Translation and Language Modeling with Pixels

Elizabeth Salesky

Sep 3, 2024

Johns Hopkins University

Center for Language and Speech Processing

Outline

Introduction and motivation	
Introducing visual representations of text	*Salesky et al. 2021*
Cross-lingual transfer with PIXEL	*Rust et al. 2023*
Multilingual modeling with visual representations	*Salesky et al. 2023*

What does it mean to be ***open-vocabulary?***

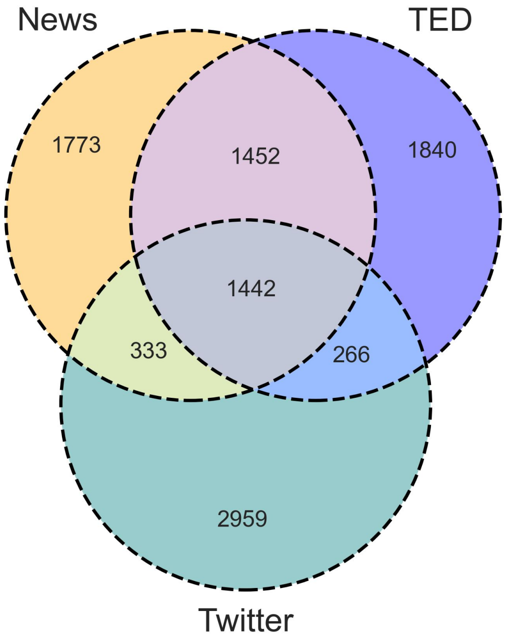
And why care about tokenization?

Open-vocabulary modeling

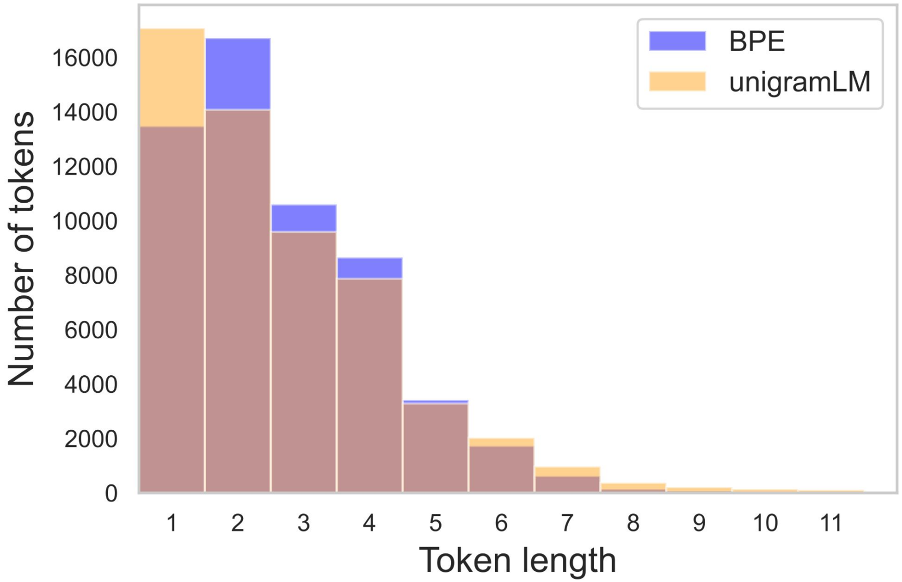
- Ideally our models should be able to represent ***all*** words in a given language
  - Not just avoid a placeholder token for unknown words, but appropriately model unseen input
  - Whether observed in training or not
- Typical techniques:
  - C h a r a c t e r s
  - Learned _sub words
  - Bytes

Unobserved components can (hopefully) be broken into observed components

Optimal vocabulary and size varies by task

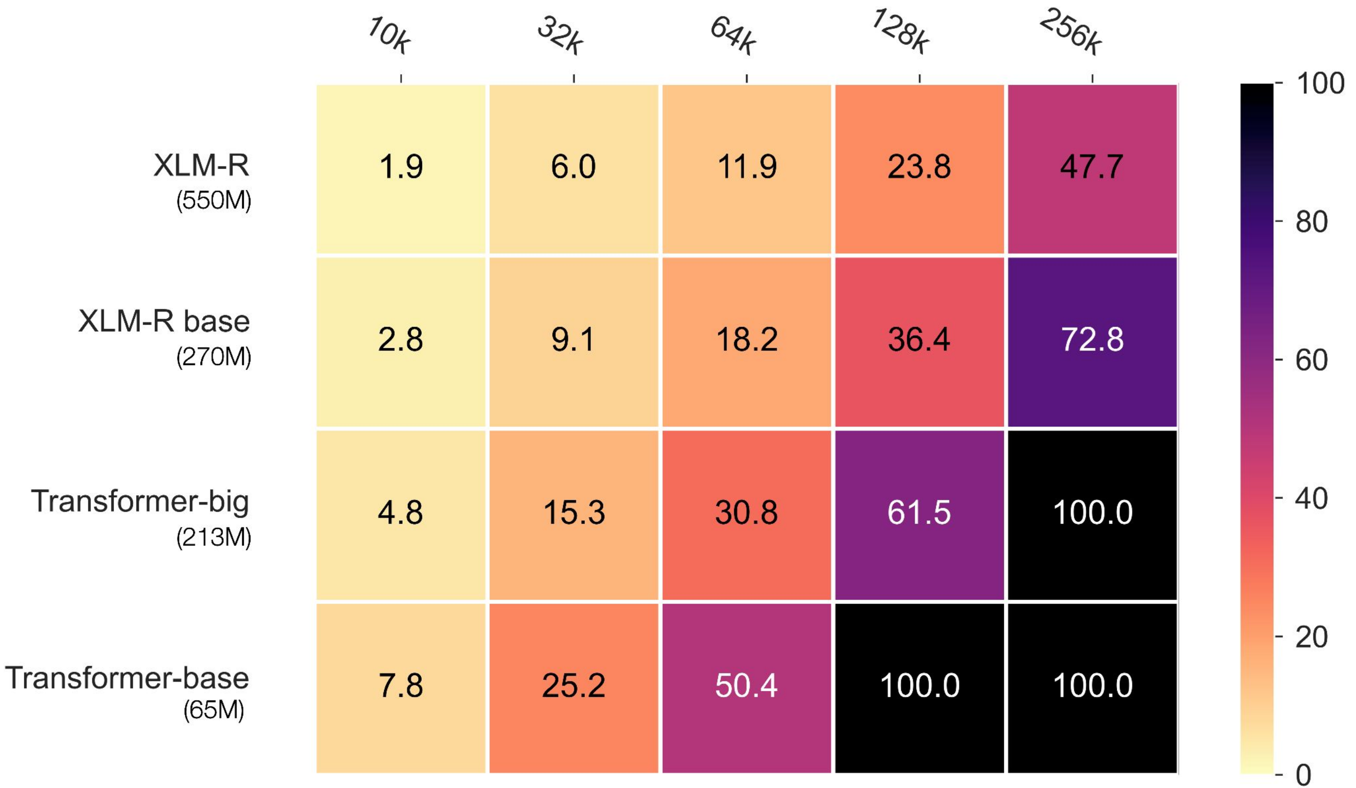


vocabulary overlap across domains



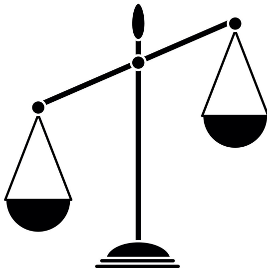
fertility across learned tokenizations

Compute bottleneck limits vocabulary sizes



Finite vocabularies limit language coverage

- Language support is limited by ***finite model vocabularies***
  - Larger vocab = Softmax and data sparsity problems
  - Smaller vocab = Sequence length or coverage problems
- We'll call this the ***vocabulary bottleneck***



