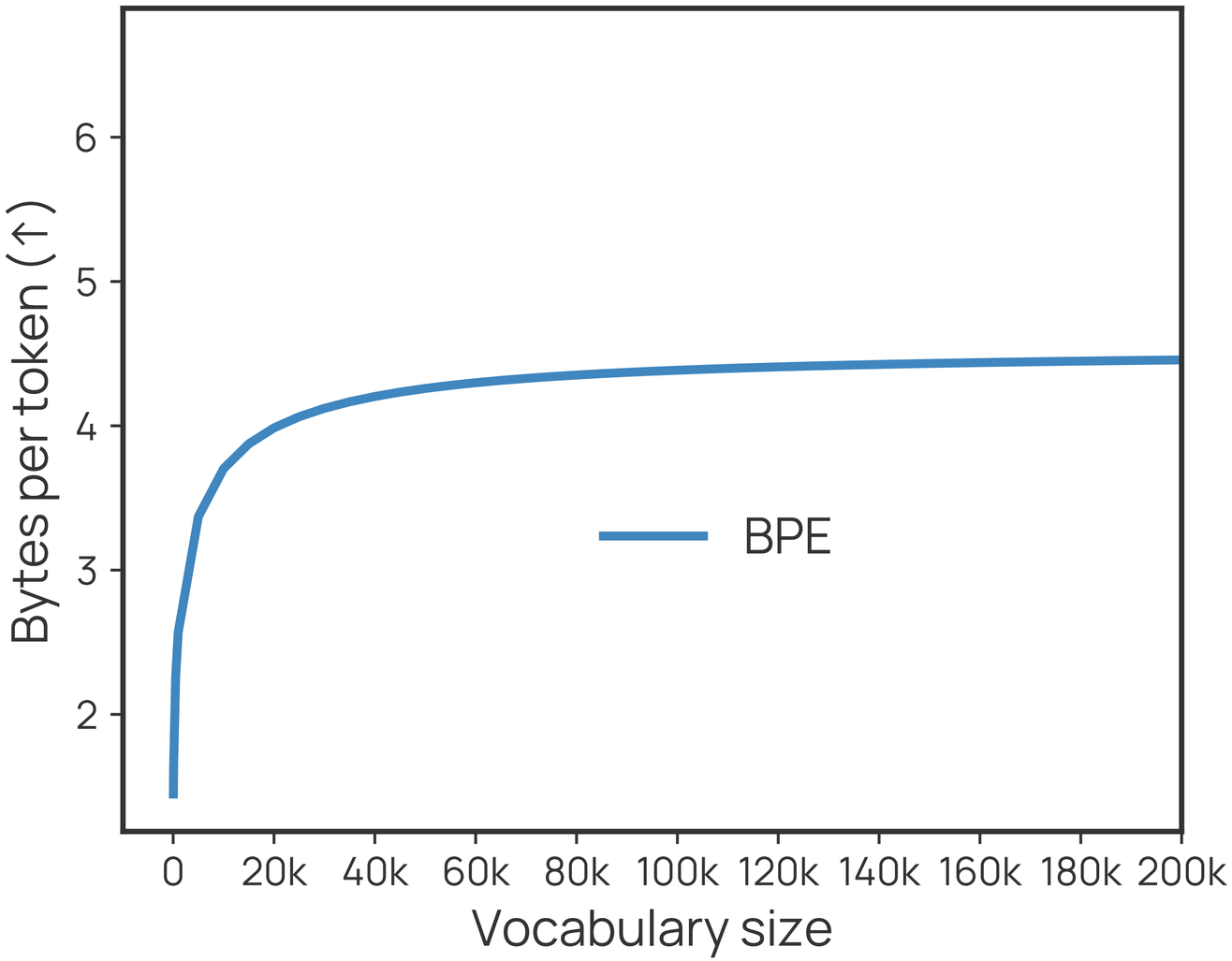
GPT-4 Tokenizer with vocab size **100k** and trained on coding data

Token count
77

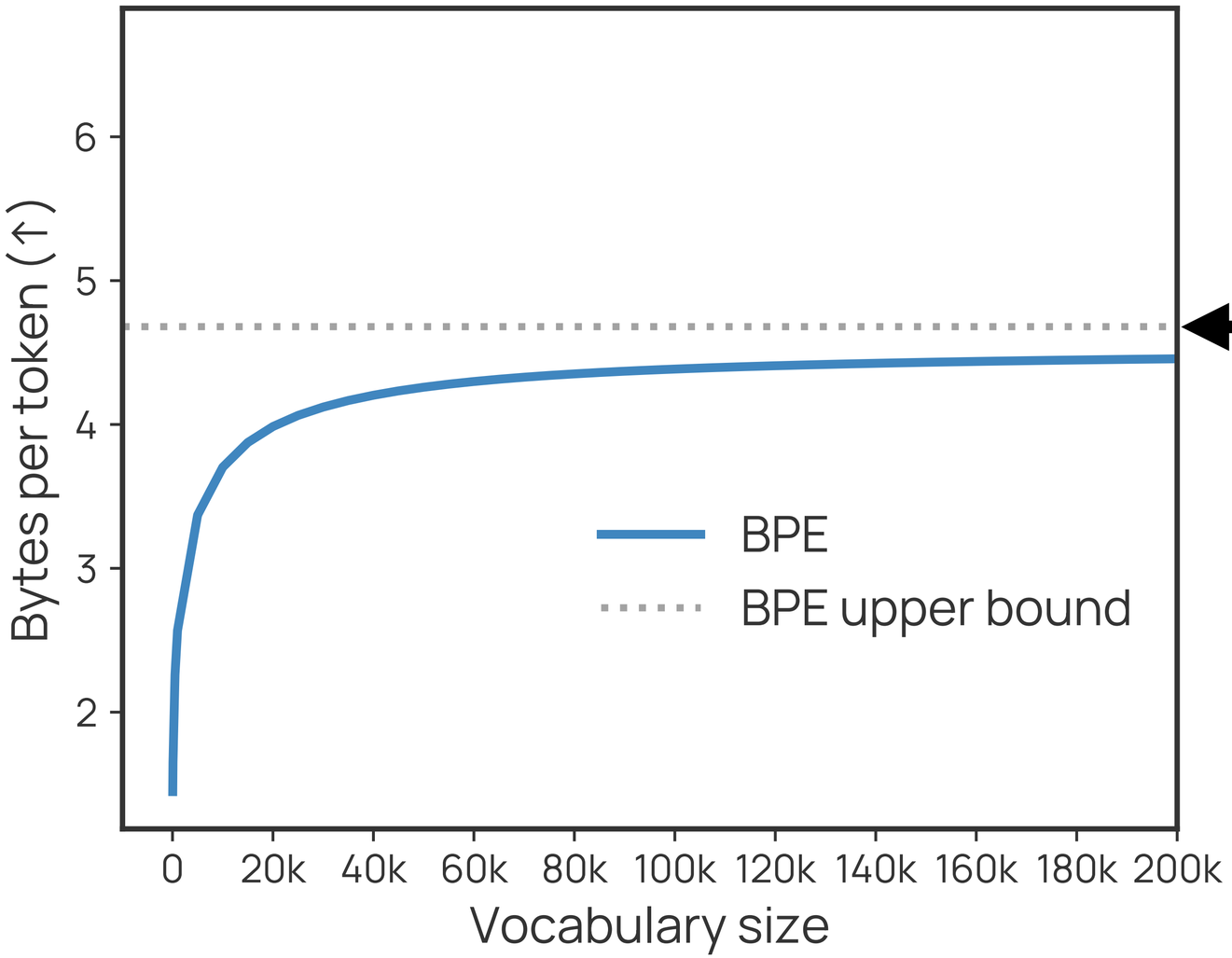
```
def·fizz():\n
····for·i·in·range(1,·101):\n
·······if·i·%·5·==·0·and·i·%·3·==·0:\n
···········print("fizzbuzz")\n
·······elif·i·%·5·==·0:\n
···········print("buzz")\n
·······elif·i·%·3·==·0:\n
···········print("fizz")\n
·······else:\n
···········print(i)
```

- Why can we not arbitrarily increase the vocab size?
- Research Question 2: How do we know what training data these closed-source tokenizers are trained on?
  - courtesy of https://github.com/openai/tiktoken

Trade-off between vocab size and efficiency



Trade-off between vocab size and efficiency



- Research Question 1: Can we beat this fundamental limit of sub-word tokenization?

Challenges with BPE tokenization (in GPT-2)

- Loss of performance for non-English languages
  - Reason: Training data contains majority of English text, which causes majority of tokens being assigned to compress English
- Loss of performance for Python
  - Lots of whitespaces for indentation requires a lot of tokens
- YAML works better than JSON

YAML is more efficiently tokenized than JSON for this tokenizer

gpt-3.5-turbo

Token count
29

Price per prompt
$0.000029

```yaml
loggingLevel: DEBUG
database:
  host: localhost
  port: 5432
  user: admin
  password: secret›
```

```
[26330, 4549, 25, 12946, 198, 12494, 512, 220, 3552, 2
5, 48522, 198, 220, 2700, 25, 220, 19642, 17, 198, 22
0, 1217, 25, 4074, 198, 220, 3636, 25, 6367, 69209]
```

gpt-3.5-turbo

Token count
46

Price per prompt
$0.000046

```json
{
  "loggingLevel": "DEBUG",
  "database": {
    "host": "localhost",
    "port": 5432,
    "user": "admin",
    "password": "secret"
  }
}
```

```
[517, 220, 330, 26330, 4549, 794, 330, 5261, 761, 220,
330, 12494, 794, 341, 262, 330, 3875, 794, 330, 8465,
761, 262, 330, 403, 794, 220, 19642, 17, 345, 262, 33
0, 882, 794, 330, 2953, 761, 262, 330, 3918, 794, 330,
21107, 702, 220, 457, 92]
```

Research Question 1:
Why do we need to limit tokens to parts of words?

SuperBPE: Space Travel for Language Models

***Alisa Liu**[♡♠] ***Jonathan Hayase**[♡]
Valentin Hofmann[♢♡] **Sewoong Oh**[♡] **Noah A. Smith**[♡♢] **Yejin Choi**[♠]
[♡]University of Washington [♠]NVIDIA [♢]Allen Institute for AI

Abstract

The assumption across nearly all language model (LM) tokenization schemes is that tokens should be *subwords*, i.e., contained within word boundaries. While providing a seemingly reasonable inductive bias, is this common practice limiting the potential of modern LMs? Whitespace is not a reliable delimiter of meaning, as evidenced by multi-word expressions (e.g., *by the way*), crosslingual variation in the number of words needed to express a concept (e.g., *spacesuit helmet* in German is *Raumanzughelm*), and languages that do not use whitespace at all (e.g., Chinese). To explore the potential of tokenization beyond subwords, we introduce a "superword" tokenizer, **SuperBPE**, which incorporates a simple pretokenization curriculum into the byte-pair encoding (BPE) algorithm to first learn subwords, then superwords that bridge whitespace. This brings dramatic improvements in encoding efficiency: when fixing the vocabulary size to 200k, SuperBPE encodes a fixed piece of text with up to 33% fewer tokens than

Research Question 1: Why do we need to limit tokens to parts of words?

- Multi-word expressions

  “*by the way,” “by accident,” “for a living,” “in the long run”*

- Some languages (e.g., Chinese) do not use **whitespace** at all!

  “*This is a really long sentence that goes on and on*” → “这是一个很长的句子，没完没了”

SuperBPE

- **Phase 1**: Run standard BPE with whitespace barrier from pretokenization until t<T
- **Phase 2**: Run BPE without whitespace barrier until T
- **Intuition**: learn the basic units of meaning (**words**) in the first phase, and then merge common word sequences (**superwords**)

SuperBPE

- Phase 1: Run BPE with whitespace barrier from pretokenization until t<T
- Phase 2: Run BPE without whitespace barrier until T
- Intuition: learn the basic units of meaning (words) in the first phase, and then merge common word sequences (superwords)

POS tag	#	Random examples
NN, IN	906	_case_of, _depend_on, _availability_of, _emphasis_on, _distinction_between
VB, DT	566	_reached_a, _discovered_the, _identify_the, _becomes_a, _issued_a
DT, NN	498	_this_month, _no_idea, _the_earth, _the_maximum, _this_stuff
IN, NN	406	_on_top, _by_accident, _in_effect, _for_lunch, _in_front
IN, DT, NN	333	_for_a_living, _by_the_way, _into_the_future, _in_the_midst
IN, DT, NN, IN	33	_at_the_time_of, _in_the_presence_of, _in_the_middle_of, _in_a_way_that

Training Data

```
Proof of the Milky Way consisting of many stars came in 1610 when Galileo Galilei used a telescope to study the Milky Way and discovered that it is composed of a huge number of faint stars.
```

- **1st phase**:
  - Run standard BPE with whitespace pretokenization until vocab size reaches some ***t<T***

Training Data

```
{Proof_of_the_Milky_Way_co
nsisting_of_many_stars_cam
e_in_, 1610,
_when_Galileo_Galilei_used
_a_telescope_to_study_the_
Milky_Way_and_discovered_t
hat_it_is_composed_of_a_hu
ge_number_of_faint_stars.}
```

- **2nd phase**:
  - Skip whitespace pretokenization
  - but can still use other pretokenization rules, e.g., numbers/non-numbers are separated in this example

comma indicates pretokenizer separation

Training Data

```
{P r o o f _ o f _ t h e _
M i l k y _ W a y _ c o n s
i s t i n g _ o f _ m a n y
_ s t a r s _ c a m e _ i
n, _ 1 6 1 0, _ w h e n _ G
a l i l e o _ G a l i l e i
_ u s e d _ a _ t e l e s c
o p e _ t o _ s t u d y _ t
h e _ M i l k y _ W a y _ a
n d _ d i s c o v e r e d _
t h a t _ i t _ i s _ c o m
p o s e d _ o f _ a _ h u g
e _ n u m b e r _ o f _ f a
i n t _ s t a r s .}
```

Split D into sequence of bytes

comma indicates pretokenizer separation

Training Data

```
{Proof _of _the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}
```

Apply tokenizer learned so far

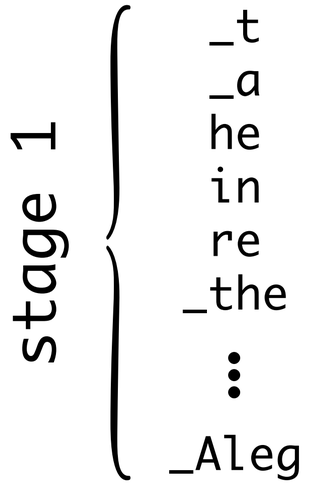
Training Data

{Proof _of _the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_of _the	517482
' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233

Vocabulary



Training Data

{Proof _of _the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_of _the	517482
' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

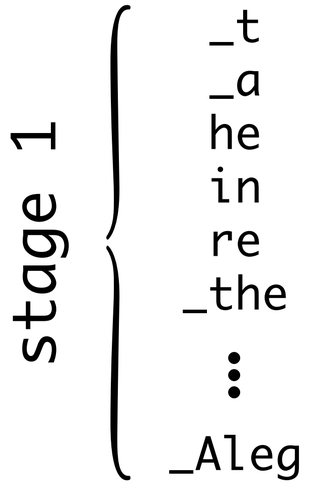
Training Data

{Proof _of_the _Milky _Way _consisting _of _many _stars _came _in_, 1 610, _when _Gal ileo _Galilei _used _a _telescope _to _study _the _Milky _Way _and _discovered _that _it _is _composed _of _a _huge _number _of _faint _stars.}