Word-level vocabulary

Byte and character coverage

Character n-grams

BPE / WordPiece

UTF-8 bytes

UTF-32 codepoints

Model	Vocabulary size
FastText	250,000 (English)
XLM-R	250,000
BERT	30,000
ByT5	256
CANINE	144,697 characters

Softmax and sparsity challenges

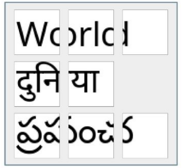
Increased sequence lengths

Finite vocabularies limit language coverage

World
English
Latin script

5 characters	W	o	r	l	d
5 codepoints	H	e	l	l	o
5 bytes	57	6f	72	6c	64
1 BPE token	10603				

3 visual tokens
2 visual tokens
3 visual tokens



OpenAI
GPT-3 tokenizer
Byte-level BPE

दुनिया
Hindi
Devanagari script

3 characters	दु						या						या					
6 codepoints	द			ु			न			ि			य			ा		
18 bytes	e0	a4	a6	e0	a5	81	e0	a4	a8	e0	a4	bf	e0	a4	bf	e0	a4	be
11 BPE tokens	11976		99	24231		223	11976		101	11976		123	11976		107	48077		

ప్రపంచం
Telugu
Telugu-Kannada script

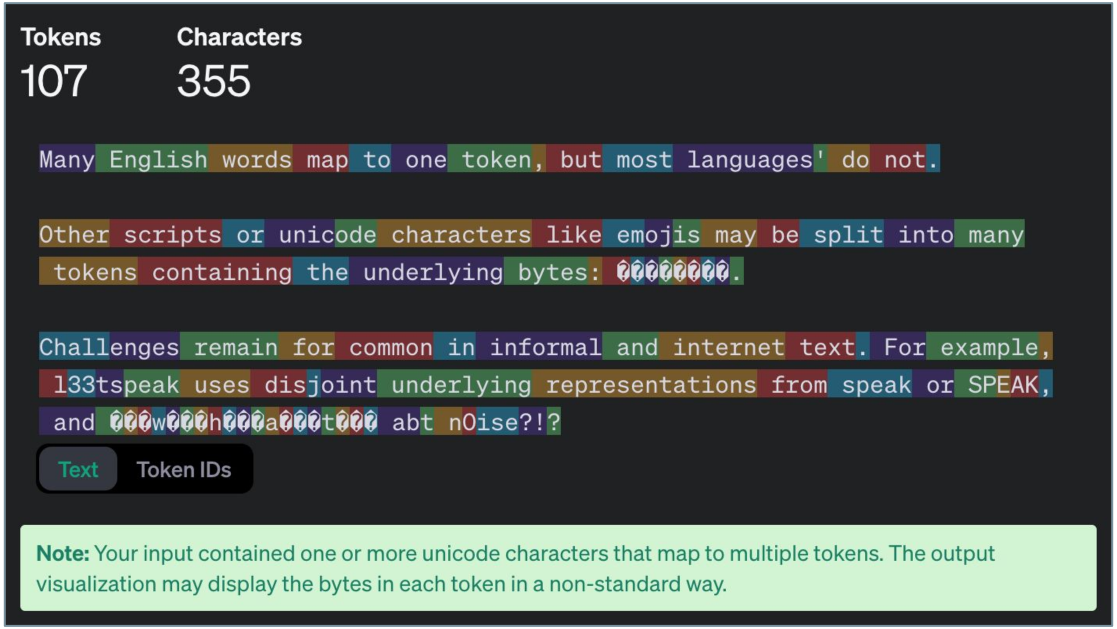
3 characters	ప్ర									పం						చం					
7 codepoints	ప			్			ర			ప			ం			చ			ం		
21 bytes	e0	b0	aa	e0	b1	8d	e0	b0	b0	e0	b0	aa	e0	b0	82	e0	b0	9a	e0	b0	82
21 BPE tokens	156	108	103	156	109	235	156	108	108	156	108	103	156	108	224	156	108	103	156	108	224

Are open vocabularies really open?

OpenAI

GPT-3 tokenizer

Byte-level BPE



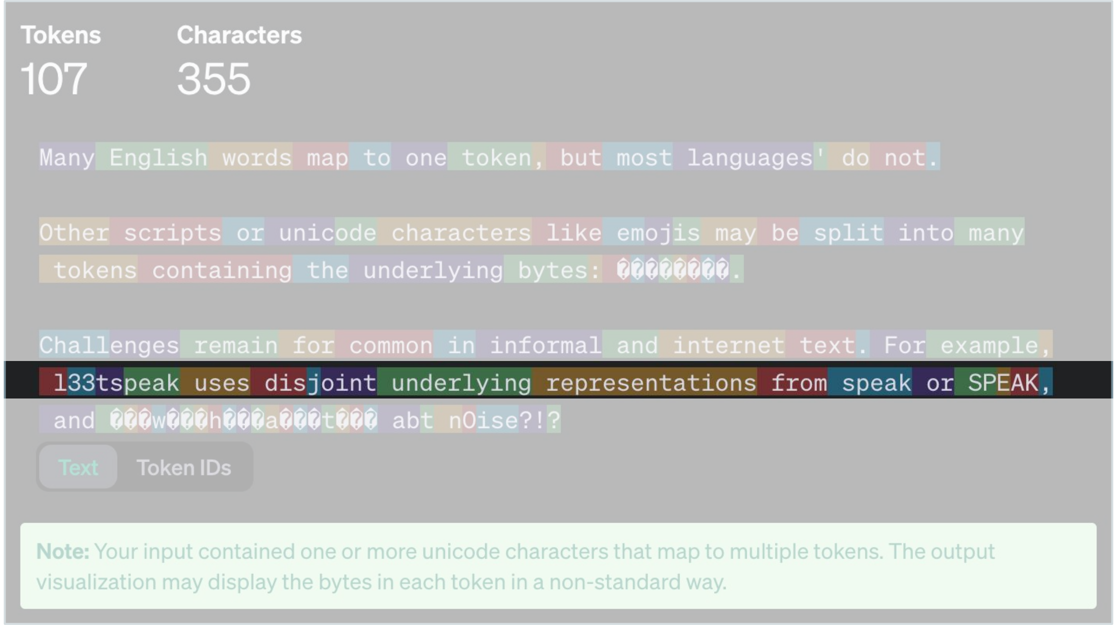
Are open vocabularies really open?

OpenAI

GPT-3 tokenizer

Byte-level BPE

s peak
speak
or
SPEAK?



Are open vocabularies really open?

Phenomena	Word	BPE	
Diacritization	كتاب	كتاب	(1)
	الكِتابُ	الك · ِ · ت · اب · ُ	(5)
Misspelling	language	language	(1)
	langauge	la · ng · au · ge	(4)
Visually Similar Characters	really	really	(1)
	rea11y	re · a · 1 · 1 · y	(5)
Shared Character Components	확인한다	확인 · 한 · 다	(3)
	확인했다	확인 · 했다	(2)

Examples of common behavior which cause divergent representations for subword models

Significant differences in sequences lengths and often disjoint embeddings

Are open vocabularies really open?

Phenomena	Word	BPE	
Diacritization	كتاب	كتاب	(1)
	الكِتابُ	الك · ِ · ت · اب · ُ	(5)
Misspelling	language	language	(1)
	langauge	la · ng · au · ge	(4)
Visually Similar Characters	really	really	(1)
	rea11y	re · a · 1 · 1 · y	(5)
Shared Character Components	확인한다	확인 · 한 · 다	(3)
	확인했다	확인 · 했다	(2)

Examples of common behavior which cause divergent representations for subword models

lhm

l h m

lh hm

laham

l a **h** a **m**

la ah ha am

lah ham

laha aham

Few possible subwords in common, 97% of pixels shared

Are open vocabularies really open?