Pair counts

_of _the	517482
' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233

Vocabulary



_of _the

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

' s	456028
, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

, _and	413189
_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

, _and

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101
_to _be	147449

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

, _and

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101
_to _be	147449

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

, _and

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101
_to _be	147449

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

, _and

_in _the

Training Data

{Proof _of_the _Milky _Way
_consisting _of _many
_stars _came _in_, 1 610,
_when _Gal ileo _Galilei
_used _a _telescope _to
_study _the _Milky _Way
_and _discovered _that _it
_is _composed _of _a _huge
_number _of _faint _stars.}

Pair counts

_in _the	362529
' t	247975
. _The	232178
, _the	226412
_to _the	222524
, _but	200360
_on _the	164233
. _I	159471
? _	148101
_to _be	147449

Vocabulary

stage 1

_t
_a
he
in
re
_the
⋮
_Aleg

_of _the

' s

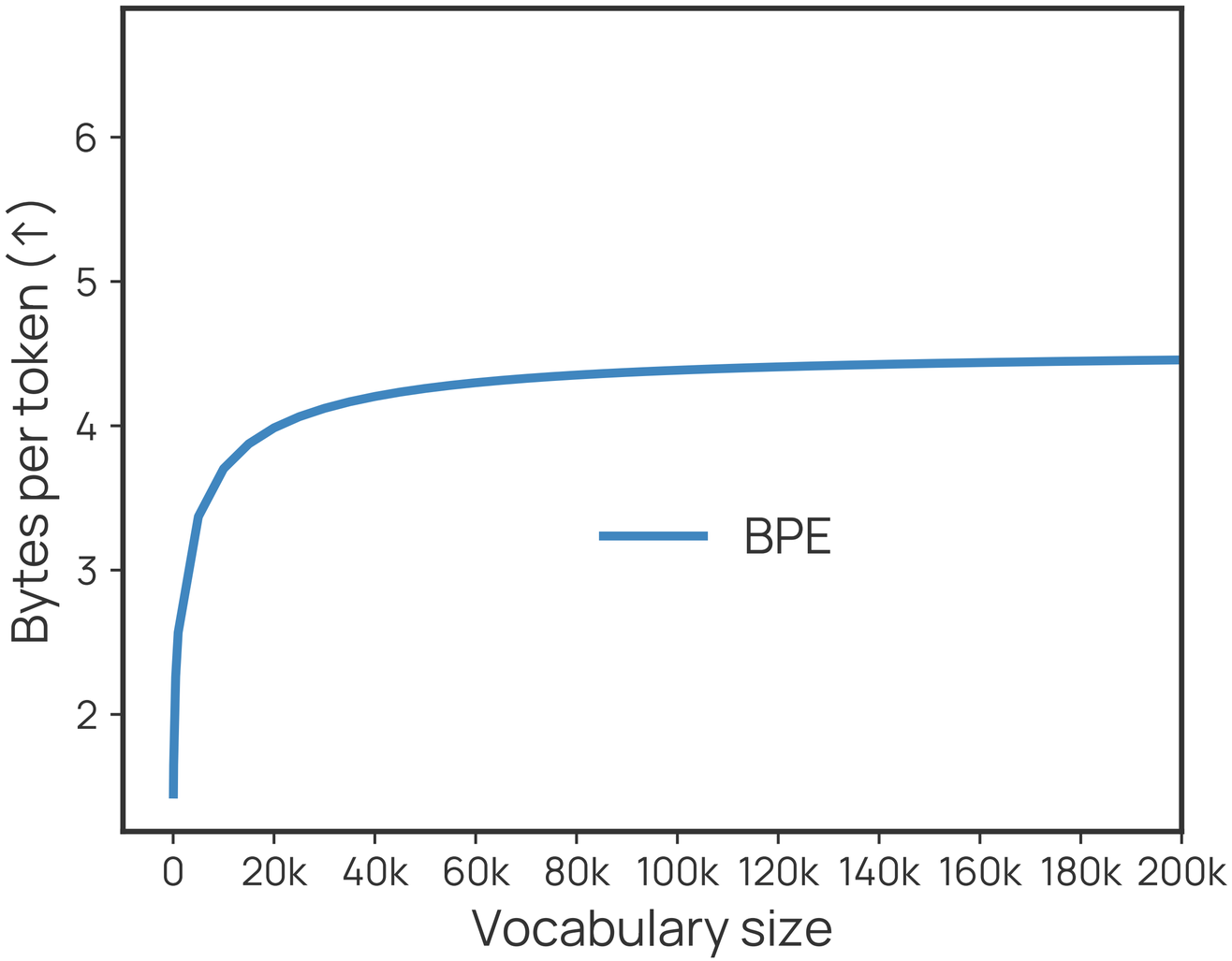
, _and

_in _the

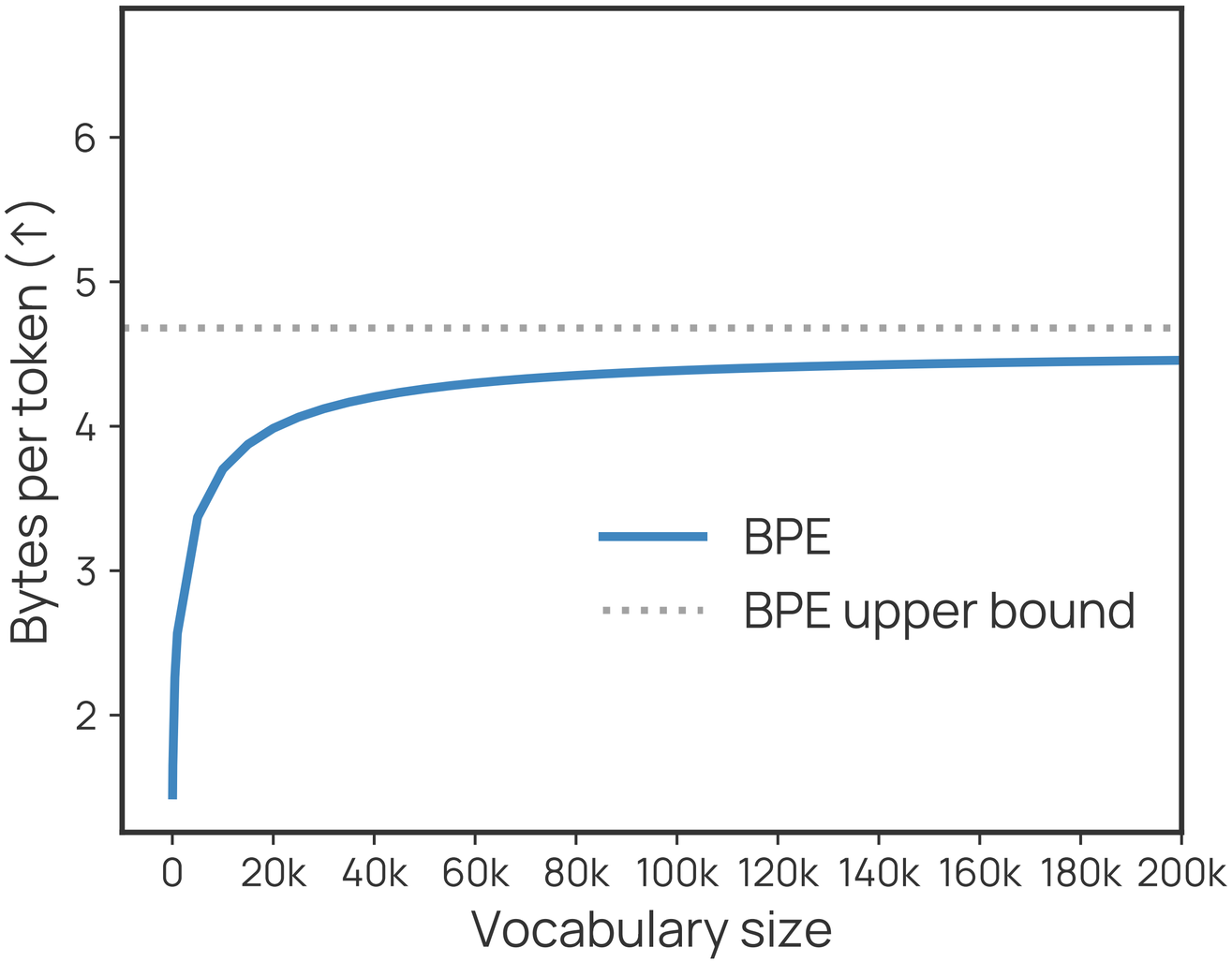
⋮

until we reach desired vocab size T

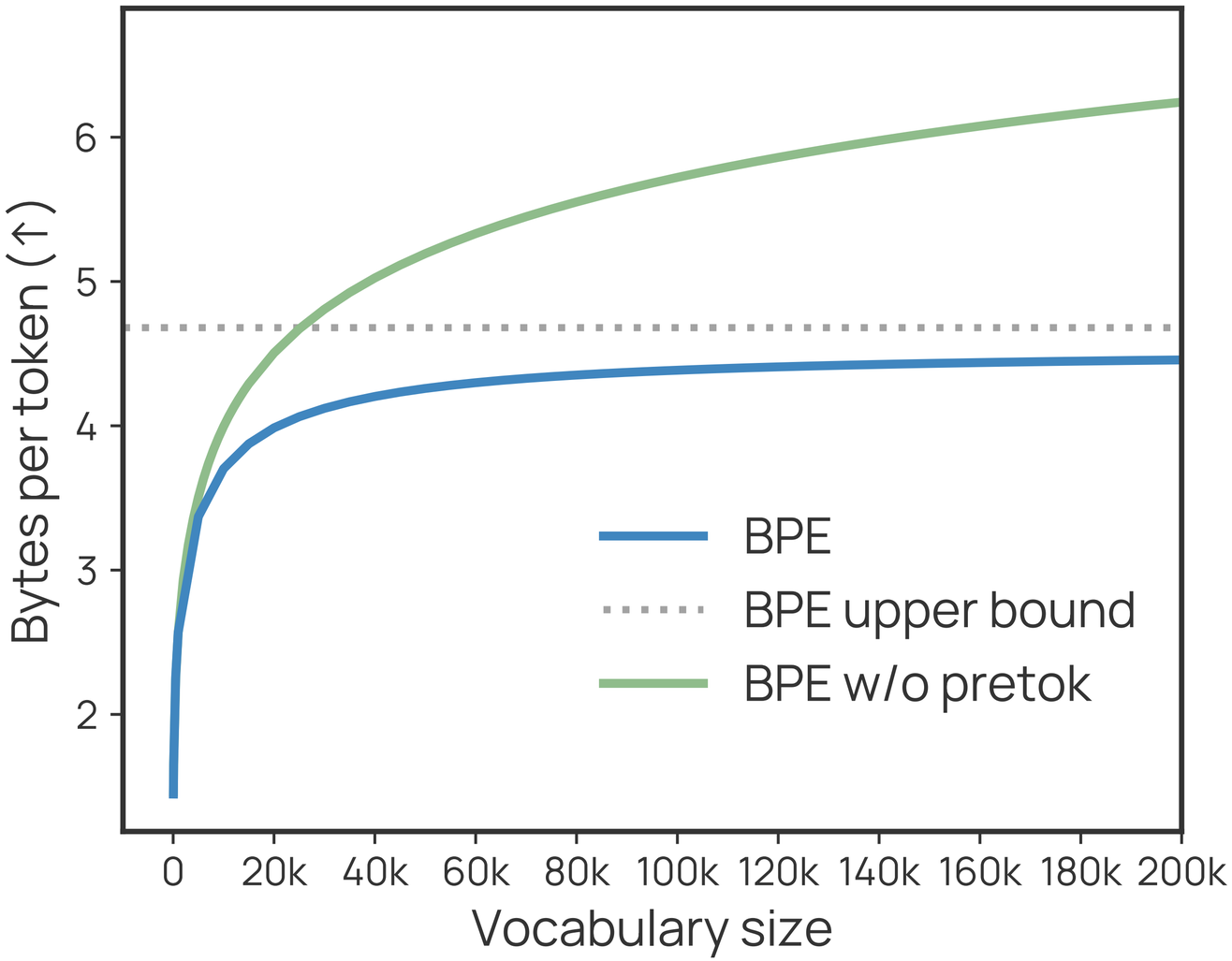
SuperBPE encodes text more efficiently



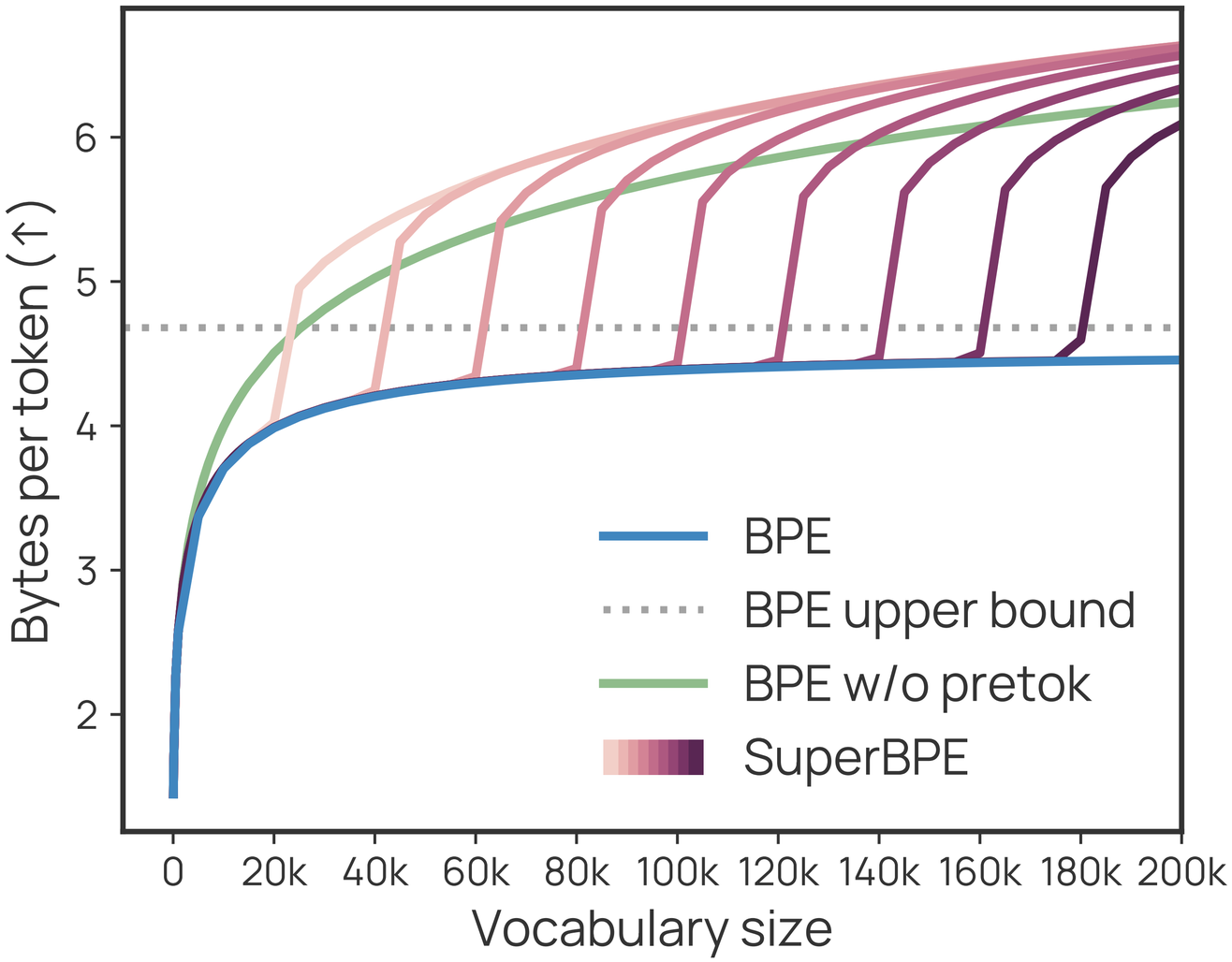
SuperBPE encodes text more efficiently



SuperBPE encodes text 35% more efficiently because many useful units are larger than words



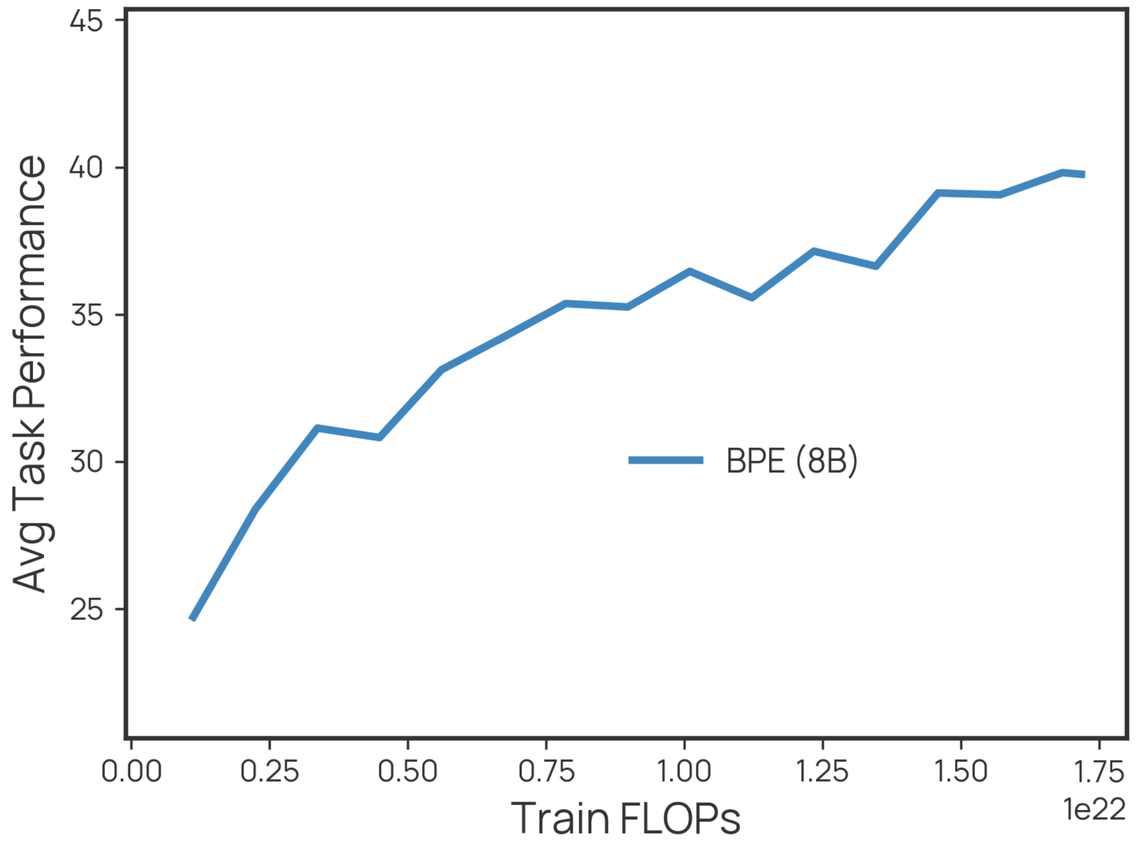
SuperBPE encodes text 35% more efficiently because many useful units are larger than words



Changing tokenizer requires pretraining LLM

Baseline: **BPE 8B** (Olmo2 @ 330B tokens)

- Tokenizer: **BPE** with 200k tokens
- Model size: **8B** parameters
- Number of tokens in training: **330B** tokens
- Evaluation
  - Average performance on 30 tasks

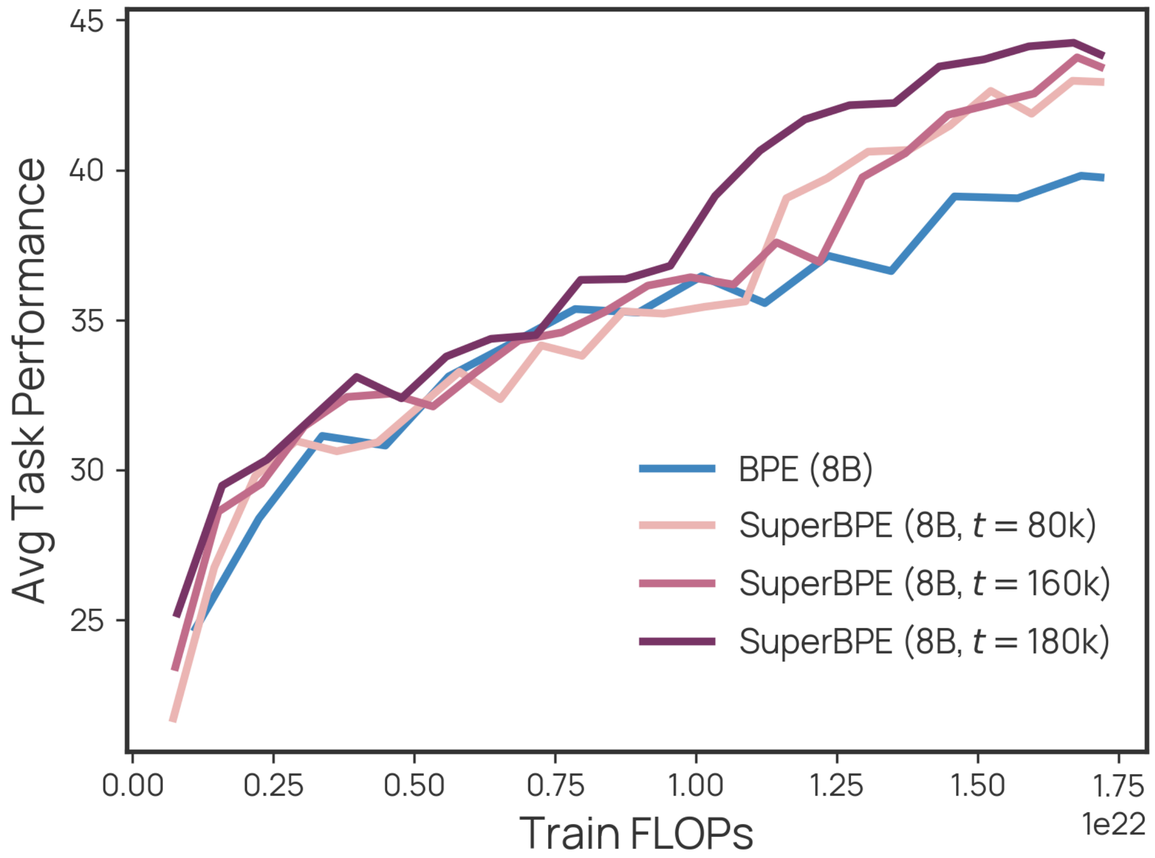


In a fair comparison, SuperBPE outperforms in 30 downstream tasks

Baseline: **BPE 8B** (Olmo2 @ 330B tokens)

SuperBPE 8B

- ✅ model size
- ✅ training compute
- ❌ inference compute (35% less)
- ❌ amount of text seen (41% more)

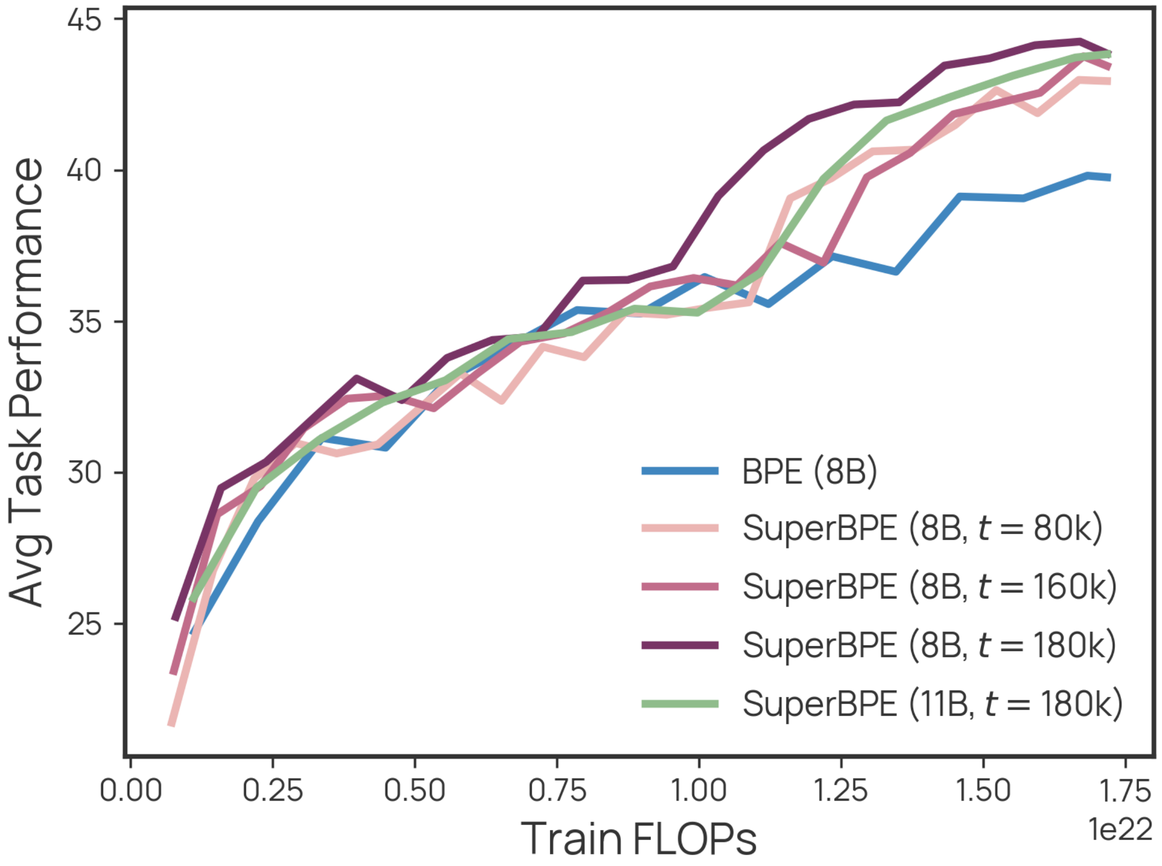


In a fair comparison, SuperBPE outperforms in 30 downstream tasks

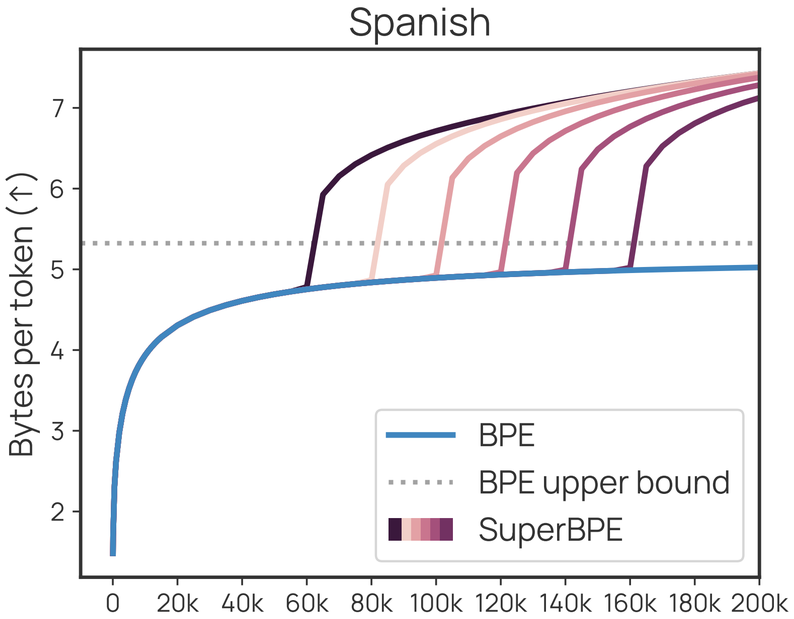
Baseline: **BPE 8B** (Olmo2 @ 330B tokens)

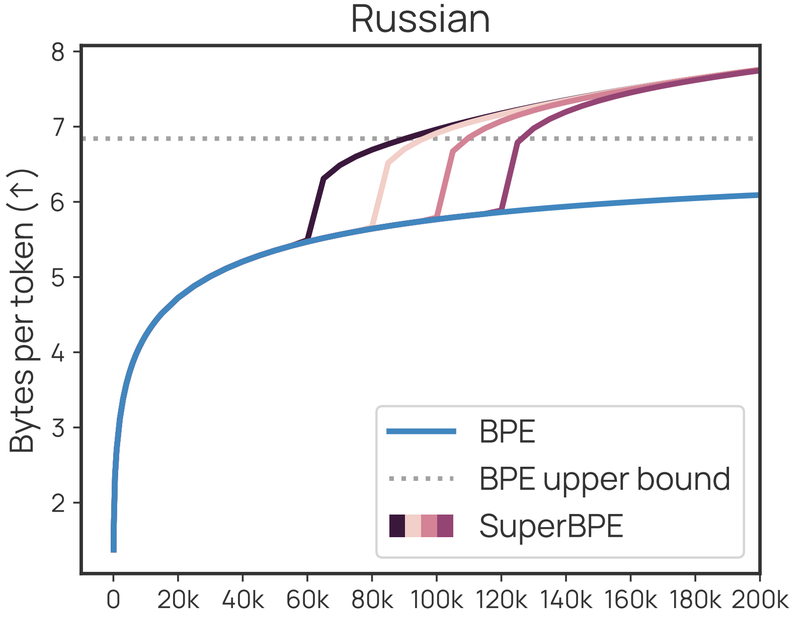
SuperBPE 11B

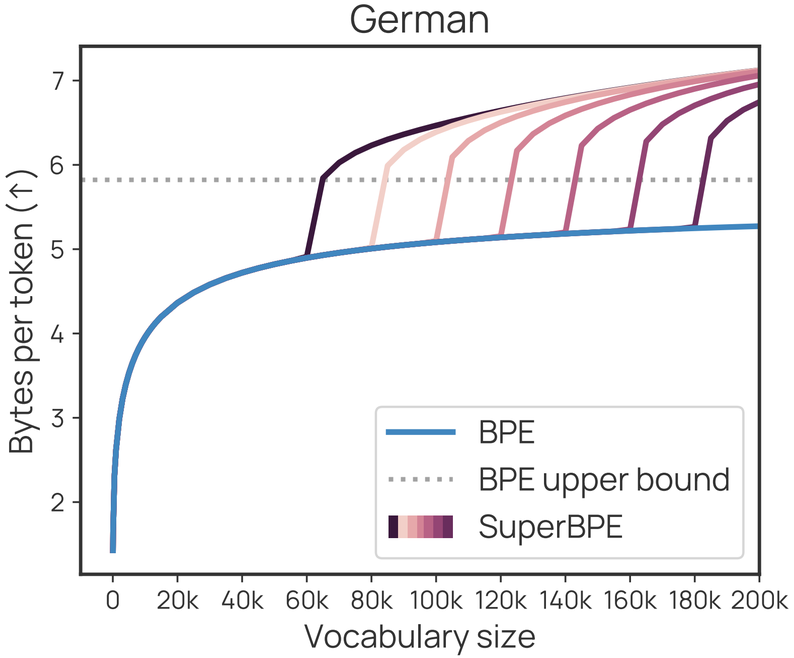
- ✗ model size (39% bigger)
- ✓ training compute
- ✓ inference compute
- ✓ amount of text seen