Phenomena	Word	BPE	
Diacritization	كتاب	كتاب	(1)
	الكِتابُ	الك · ِ · ت · اب · ُ	(5)
Misspelling	language	language	(1)
	langauge	la · ng · au · ge	(4)
Visually Similar Characters	really	really	(1)
	rea11y	re · a · 1 · 1 · y	(5)
Shared Character Components	확인한다	확인 · 한 · 다	(3)
	확인했다	확인 · 했다	(2)

Examples of common behavior which cause divergent representations for subword models

Glyph	Codepoint
ە	U+06D5
ۀ	U+06C0
ه	U+0647
هٔ‌	U+0647, U+0654, U+200C
ۀ	U+06D5, U+0654
هٔ	U+0647, U+0654

Glyph	Codepoint
ي	U+064A
ی	U+06CC
ى	U+0649
دري	U+1583, U+1585, U+1744
دري	U+1583, U+1585, U+064A

Different underlying unicode codepoints render visually similarly

Open-vocabulary modeling

- Ideally our models should be able to represent ***all*** words in a given language
  - Not just avoid a placeholder token for unknown words, but appropriately model unseen input
  - Whether observed in training or not
- Typical techniques:
  - C h a r a c t e r s
  - Learned _sub words
  - Bytes

Unobserved components can (hopefully) be broken into observed components

Potential issues

How to construct an optimal finite vocabulary?
26 Latin characters, >10k Chinese characters

In-vocabulary does not guarantee good results
Long-tail vocabulary, unicode noise, emojis, ...

Are open vocabularies really open?

No, not really!

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**.

Introducing visual representations of text

Underlying units

Café :

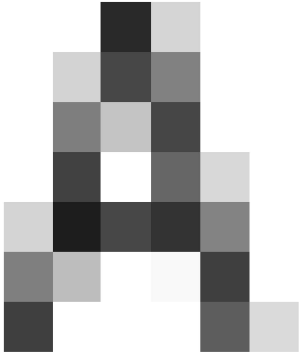
UTF-8 Bytes	43	61	66	65	CC 81
Unicode codepoints	U+0043	U+0061	U+0066	U+0065	U+0301
Characters	C	a	f	e	◌́
Graphemes	C	a	f	é	

Robustness

a 0061 LATIN SMALL LETTER A	ɑ 0251 LATIN SMALL LETTER ALPHA	α 03B1 GREEK SMALL LETTER ALPHA	а 0430 CYRILLIC SMALL LETTER A	⍺ 237A APL FUNCTIONAL SYMBOL ALPHA	𝐚 1D41A MATHEMATICAL BOLD SMALL A
𝑎 1D44E MATHEMATICAL ITALIC SMALL A	𝒂 1D482 MATHEMATICAL BOLD ITALIC SMALL A	𝒶 1D4B6 MATHEMATICAL SCRIPT SMALL A	𝓪 1D4EA MATHEMATICAL BOLD SCRIPT SMALL A	𝔞 1D51E MATHEMATICAL FRAKTUR SMALL A	𝕒 1D552 MATHEMATICAL DOUBLE-STRUCK SMALL A
𝖆 1D586 MATHEMATICAL BOLD FRAKTUR SMALL A	𝖺 1D5BA MATHEMATICAL SANS-SERIF SMALL A	𝗮 1D5EE MATHEMATICAL SANS-SERIF BOLD SMALL A	𝘢 1D622 MATHEMATICAL SANS-SERIF ITALIC SMALL A	𝙖 1D656 MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL A	𝚊 1D68A MATHEMATICAL MONOSPACE SMALL A
𝛂 1D6C2 MATHEMATICAL BOLD SMALL ALPHA	𝛼 1D6FC MATHEMATICAL ITALIC SMALL ALPHA	𝜶 1D736 MATHEMATICAL BOLD ITALIC SMALL ALPHA	𝝰 1D770 MATHEMATICAL SANS-SERIF BOLD SMALL ALPHA	𝞪 1D7AA MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL ALPHA	ａ FF41 FULLWIDTH LATIN SMALL LETTER A

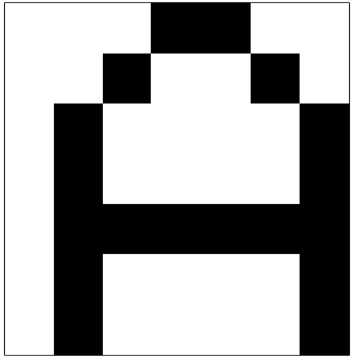
Rendering text

Letter A, font size 5



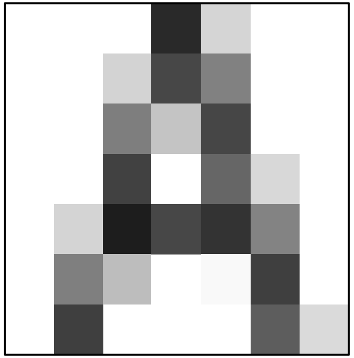
Rendering text

Bitmap font



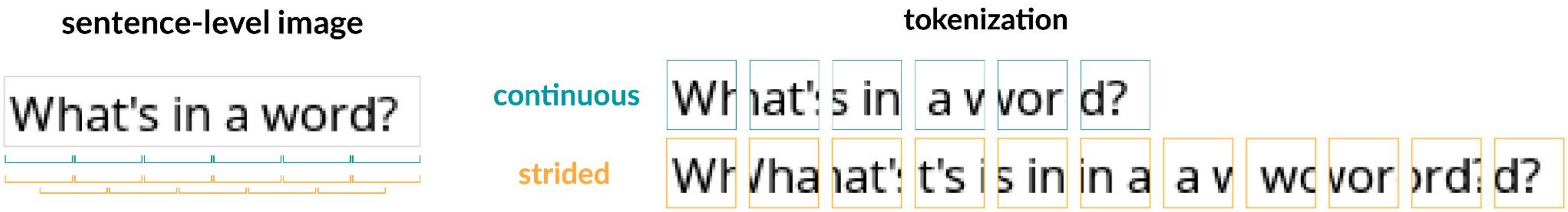
0	0	0	1	1	0	0
0	0	1	0	0	1	0
0	1	0	0	0	0	1
0	1	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	0	0	0	1
0	1	0	0	0	0	1