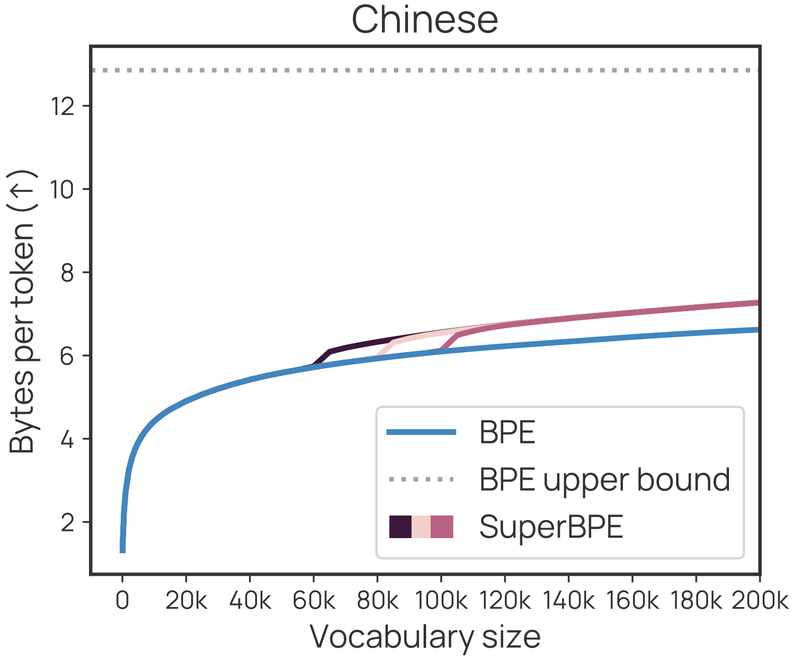


Efficiency scaling for non-English languages

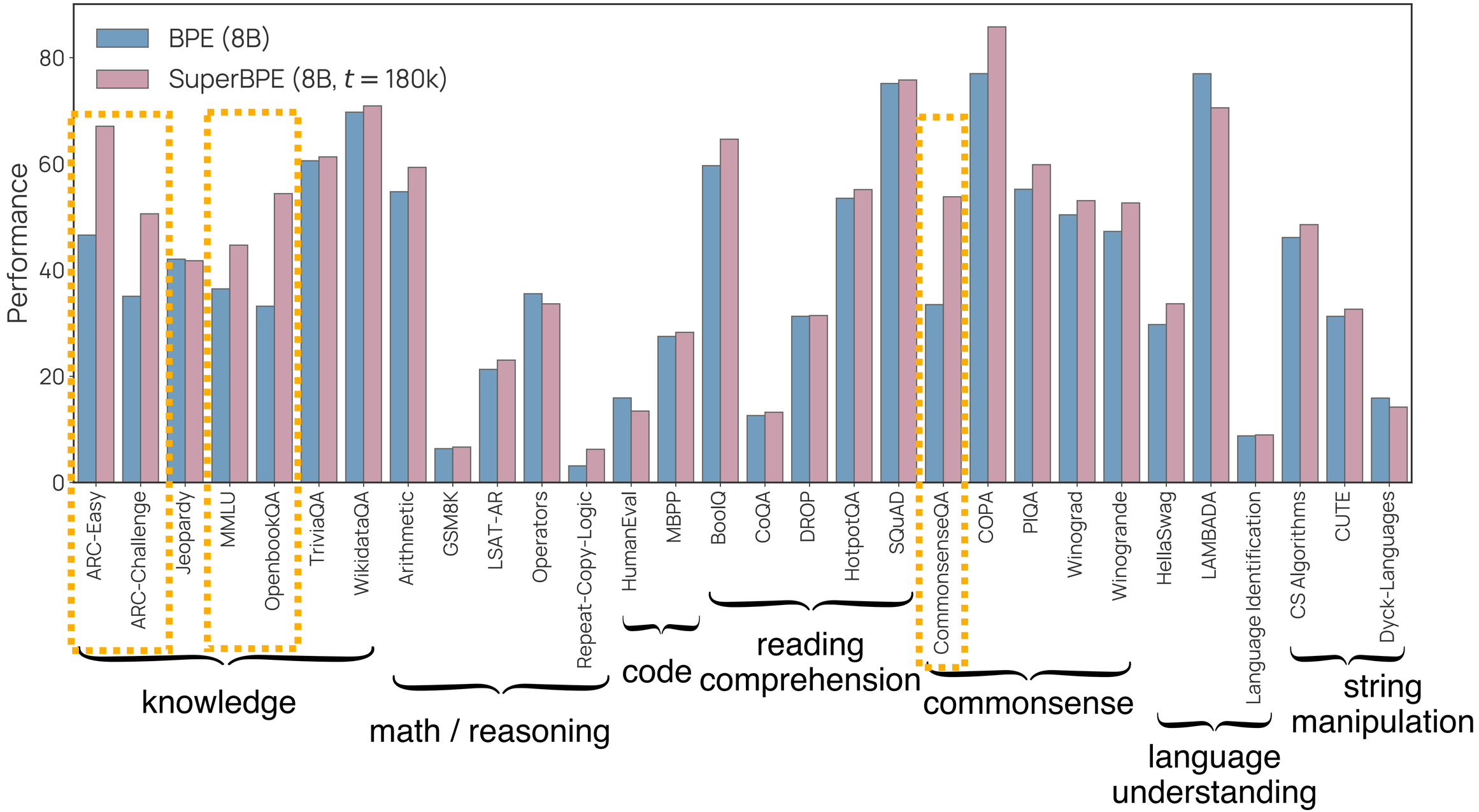








SuperBPE downstream performance



Examples of multiple choice tasks

- ARC-Challenge measures common sense
  - Q. George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?
  - (A) **dry palms**, (B) wet palms, (C) palms covered with oil, (D) palms covered with lotion

- CommonsenseQA
  - Q. What do all humans want to experience in their own home?
  - (A) **feel comfortable**, (B) work hard, (C) fall in love, (D) lay eggs, (E) live forever

Examples of multiple choice tasks

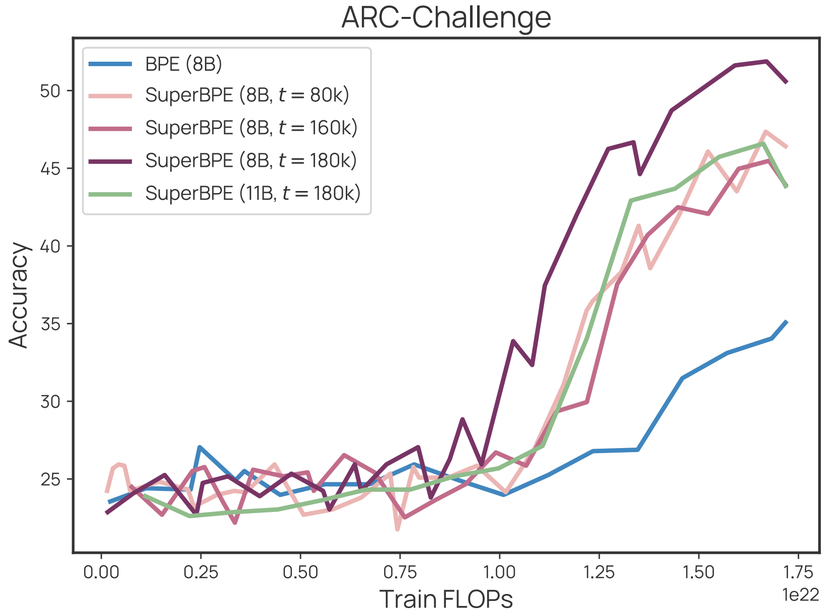
- MMLU (Massive Multitask Language Understanding)
  - Find all c in $\mathbb{Z}_3$ such that $\frac{\mathbb{Z}_3[x]}{(x^2+c)}$ is a field.
  - (A) 0 , (B) **1** , (C) 2 , (D) 3

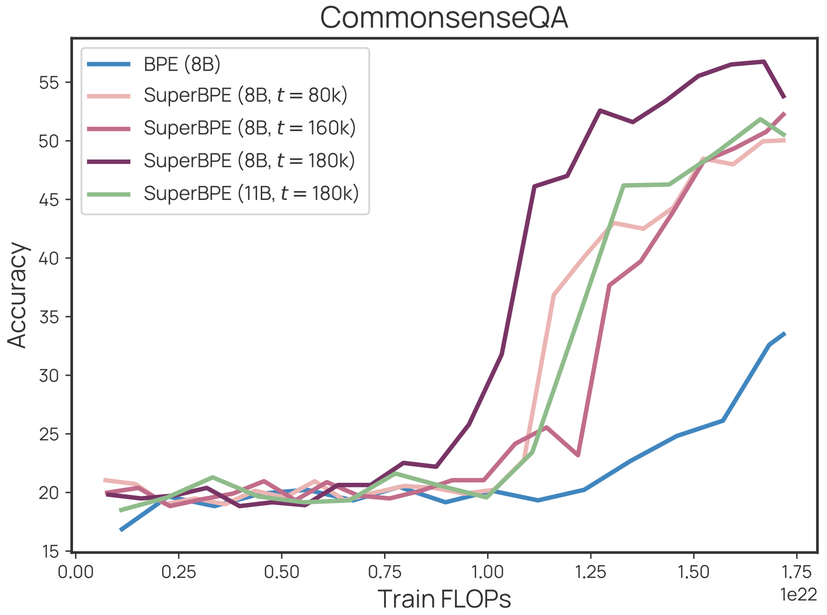
- OpenbookQA
  - Q. As a car approaches you in the night
  - (A) **the headlights become more intense**,
    (B) the headlights recede into the dark,
    (C) the headlights remain at a constant,
    (D) the headlights turn off

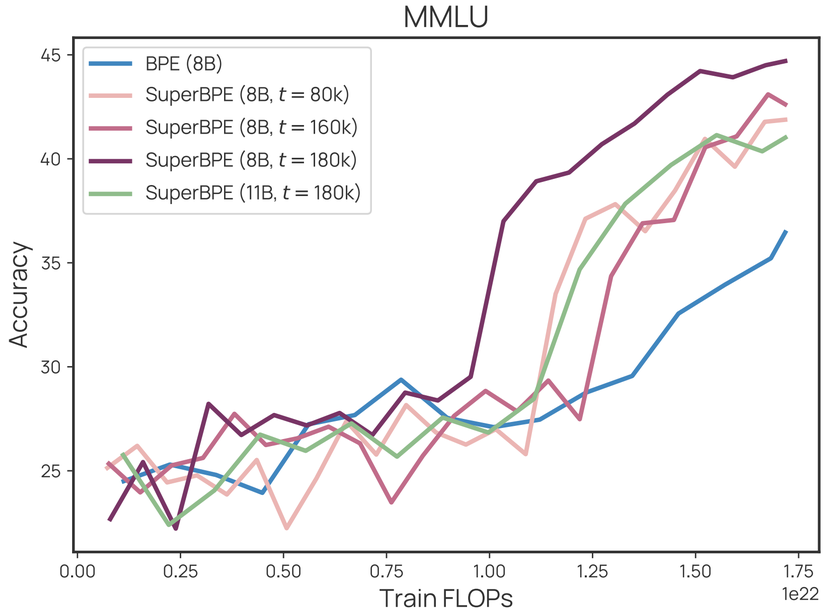
Multiple Choice

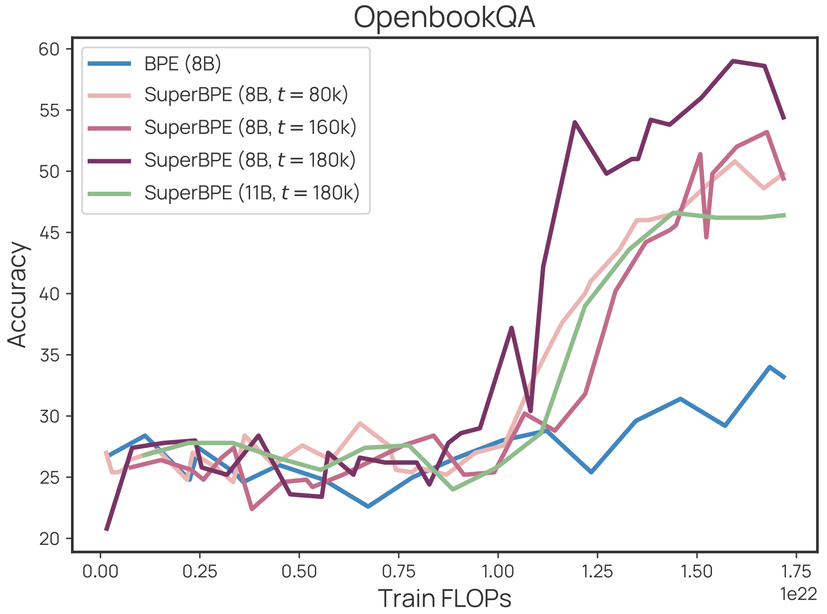
SuperBPE achieves large improvements in MC

All models begin to achieve better-than-random performance at a particular moment









Research Question 2:
How do we know what data is used to train the tokenizer?

Data Mixture Inference: What do BPE Tokenizers Reveal about their Training Data?

***Jonathan Hayase**[♡] ***Alisa Liu**[♡] **Yejin Choi**[♡♣] **Sewoong Oh**[♡] **Noah A. Smith**[♡♣]
[♡]University of Washington [♣]Allen Institute for AI
{jhayase,alisaliu}@cs.washington.edu

Abstract

The pretraining data of today's strongest language models is opaque; in particular, little is known about the proportions of various domains or languages represented. In this work, we tackle a task which we call *data mixture inference*, which aims to uncover the distributional make-up of training data. We introduce a novel attack based on a previously overlooked source of information: byte-pair encoding (BPE) tokenizers, used by the vast majority of modern language models. Our key insight is that the ordered list of merge rules learned by a BPE tokenizer naturally reveals information about the token frequencies in its training data. Given a tokenizer's merge list along with example data for each category of interest, we formulate a linear program that solves for the proportion of each category in the tokenizer's training set. In controlled experiments, we show that our attack recovers mixture ratios with high precision for tokenizers trained on known mixtures of natural languages, programming languages, and data sources. We then apply our approach to off-the-shelf tokenizers released with recent LMs. We confirm much publicly disclosed information about these models, and also make several new inferences: GPT-4O and MISTRAL NEMO's tokenizers are much more multilingual than their predecessors, training on 39% and 47% non-English language data, respectively; LLAMA 3 extends GPT-3.5's tokenizer primarily for multilingual (48%) use; GPT-3.5's and CLAUDE's tokenizers are trained on predominantly code ($\sim$60%). We hope our work sheds light on current design practices for pretraining data, and inspires continued research into data mixture inference for LMs.[1]

Data Mixture Inference

English $\mathcal{D}_{\text{En}}$

Normalize the digits, then ensure that they sum to 1.

Python $\mathcal{D}_{\text{Py}}$

```
x = logits.softmax()  # get probs
assert x.sum().item() == 1  # compare
```

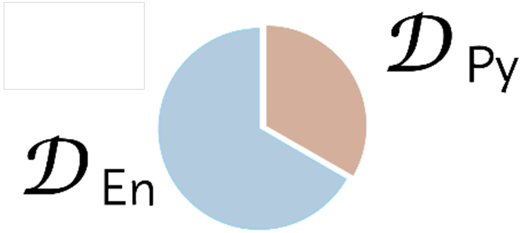
Data Mixture Inference

English $\mathcal{D}_{\text{En}}$

Normalize the digits, then ensure that they sum to 1.