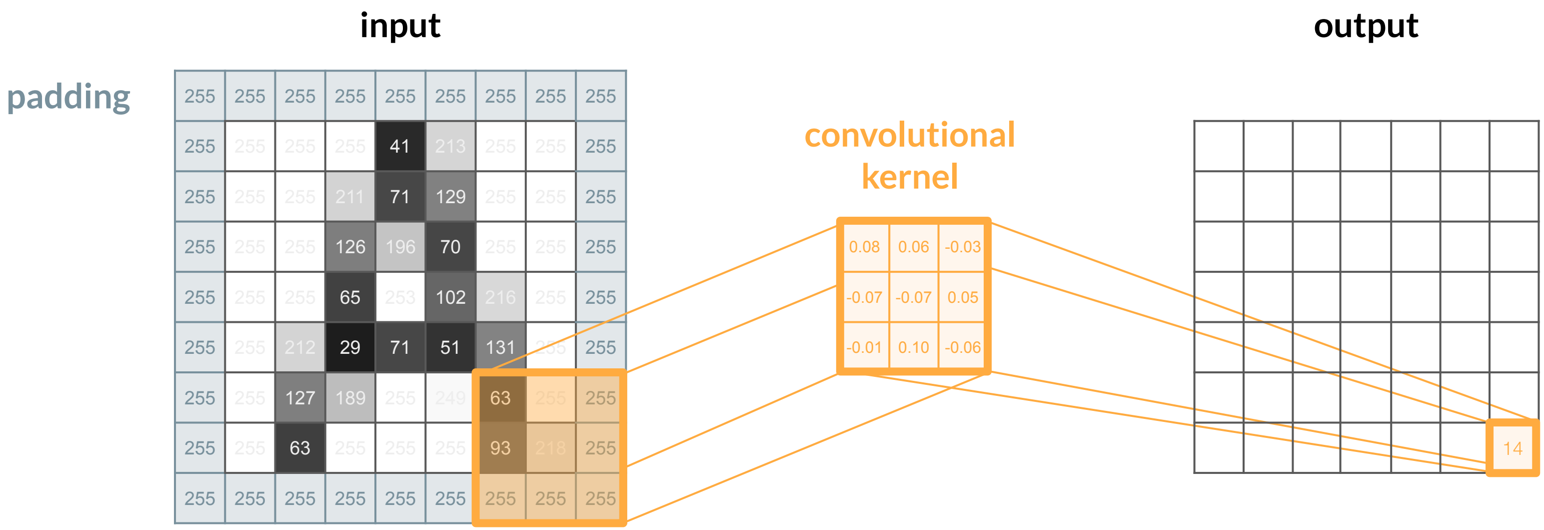
Vector font



255	255	255	41	213	255	255
255	255	211	71	129	255	255
255	255	126	196	70	255	255
255	255	65	253	102	216	255
255	212	29	71	51	131	255
255	127	189	255	249	63	255
255	63	255	255	255	93	218

Tokenizing rendered text

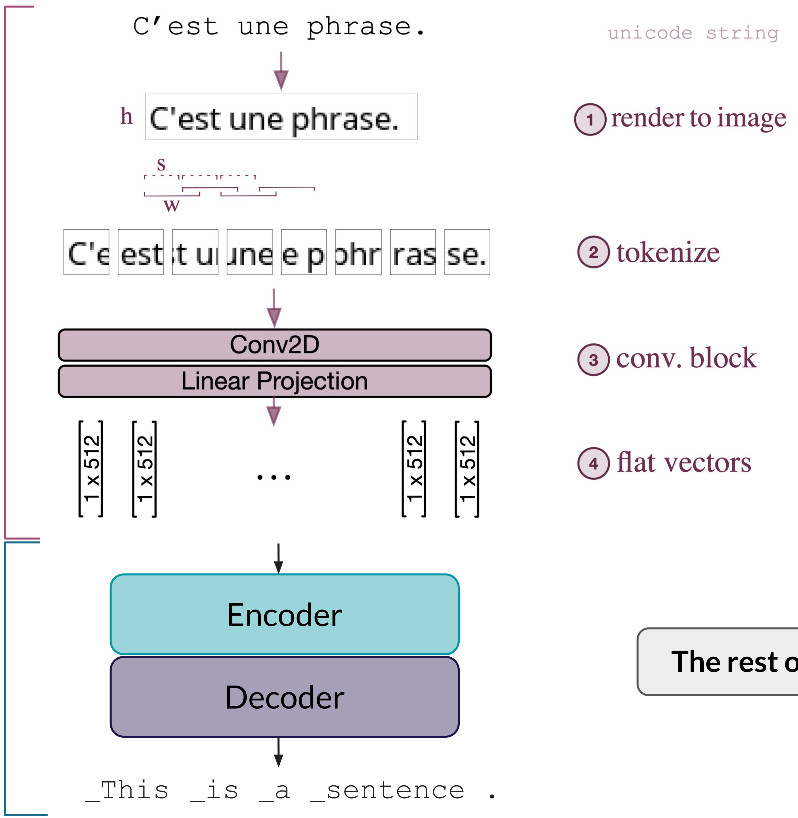




Machine translation with visual representations

Visual embedder

Standard Transformer



Parameters:
~200k,
constant wrt V

<1% of total model parameters

The rest of the model proceeds as usual

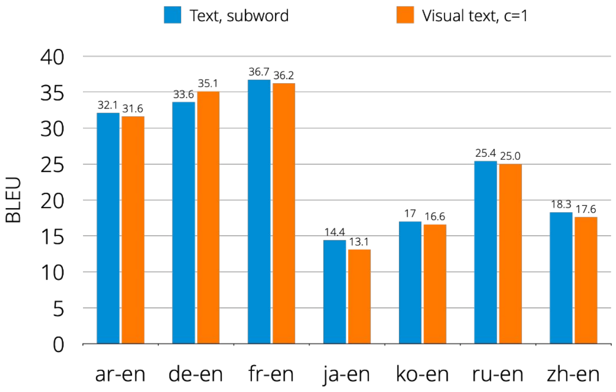
Initial experiments: machine translation

- **Language pairs (7)**
  - Source, multiple scripts: Arabic عربي, Chinese 中文, French Français, German Deutsch, Japanese 日本語, Korean 한국어, Russian русский
  - Target language: English
- **Datasets (2)**
  - "Small" — MTTT (TED) ar zh fr de ja ko ru
  - "Larger" — WMT (filtered) zh de
- **Visual architecture**
  - Significant hyperparameters unknown at the offset — new approach!
  - Convolutional blocks {0,1,7} *0≅Vision Transformer; 7≅OCR*

Clean translation results

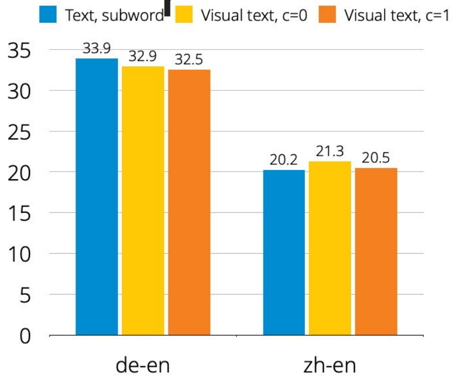
- Results are on par with heavily tuned subword models at “smaller” and “larger” data scales for multiple language pairs with different scripts

smaller



Standardized MTTT test set
c = num. convolution blocks

large
r



WMT’20 newstest sets
c = num. convolution blocks

Robustness

a 0061 LATIN SMALL LETTER A	ɑ 0251 LATIN SMALL LETTER ALPHA	α 03B1 GREEK SMALL LETTER ALPHA	а 0430 CYRILLIC SMALL LETTER A	⍺ 237A APL FUNCTIONAL SYMBOL ALPHA	𝐚 1D41A MATHEMATICAL BOLD SMALL A
𝑎 1D44E MATHEMATICAL ITALIC SMALL A	𝒂 1D482 MATHEMATICAL BOLD ITALIC SMALL A	𝒶 1D4B6 MATHEMATICAL SCRIPT SMALL A	𝓪 1D4EA MATHEMATICAL BOLD SCRIPT SMALL A	𝔞 1D51E MATHEMATICAL FRAKTUR SMALL A	𝕒 1D552 MATHEMATICAL DOUBLE-STRUCK SMALL A
𝖆 1D586 MATHEMATICAL BOLD FRAKTUR SMALL A	𝖺 1D5BA MATHEMATICAL SANS-SERIF SMALL A	𝗮 1D5EE MATHEMATICAL SANS-SERIF BOLD SMALL A	𝘢 1D622 MATHEMATICAL SANS-SERIF ITALIC SMALL A	𝙖 1D656 MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL A	𝚊 1D68A MATHEMATICAL MONOSPACE SMALL A
𝛂 1D6C2 MATHEMATICAL BOLD SMALL ALPHA	𝛼 1D6FC MATHEMATICAL ITALIC SMALL ALPHA	𝜶 1D736 MATHEMATICAL BOLD ITALIC SMALL ALPHA	𝝰 1D770 MATHEMATICAL SANS-SERIF BOLD SMALL ALPHA	𝞪 1D7AA MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL ALPHA	ａ FF41 FULLWIDTH LATIN SMALL LETTER A

zero-shot

diacritics & unicode

Robustness

ar-en

src أنا كندية، وأنا أصغر أخواني السبعة

noised أنا كَندِيَّة ، وَأَنا أَصْغَرِ إِخْوانِي السَبْعَة

ref I'm Canadian, and I'm the youngest of seven kids.

visrep

I'm a Canadian, and I'm the youngest of my seven sisters.