Python $\mathcal{D}_{\text{Py}}$

```
x = logits.softmax()  # get probs
assert x.sum().item() == 1  # compare
```



Training data mixture

Given data, BPE learns a merge list

merge list

```
1   _ t
2  _t h
3 _th e
4   s u
5   i t
6   ( )
```

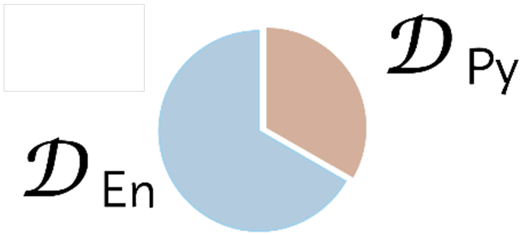
Data Mixture Inference

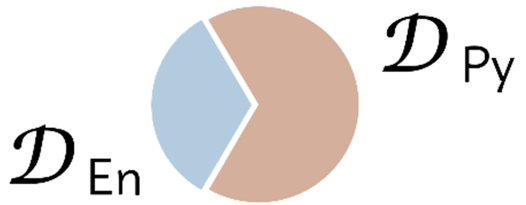
English $\mathcal{D}_{\text{En}}$

Normalize the digits, then ensure that they sum to 1.

Python $\mathcal{D}_{\text{Py}}$

```
x = logits.softmax()  # get probs
assert x.sum().item() == 1  # compare
```





Given data, BPE learns a merge list

merge list

1	_	t
2	_t	h
3	_th	e
4	s	u
5	i	t
6	(	)

1	(	)
2	i	t
3	_	t
4	s	u
5	_	#
6	_	=

The learned merge list is (very) sensitive to the mixture ratio of data distributions

Data Mixture Inference

English $\mathcal{D}_{\text{En}}$

Normalize the digits, then ensure that they sum to 1.

Python $\mathcal{D}_{\text{Py}}$

```
x = logits.softmax()  # get probs
assert x.sum().item() == 1  # compare
```

1	( )
2	i t
3	_ t
4	s u
5	_ #
6	_ =

Given a merge list, can we solve for the mixture ratio?

The learned merge list is (very) sensitive to the mixture ratio of data distributions

Data Mixture Inference

English $\mathcal{D}_{\text{En}}$

Normalize the digits, then ensure that they sum to 1.

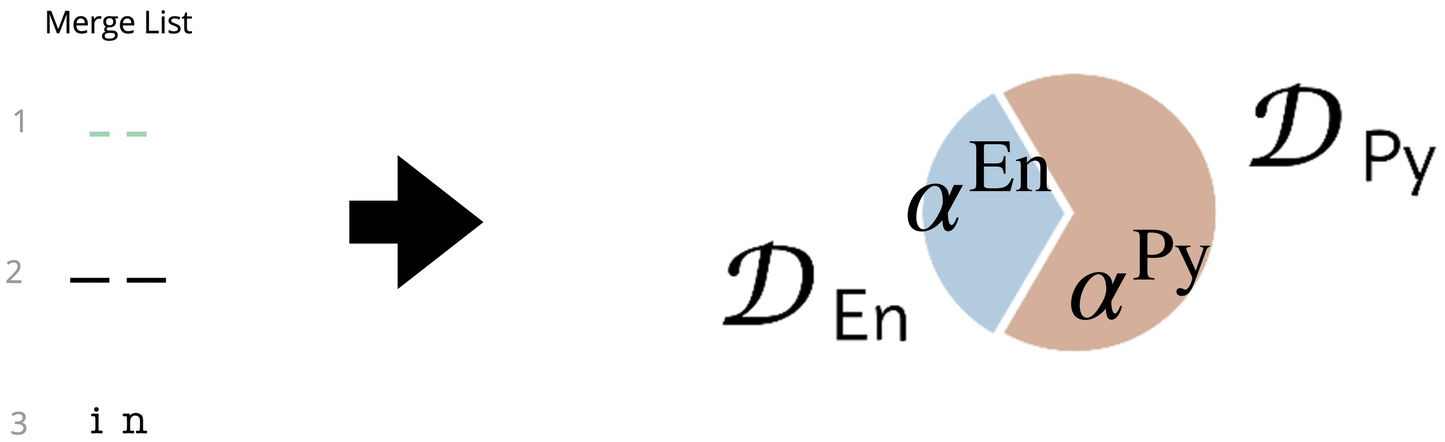
Python $\mathcal{D}_{\text{Py}}$

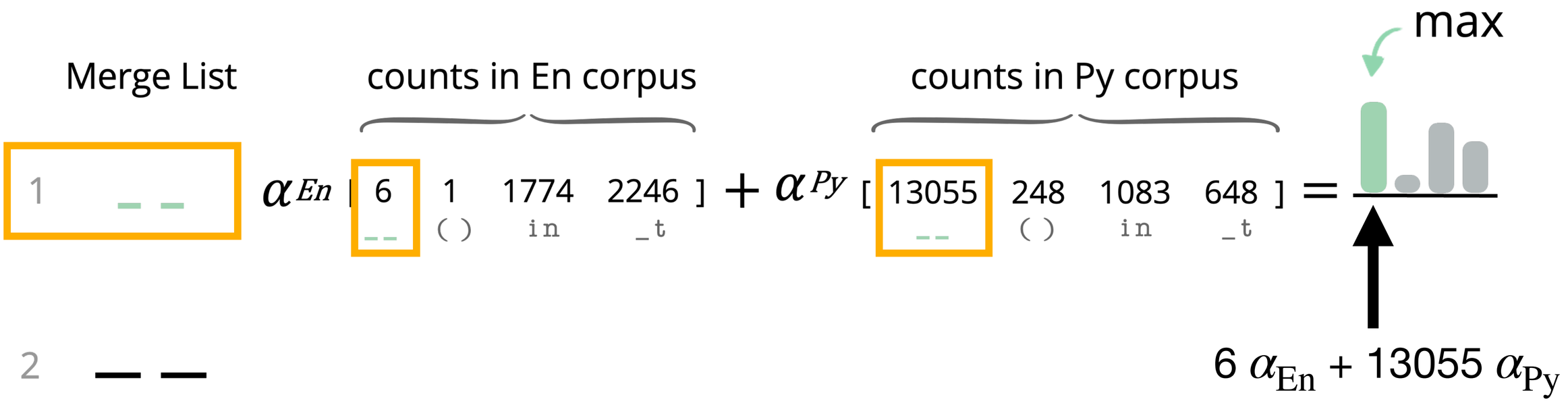
```
x = logits.softmax()  # get probs
assert x.sum().item() == 1  # compare
```

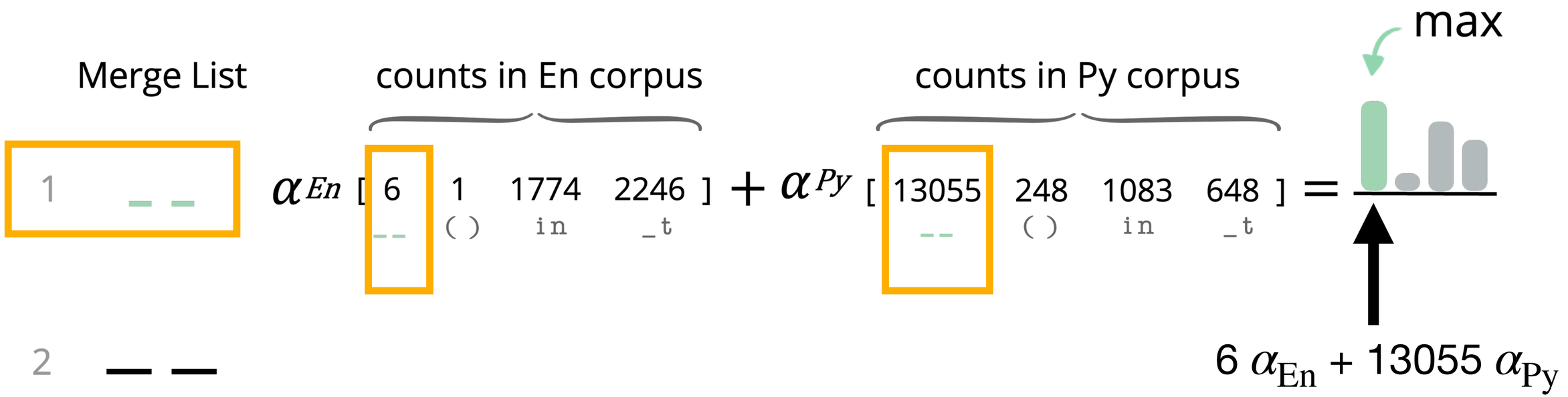
1	( )
2	i t
3	_ t
4	s u
5	_ #
6	_ =

Given a merge list, can we solve for the mixture ratio?

- Because BPE learns the most frequent pair at each step, the merge order leaks constraints on the underlying mixture of corpora.
- Mixture inference problem:
  - Given a BPE merge list
  - Find the mixture weights $\{\alpha^{\text{En}}, \alpha^{\text{Py}}, \ldots\}$ for each dataset

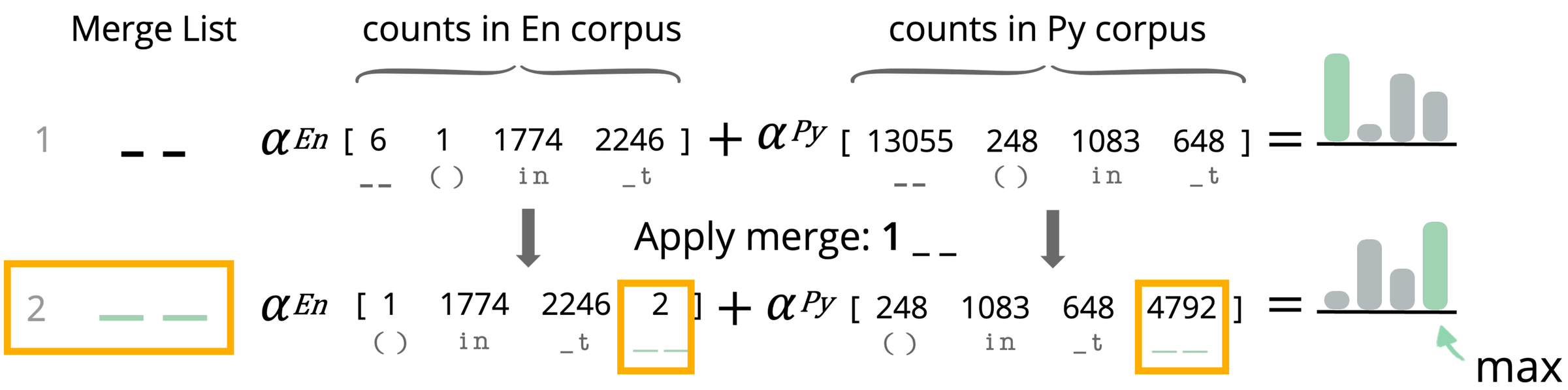



Merge List
counts in En corpus
counts in Py corpus
max
1 _ _
α^{En} [6 1 1774 2246] + α^{Py} [13055 248 1083 648] =
__ ( ) in _t
__ ( ) in _t
2 — —
3 i n
6 α_{En} + 13055 α_{Py}



Each token gives a specific linear condition that α_{En} and α_{Py} need to satisfy, for example:

$$6\,\alpha_{\mathrm{En}} + 13055\,\alpha_{\mathrm{Py}} \geq \max_{token\,!=\_\_} \left\{ \alpha_{\mathrm{En}} C^{(1)}_{\mathrm{En},token} + \alpha_{\mathrm{Py}} C^{(1)}_{\mathrm{Py},token} \right\}$$



Each token gives a specific linear condition that α_{En} and α_{Py} need to satisfy, for example:

$$2\,\alpha_{\text{En}} + 4792\,\alpha_{\text{Py}} \geq \max_{token\ !=\ \_\_\ \_\_} \left\{ \alpha_{\text{En}} C^{(2)}_{\text{En},token} + \alpha_{\text{Py}} C^{(2)}_{\text{Py},token} \right\}$$

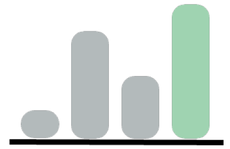
Merge List

counts in En corpus

counts in Py corpus

1 _ _ α^{En} [6 (_ _) 1 (( )) 1774 (i n) 2246 (_ t)] + α^{Py} [13055 (_ _) 248 (( )) 1083 (i n) 648 (_ t)] =

2 __ __ α^{En} [1 (( )) 1774 (i n) 2246 (_ t) 2 (__ __)] + α^{Py} [248 (( )) 1083 (i n) 648 (_ t) 4792 (__ __)] =



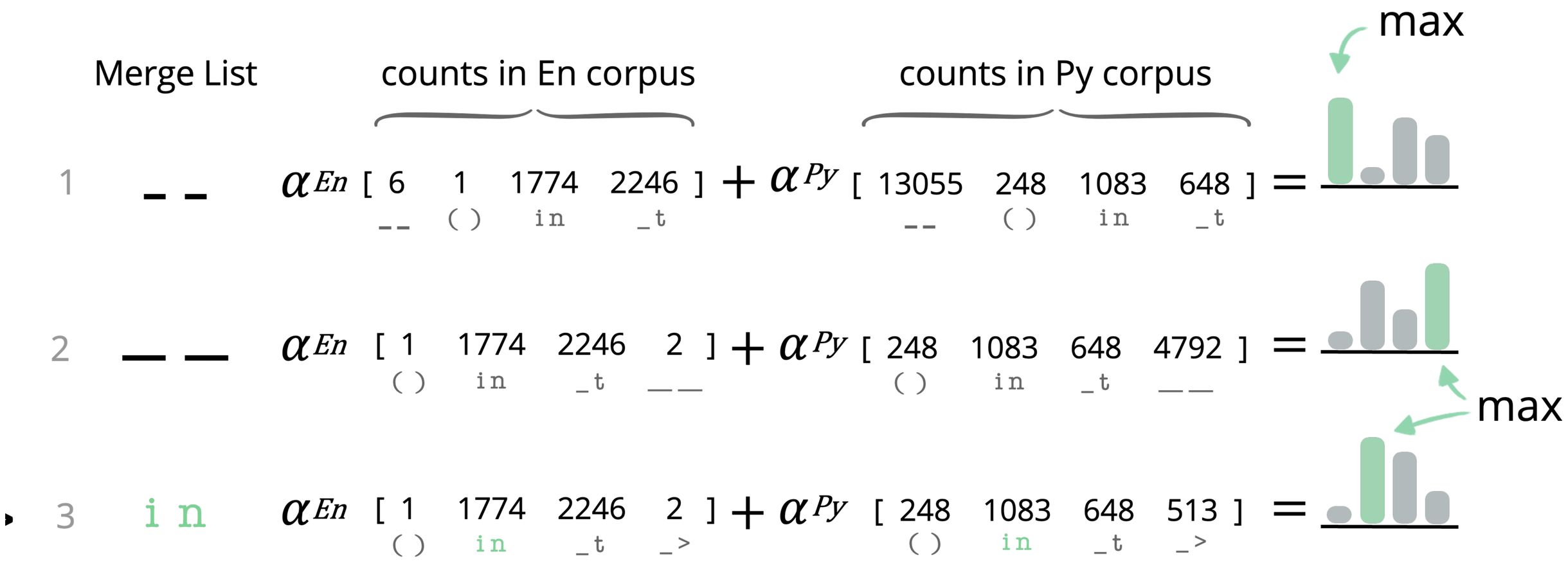
Apply merge: **2** __ __

max

3 i n α^{En} [1 (( )) 1774 (i n) 2246 (_ t) 2 (_ >)] + α^{Py} [248 (( )) 1083 (i n) 648 (_ t) 513 (_ >)] =

$$1774\,\alpha_{\mathrm{En}} + 1083\,\alpha_{\mathrm{Py}} \geq \max_{token\ !=\ in} \left\{ \alpha_{\mathrm{En}} C^{(3)}_{\mathrm{En},token} + \alpha_{\mathrm{Py}} C^{(3)}_{\mathrm{Py},token} \right\}$$

Data mixture inference is now a matter of finding a solution that satisfied all such constraints



At every step, the mixture ratios should give a vector with the true merge's index as the max value.

$$\sum_{i=1}^{n} \alpha_i c_{i,m^{(t)}}^{(t)} \geq \sum_{i=1}^{n} \alpha_i c_{i,p}^{(t)} \quad \text{for all } p \neq m^{(t)}$$

Controlled Experiments

Evaluate attack on tokenizers trained with known mixtures!