BPE _ ِأ َان _ك َن َد ِي ّ َة ٍ_, _و َ_أ َان _أ َص ْغ َر ِ_إ ِخ ْاو ن ِي _سلا َب ْع َ_ة .

We grew up as a teacher, and we gave me a hug.

zero-shot

diacritics & unicode

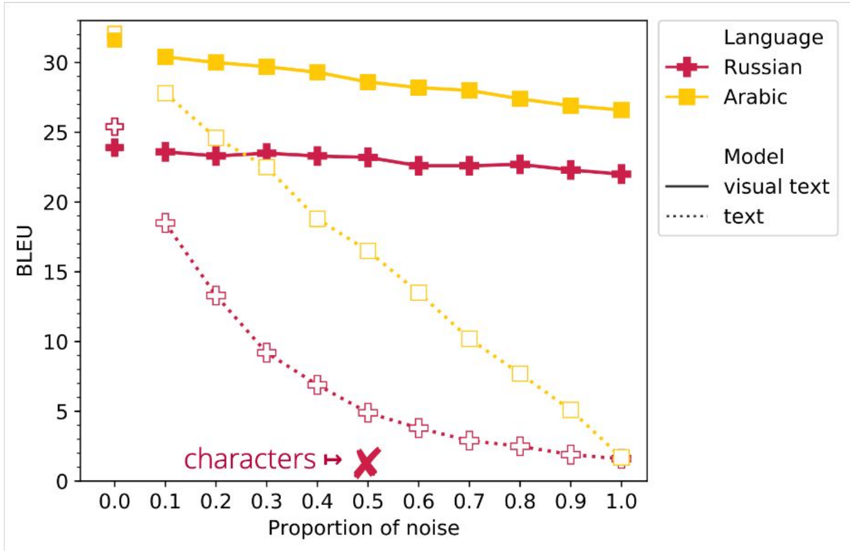
Robustness

- Large changes to unicode sequences; visually, changes to only 0-5% pixels
  - Unsurprising our method does so well!

WIPO

The invention belongs to the field of biotechnology, pharmaceutics and medicine, it could be applied for the production of drugs and for the realization of medicinal technologies, particularly for the immunotherapy of oncological diseases.

Cyrillic Latin

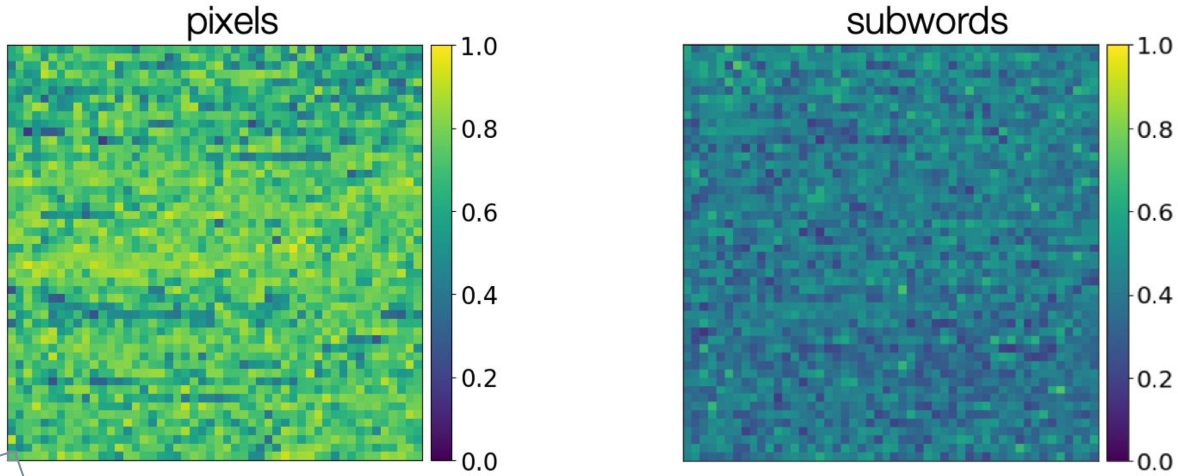


zero-shot

diacritics

Robustness

Avg: 0.73



Avg: 0.41

Every square is a unique word in the dev set

Cosine similarity between representations with and without diacritics, by model

zero-shot

character permutations

Robustness

de-en

src Aber Sie müssen zuerst zwei Dinge über mich wissen.

noised Abre Sie müssen zuerts wzei Dnige über mcih wisse.n

ref But first you need to know two things about me.

visrep  . . .

But you have to know two things about me first.

BPE _Ab re _Sie _müssen _zu ert s _w z ei _D n ige _über _m ci h _ wiss e . n

But you've got to get into a little about you.

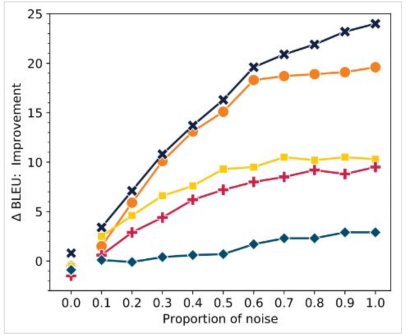
zero-shot

character permutations

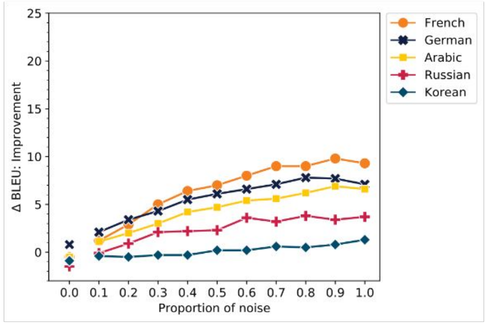
Robustness

- Significant improvements for all pairs, even if slight performance gap on clean text
  - Highlighting German–English:
    - At `swap p=1.0`, the visrep model is usable (25.9 BLEU) while the text model is not (1.9 BLEU)

swap



cmabrigde



zero-shot

character permutations

Robustness

Why does this work?