- **Natural languages** (112) from Oscar (web data)
- **Programming languages** (37) from raw Github data
- **Domains** (5) from RedPajama (all English) — web, books, Wiki, code, ArXiv

For $n \in \{5,10,30,112\}$, sample n categories and weights uniformly.

Sample 10G of data with the desired mixture ratio for tokenizer training. For the attack, sample 1G of data per category.

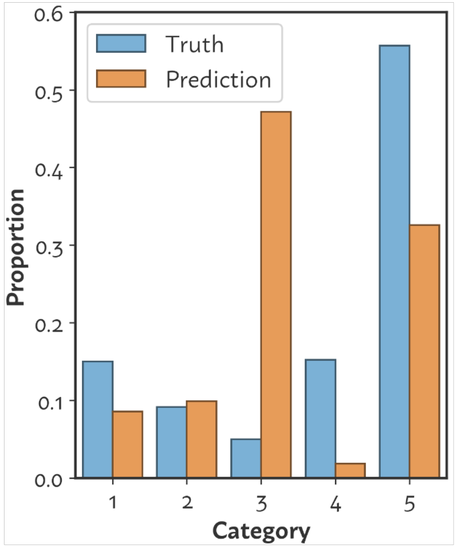
Report MSE $= \frac{1}{n}\sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i)^2$.

Results

Log_{10} MSE (↓)

number of categories

n	Random	Languages	Code	Domains
5				
10				
30				
112				

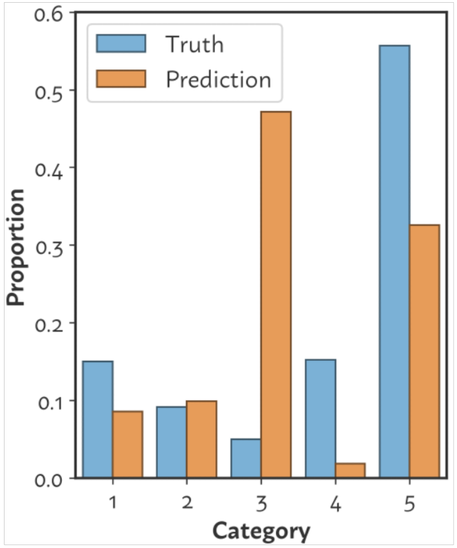


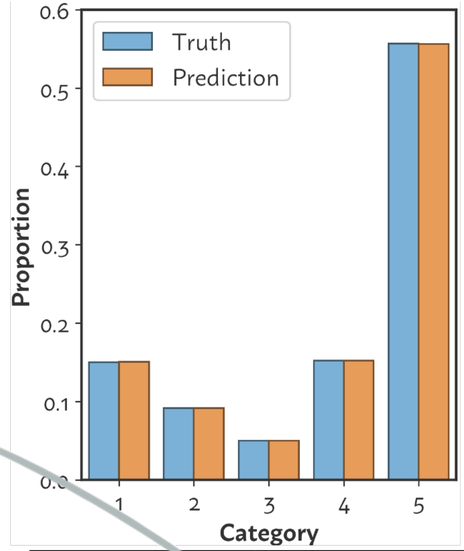
Results

Log_{10} MSE (↓)

n	random guess baseline	Languages	Code	Domains
5	-1.39			
10				
30				
112				

number of categories





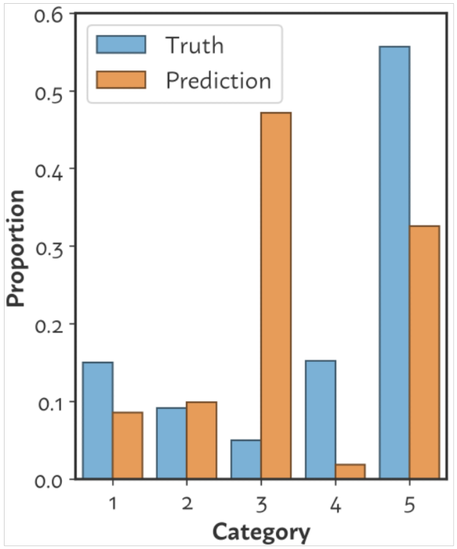
Results

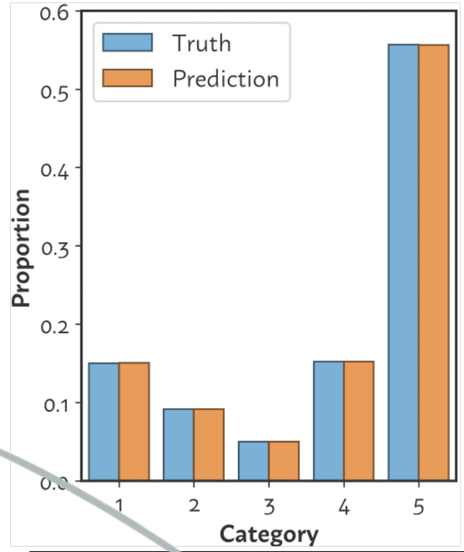
Log_{10} MSE (↓)

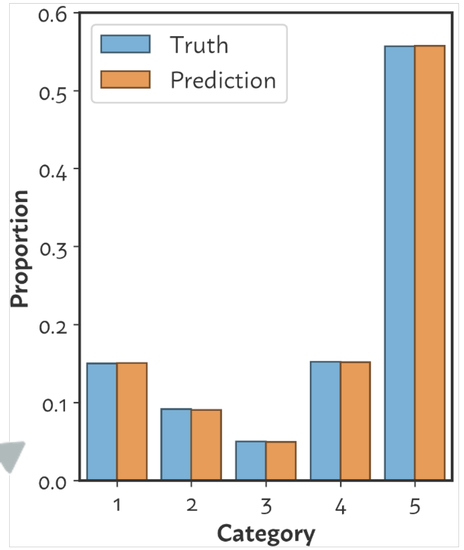
number of categories

n	Random	Languages	Code	Domains
5	-1.39	-7.30		
10				
30				
112				

Results





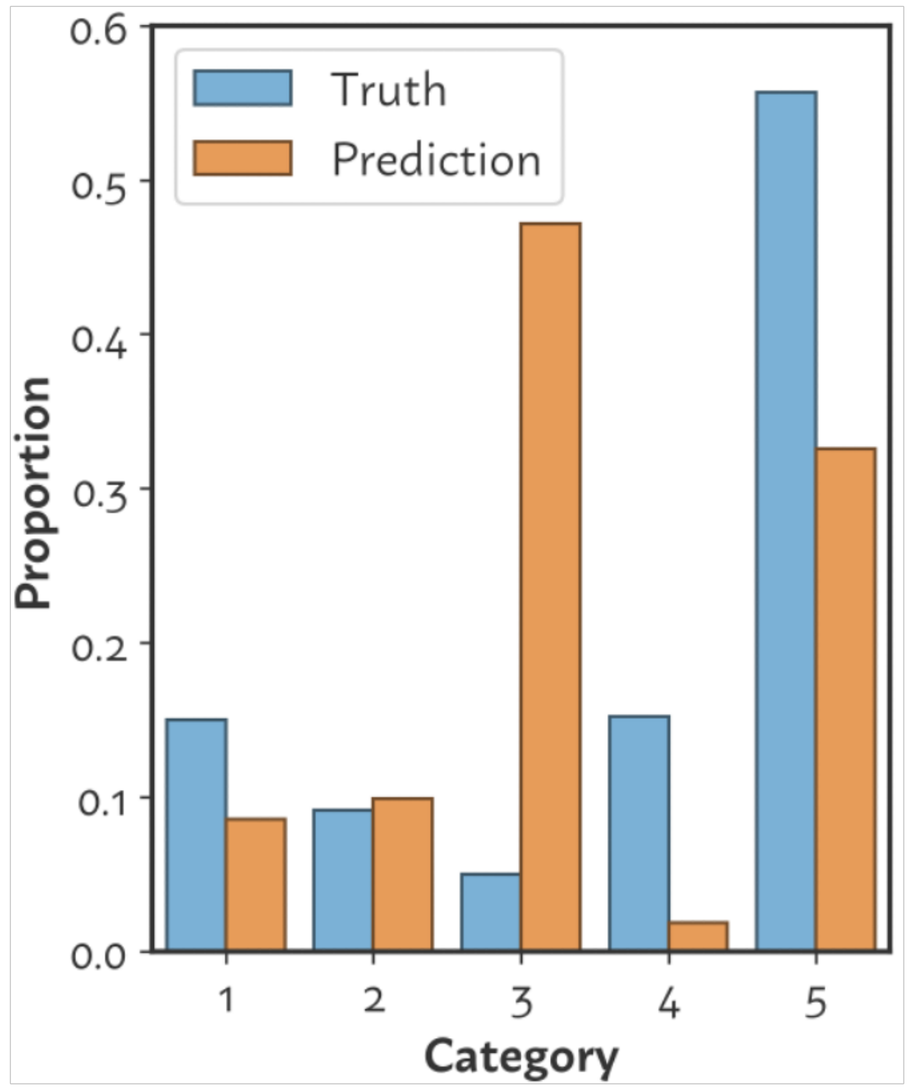


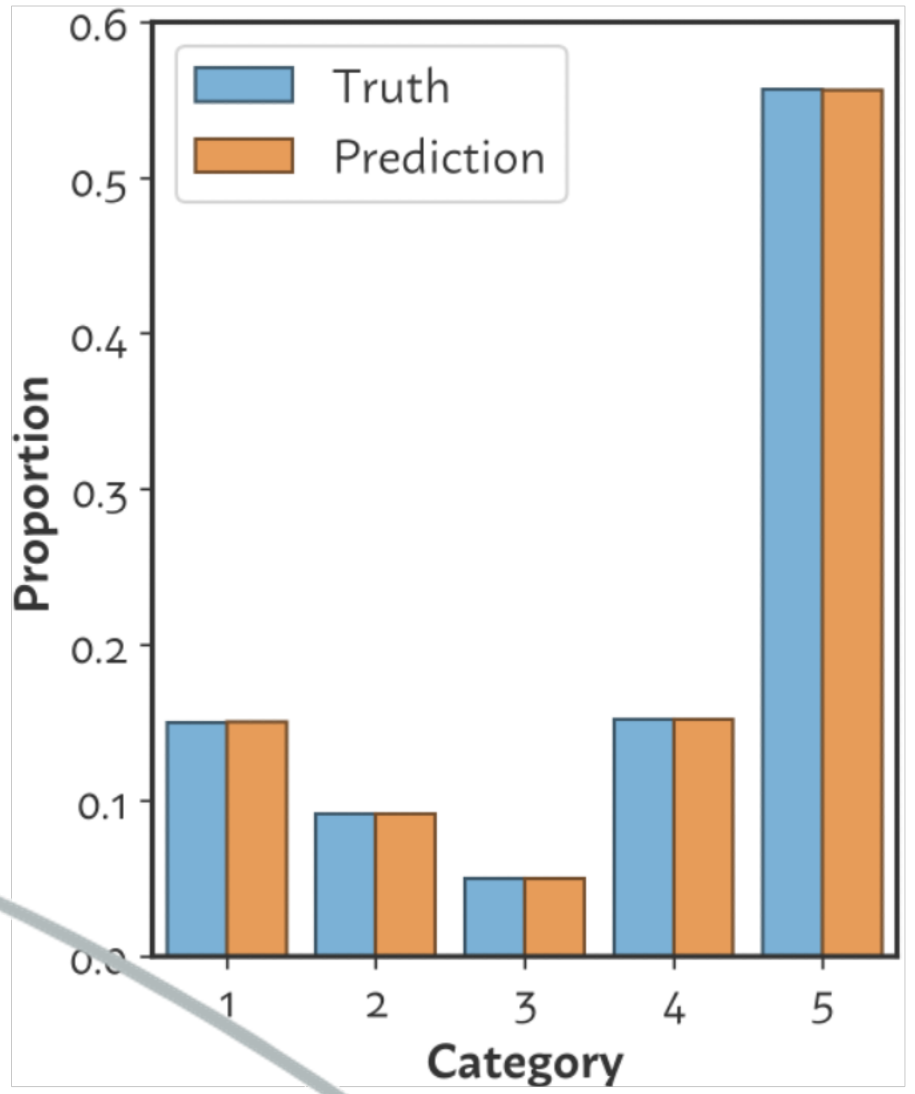
Log_{10} MSE (↓)

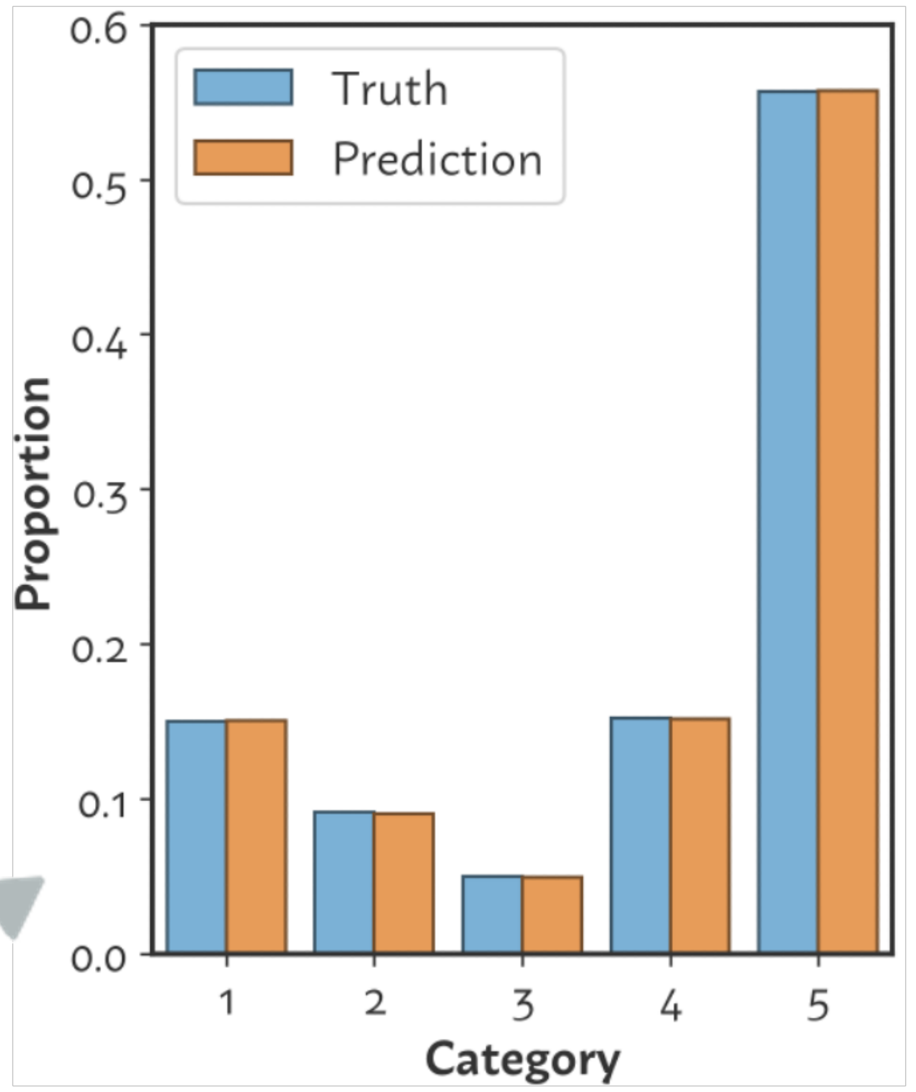
number of categories

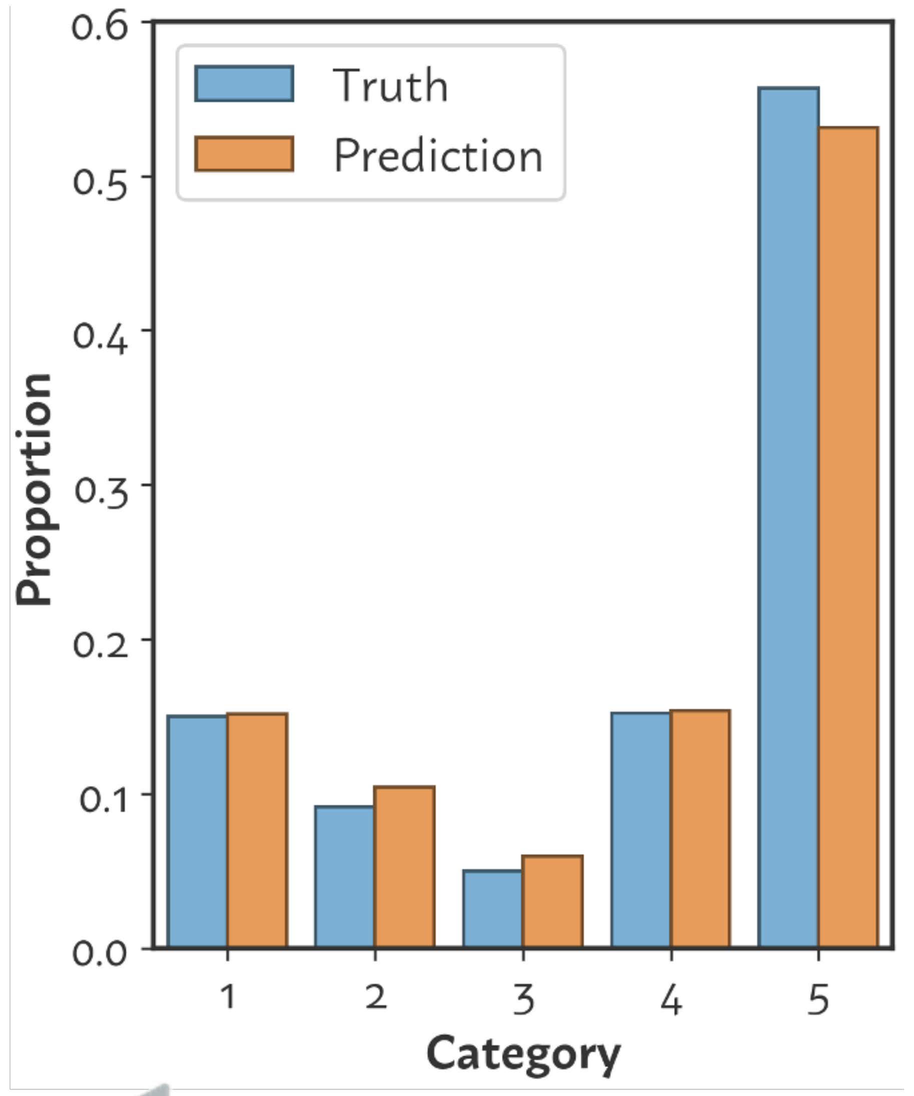
n	Random	Languages	Code	Domains
5	-1.39	-7.30	-6.46	
10				
30				
112				

Results









$\log_{10}$ MSE (↓)

n	Random	Languages	Code	Domains
5	-1.39	-7.30	-6.46	-3.74
10				
30				
112				

number of categories

Log_{10} MSE (↓)

number of categories

n	Random	Languages	Code	Domains
5	-1.39	-7.30	-6.46	-3.74
10	-1.84	-7.66	-6.30	-
30	-2.70	-7.73	-5.98	-
112	-3.82	-7.69	-	-

Our attack achieves performance 10^2 to $10^6\times$ better than random!

Commercial Tokenizers

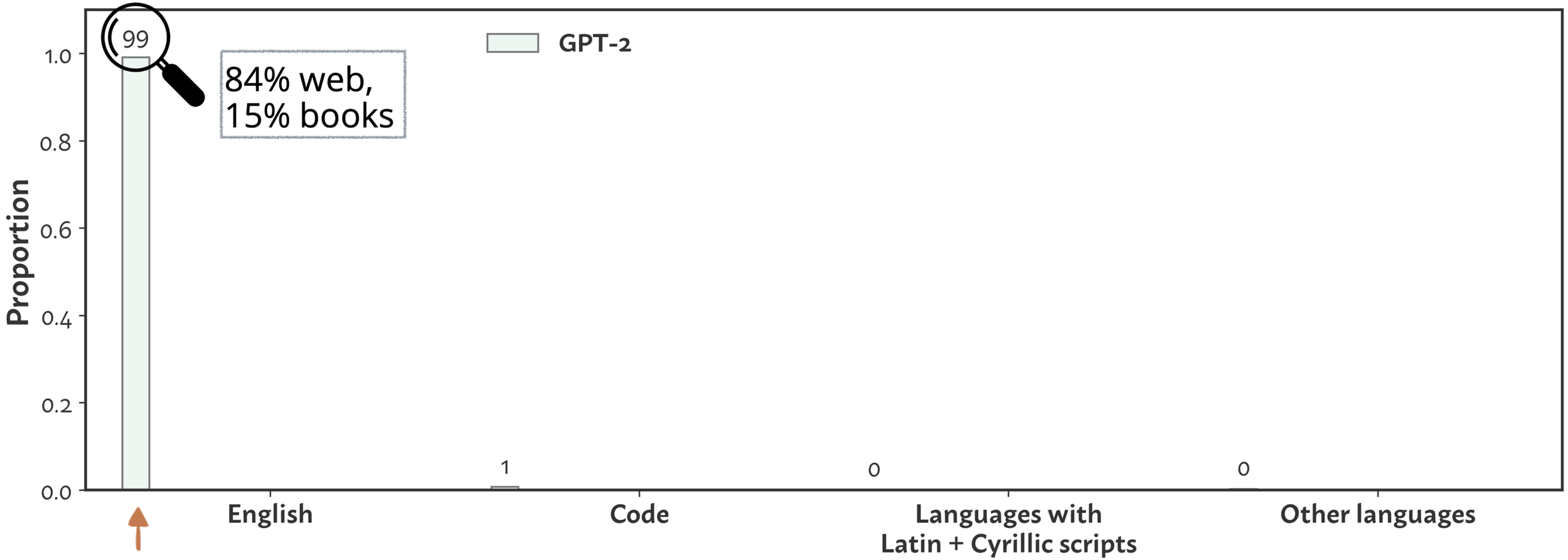
Let’s apply our attack to off-the-shelf tokenizers released with LLMs!

Total set of 116 categories: 111 languages, code, and 4 En domains.

- Split “English” into 4 En domains: web, Wikipedia, ArXiv, books.
- Combine programming languages into 1 code domain.

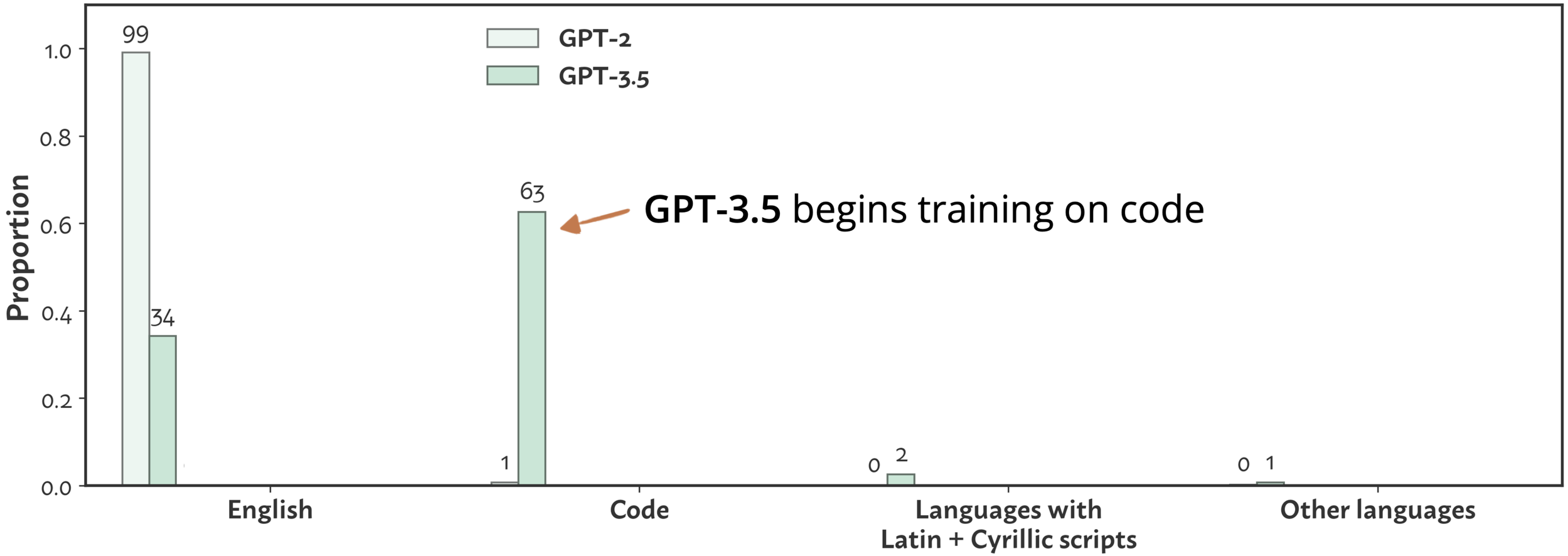
We study: GPT-2, GPT-3.5, GPT-4o, Llama, Llama 3, Mistral, Mistral-Nemo, GPT-NeoX, Gemma, Claude, Command R, …

Our Inference for LLM Tokenizers

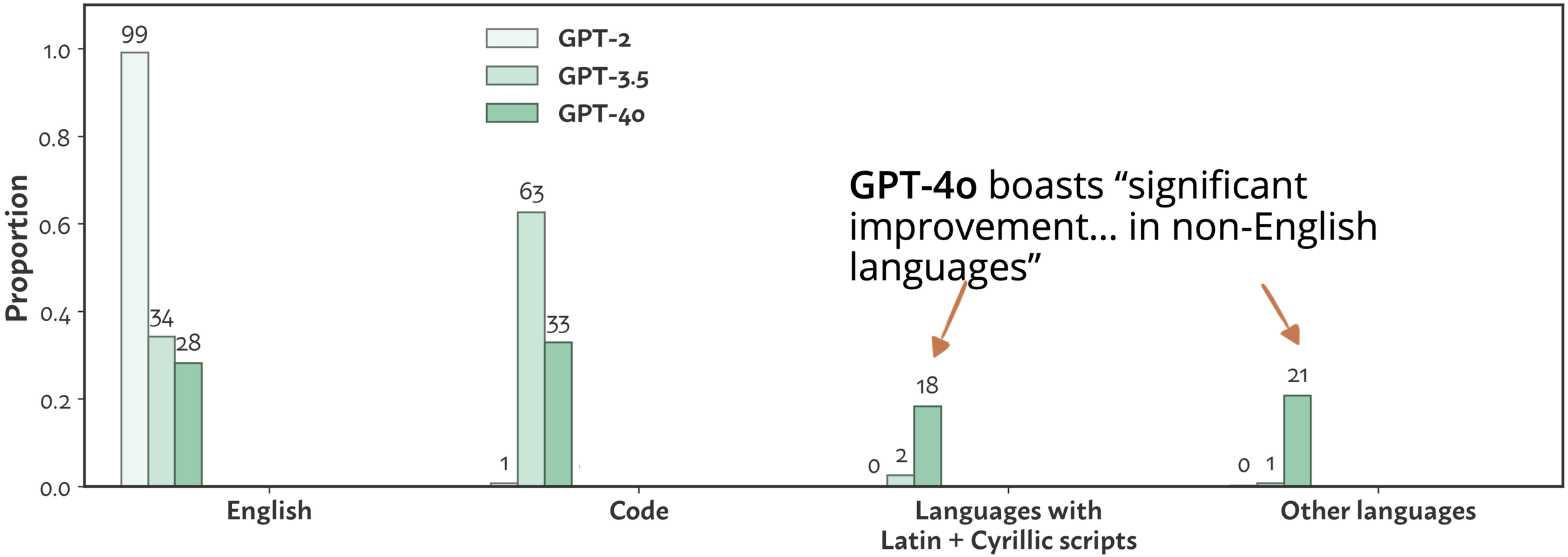


For **GPT-2**, “a filter was used to produce an English only dataset”