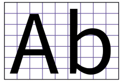
german (de)

subwords `_Ab` `re`

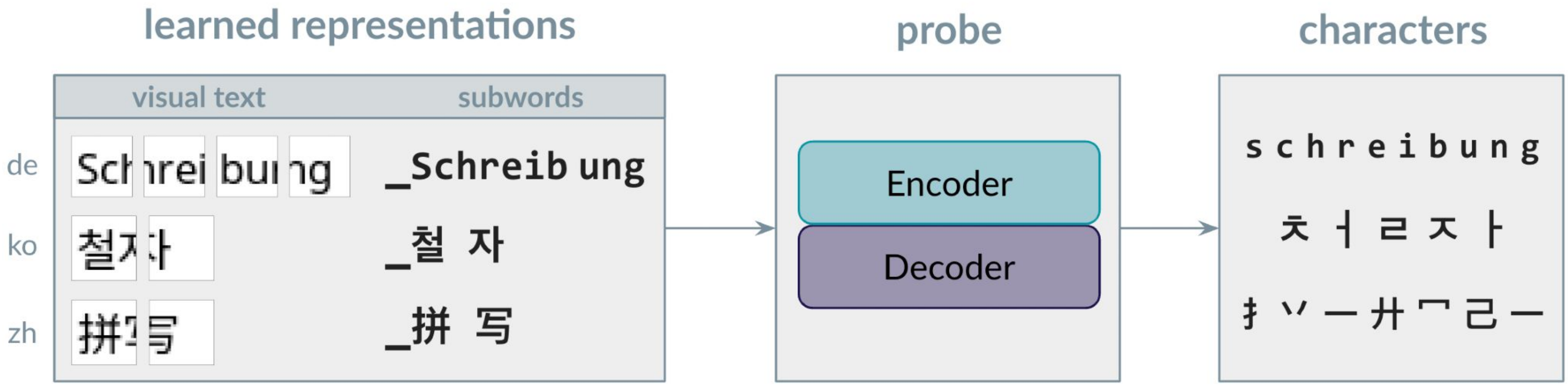
rendered text Ab re

embeddings v_4 v_{257}

pixels



Probing representations for compositionality



Probing representations for compositionality

chrF on character compositionality probing tasks for German, Korean, and Chinese

		Visual Text			Subwords		
	Probing task	**German**	**Korean**	**Chinese**	**German**	**Korean**	**Chinese**
CONTROL	Embeddings (updatable)	68.7	67.7	42.3	38.3	59.2	36.6
PROBE	Embeddings (frozen)	62.2	61.6	30.8	36.7	48.0	27.2
CONTROL	Random (updatable)	60.6	54.4	39.1	32.4	44.0	33.3
CONTROL	Random (frozen)	26.4	39.1	29.8	21.9	29.1	25.0

Recover composition
12-43% better using visual text representations

Probing representations for compositionality

chrF on character compositionality probing tasks for German, Korean, and Chinese

	Probing task	Visual Text German	Visual Text Korean	Visual Text Chinese	Subwords German	Subwords Korean	Subwords Chinese
CONTROL	Embeddings (updatable)	68.7	67.7	42.3	38.3	59.2	36.6
PROBE	Embeddings (frozen)	62.2	61.6	30.8	36.7	48.0	27.2
CONTROL	Random (updatable)	60.6	54.4	39.1	32.4	44.0	33.3
CONTROL	Random (frozen)	26.4	39.1	29.8	21.9	29.1	25.0

Possible to learn these tasks with these representations, but not a necessary part of the translation task

Leading questions

All in extra slides, if someone is curious at the end!

- What about computational efficiency?
- Can't I just run text normalization and proceed as normal?
- Would something like subword regularization or BPE-dropout do better?
- If these representations are compositional, what about morphology?
- Ablations:
  - Sliding window segmentation without visual text representations?
  - Visual text representations without sliding window segmentation (aligned to subwords)?
  - Can you combine BPE embeddings and visual text representations?

Cross-lingual transfer with PIXEL

Cross-lingual transfer

- Model vocabularies are predetermined and do not include all scripts
  - Can require adapters or vocabulary expansion for cross-lingual transfer
    - ➢ Visual representations may transfer across scripts, as-is!

- Pixel representations remove the source embedding matrix
  - Can we design a fully vocabulary-free model?

Cross-lingual transfer

- Model vocabularies are predetermined and do not include all scripts
  - Can require adapters or vocabulary expansion for cross-lingual transfer
    - ➢ Visual representations may transfer across scripts, as-is!

- Pixel representations remove the source embedding matrix
  - Can we design a fully vocabulary-free model?

- Enter PIXEL: a ***vocabulary-free*** encoder (PIXEL: a pixel-based encoder of language)
  - Potentially supports all written languages
  - PIXEL has no embedding matrix, no finite or predetermined vocabulary
  - Easily extensible to unseen text (and scripts), as we'll see shortly

PIXEL building blocks

- Robust Open Vocabulary Translation from Visual Text Representations (Salesky et al. EMNLP 2021)

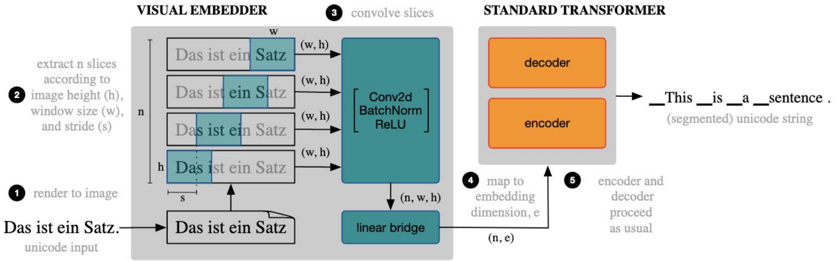
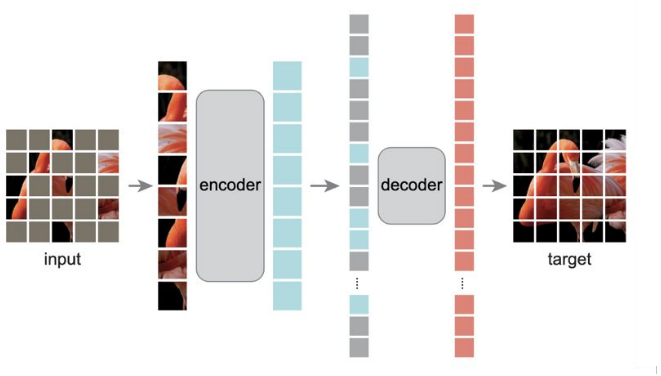
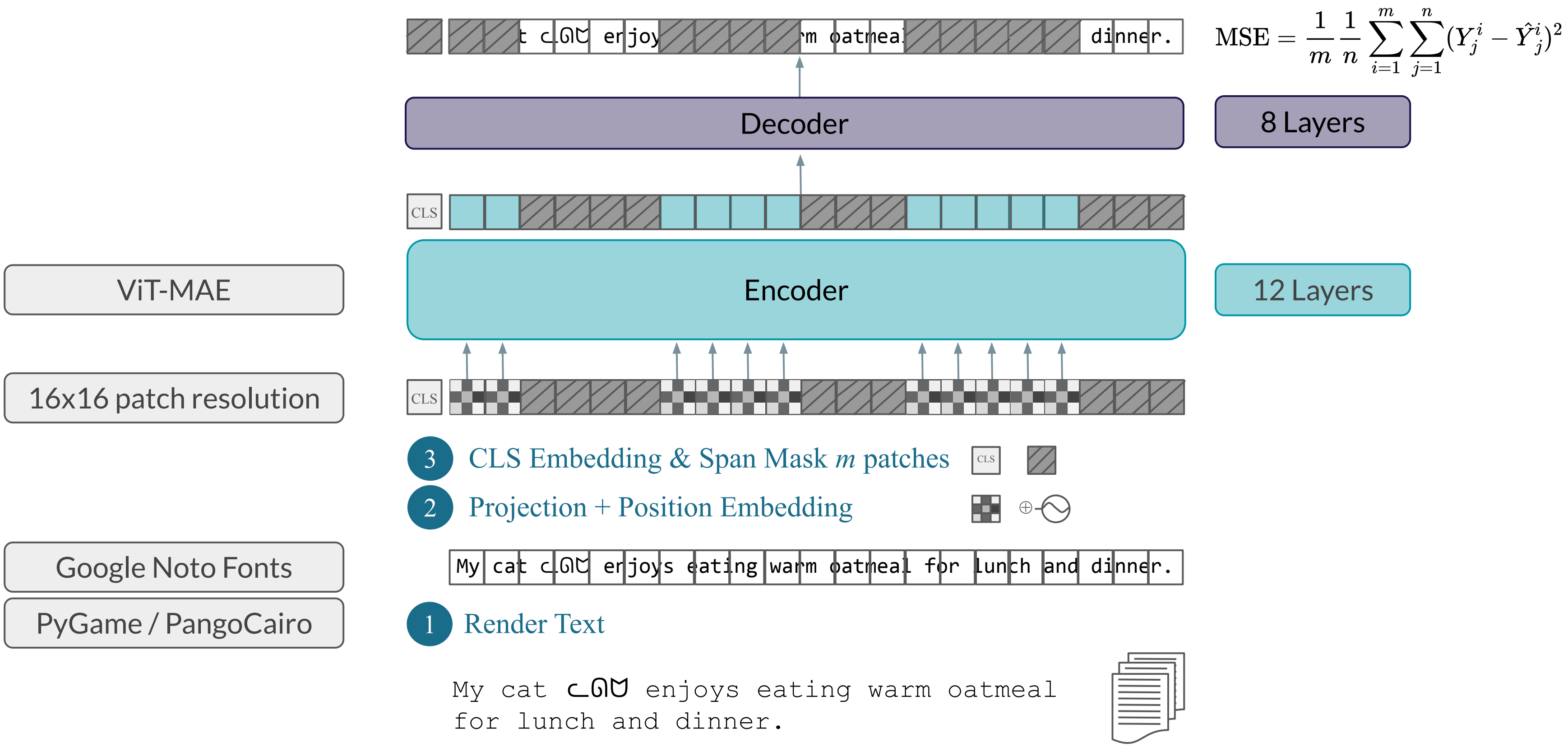


Figure 1: Visual text architecture combines network components from OCR and NMT, trained end-to-end.

- Masked Autoencoders are Scalable Visual Learners (He et al. 2021)

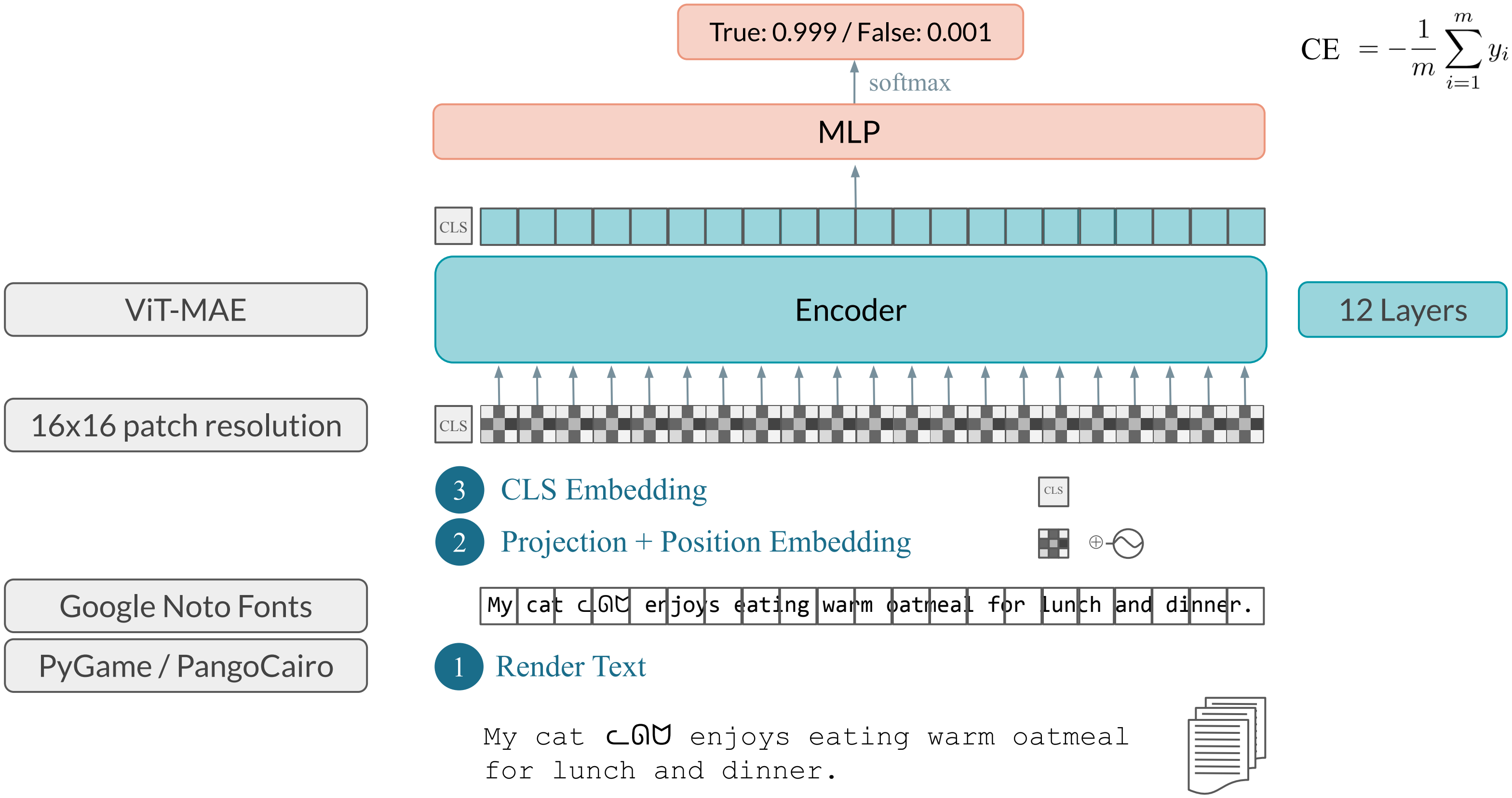


PIXEL architecture



41

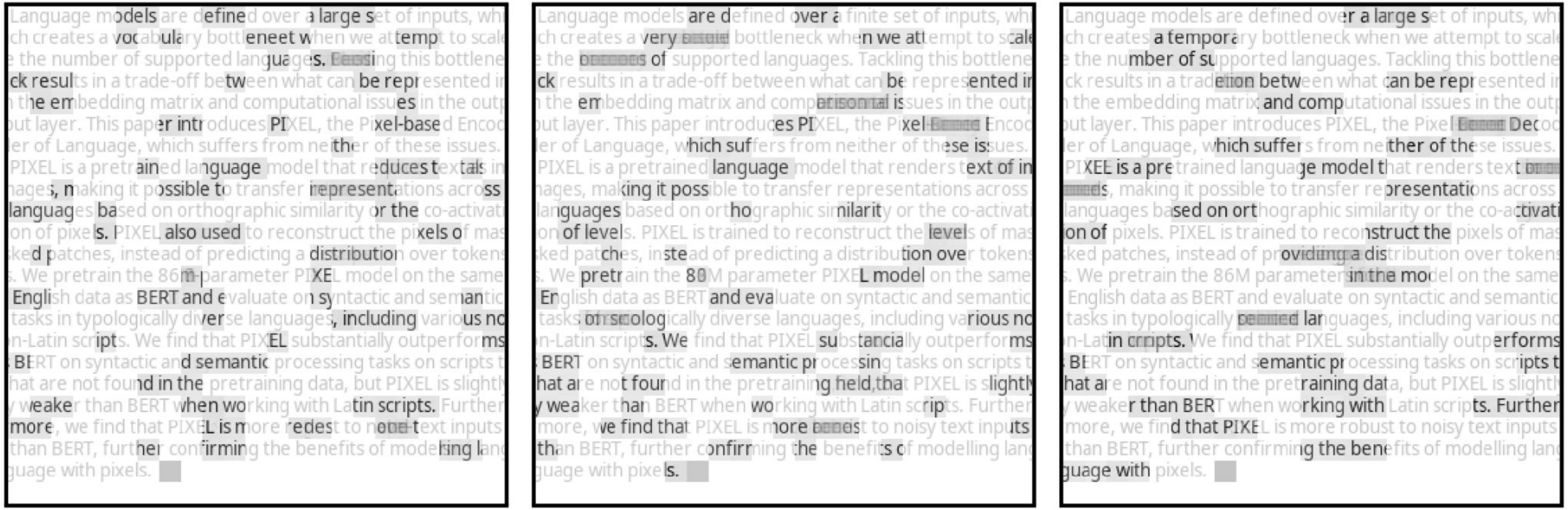
Finetuning PIXEL



$$\mathrm{CE} = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

42

PIXEL abstract reconstructions



Gradio demo: https://huggingface.co/spaces/Team-PIXEL/PIXEL monolingual

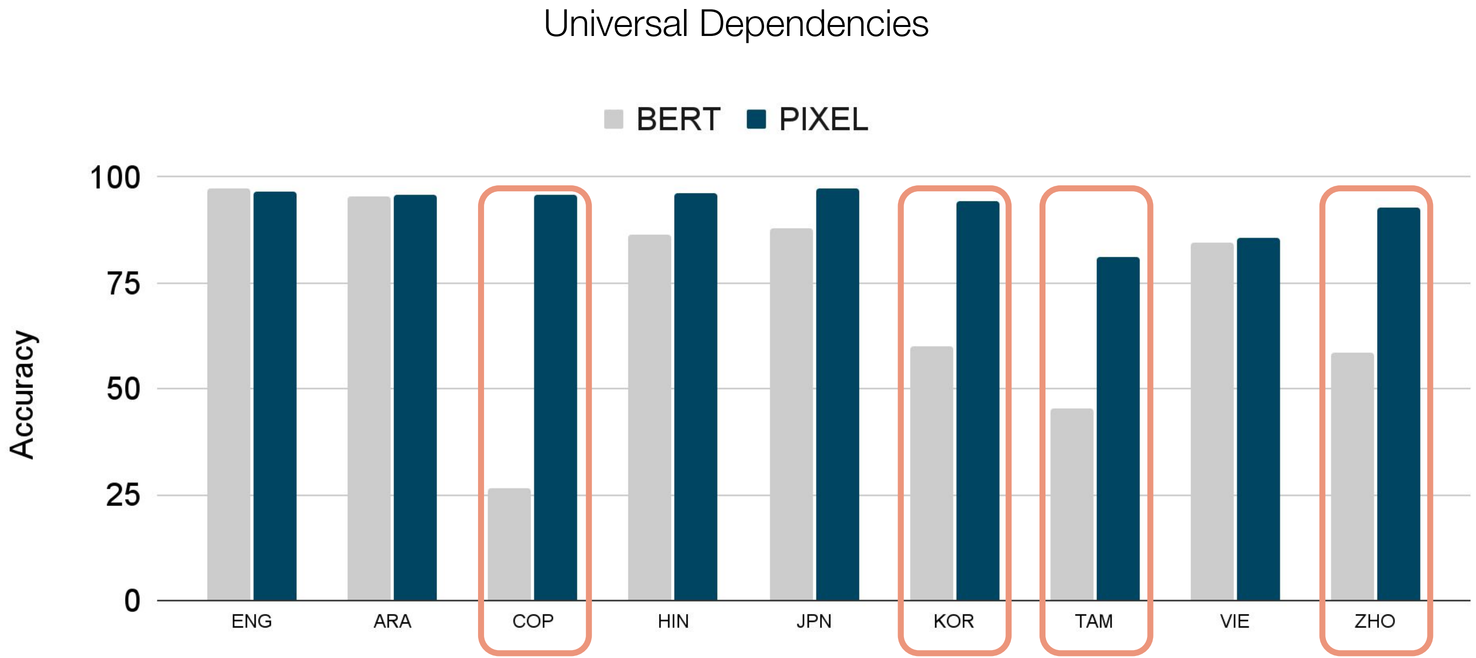
Pretraining PIXEL

- **Dataset**: English Wikipedia and Books Corpus