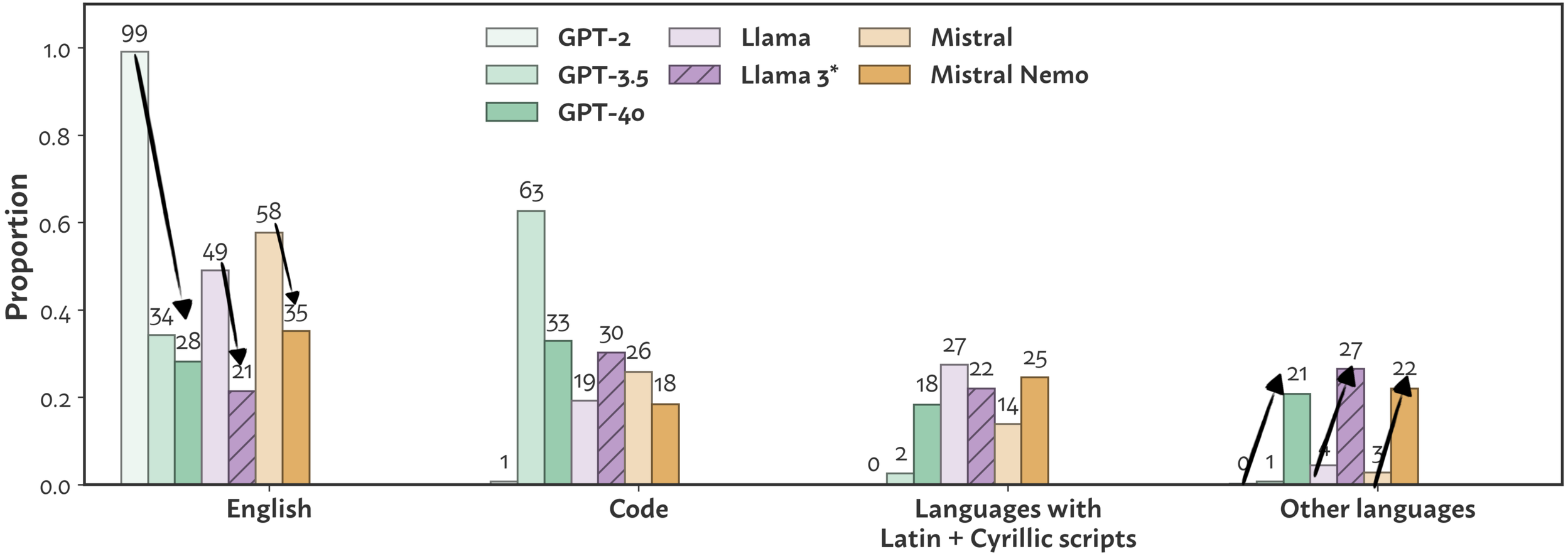
Our Inference for LLM Tokenizers



Our Inference for LLM Tokenizers

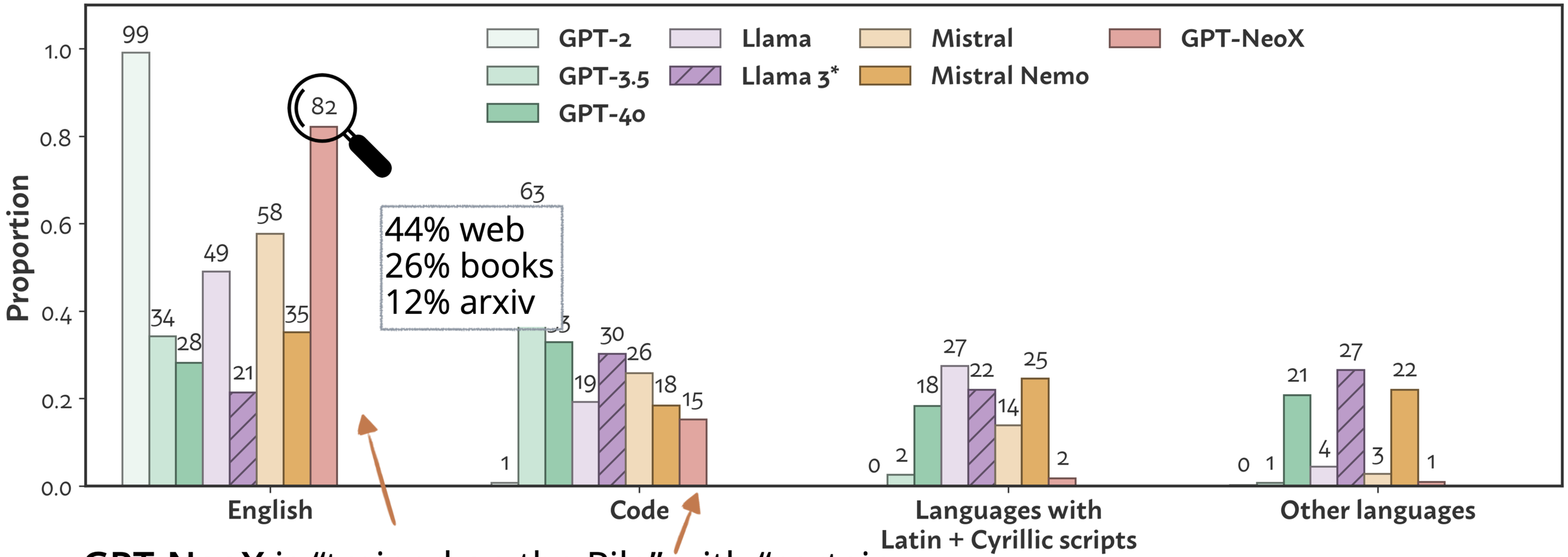


Our Inference for LLM Tokenizers



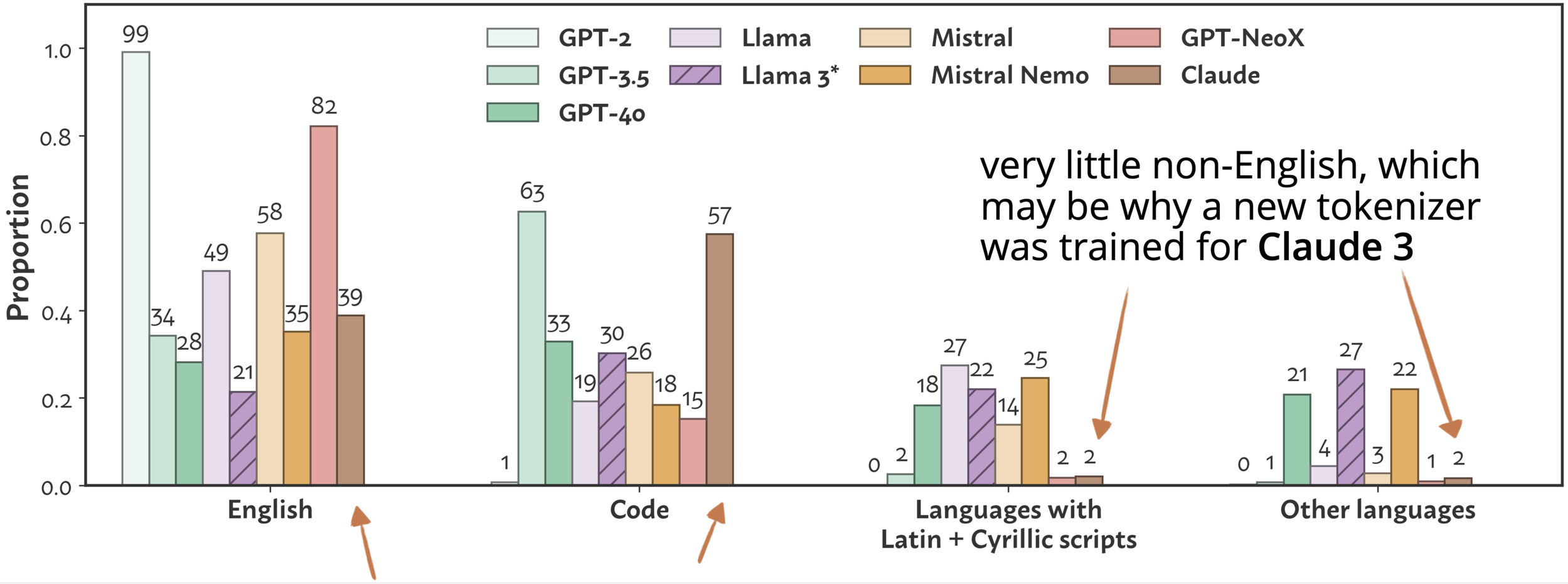
Trend: newer generations of models are more multilingual

Our Inference for LLM Tokenizers



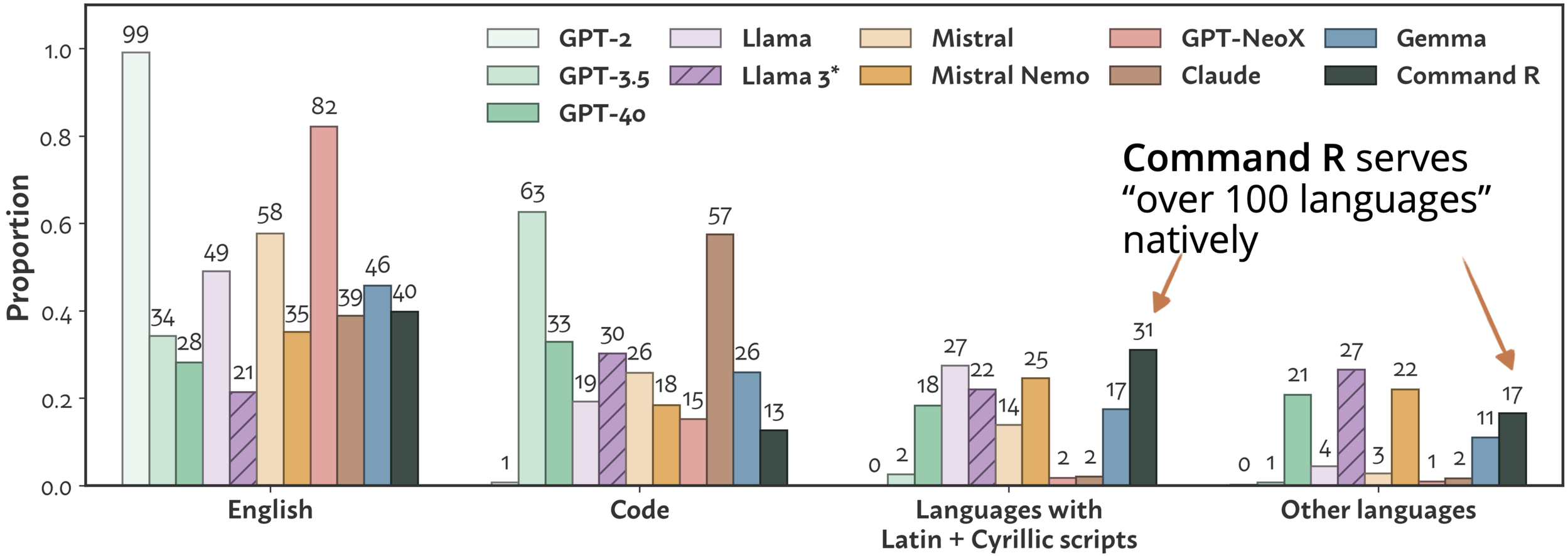
GPT-NeoX is “trained on the Pile” with “certain components… upsampled” — our findings are quite consistent but suggest books were upsampled

Our Inference for LLM Tokenizers



we don't know anything about **Claude**, but we find it's trained on more code than natural language!

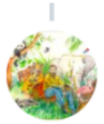
Our Inference for LLM Tokenizers



Logistics

- Project team finding doc is now available on the website
- Project instruction doc is up to date

Pitfalls of tokenization



Andrej Karpathy 
@karpathy

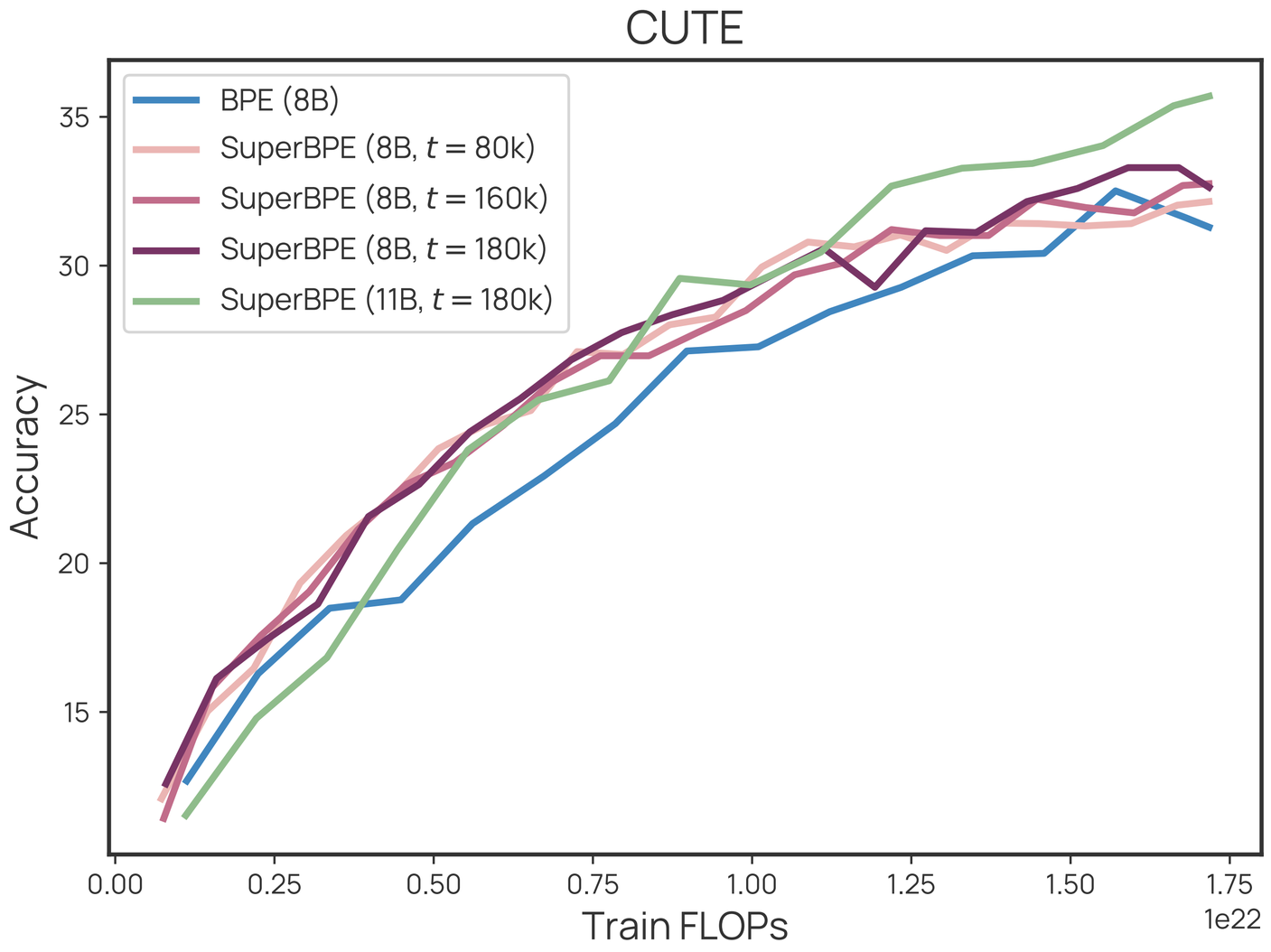
We will see that a lot of weird behaviors and problems of LLMs actually trace back to tokenization. We'll go through a number of these issues, discuss why tokenization is at fault, and why someone out there ideally finds a way to delete this stage entirely.

> Tokenization is at the heart of much weirdness of LLMs. Do not brush it off.
>
> - Why can't LLM spell words? **Tokenization**.
> - Why can't LLM do super simple string processing tasks like reversing a string? **Tokenization**.
> - Why is LLM worse at non-English languages (e.g. Japanese)? **Tokenization**.
> - Why is LLM bad at simple arithmetic? **Tokenization**.
> - Why did GPT-2 have more than necessary trouble coding in Python? **Tokenization**.
> - Why did my LLM abruptly halt when it sees the string "<|endoftext|>"? **Tokenization**.
> - What is this weird warning I get about a "trailing whitespace"? **Tokenization**.
> - Why the LLM break if I ask it about "SolidGoldMagikarp"? **Tokenization**.
> - Why should I prefer to use YAML over JSON with LLMs? **Tokenization**.
> - Why is LLM not actually end-to-end language modeling? **Tokenization**.
> - What is the real root of suffering? **Tokenization**.

9:40 AM · Feb 20, 2024 · **748.6K** Views

 58  400  2.7K  1K 

String manipulation tasks



Is there a " c " in " also "?

Delete every instance of " t " in " data ".

Swap " k " and " e " in " make ".

References

- “**SuperBPE: Space Travel for Language Models**”, Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, Yejin Choi, https://arxiv.org/pdf/2503.13423,

- **“Data Mixture Inference Attack: BPE Tokenizers Reveal Training Data Compositions”**, Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, Noah A. Smith, *NeurIPS 2024*

Sources

- Introduction to LLM tokenizers: BPE
  - https://medium.com/thedeephub/all-you-need-to-know-about-tokenization-in-llms-7a801302cf54
  - https://christophergs.com/blog/understanding-llm-tokenization
    - https://www.youtube.com/watch?v=zduSFxRajkE
    - https://hundredblocks.github.io/transcription_demo/
  - Fast implementation
    - https://github.com/openai/tiktoken
  - Failure modes of tokenizers
    - https://seantrott.substack.com/p/tokenization-in-large-language-models
  - Information theoretic approach to understand tokenization
    - https://arxiv.org/abs/2601.09039v1
  - Beyond tokenization
    - Byte latent transformers: https://arxiv.org/abs/2412.09871