  approx. BERT training corpus
- **Masking:** 25% Span Masking
- **Patch size**: `16x16` pixels
- **Maximum sequence length**: 529 patches (`368x368` pixels)
- **Compute**: 8 x A100 GPUs for ~8 days
- **Parameters**: 86M encoder + 26M decoder

There is ~0.05% non-English text in the pretraining data
(estimated by Blevins and Zettlemoyer 2022)

Syntax: Part-of-Speech Tagging Results

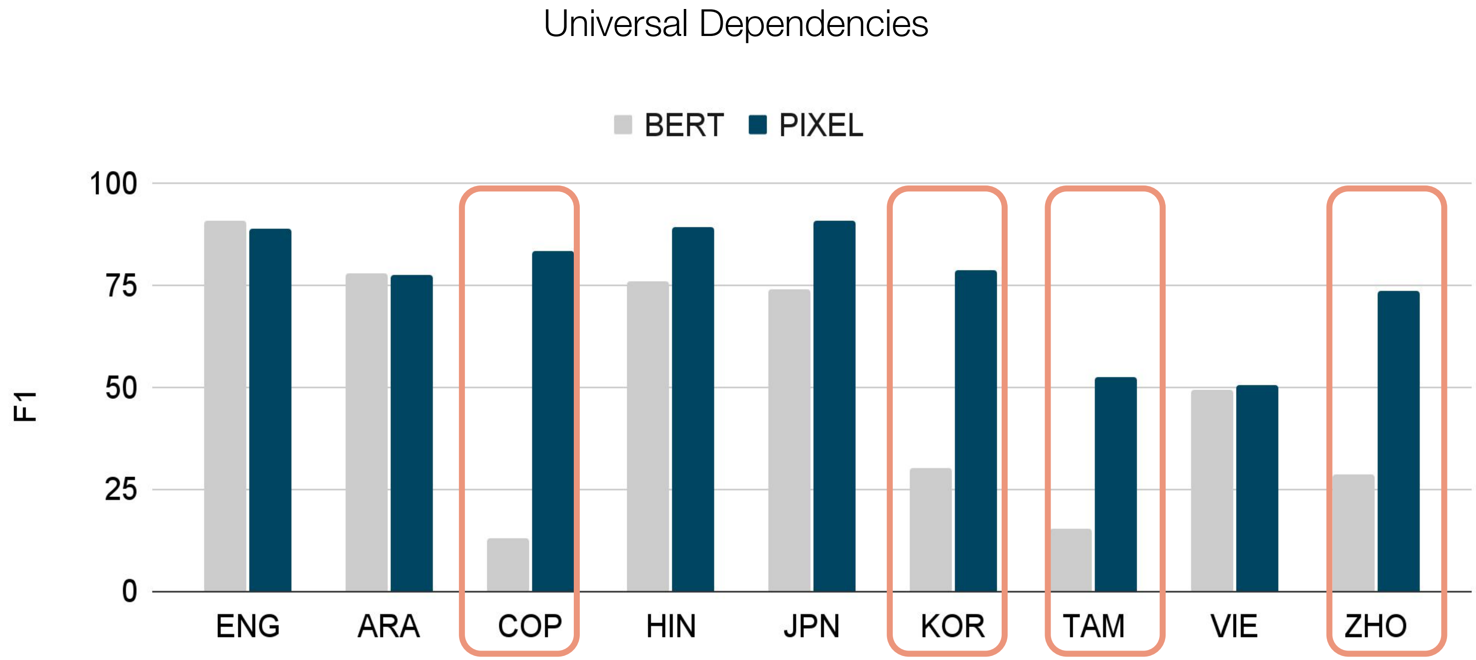


PIXEL outperforms BERT by a large margin on unseen scripts

PIXEL finetuned per language and task

POS: more local dependencies

Syntax: Dependency Parsing Results

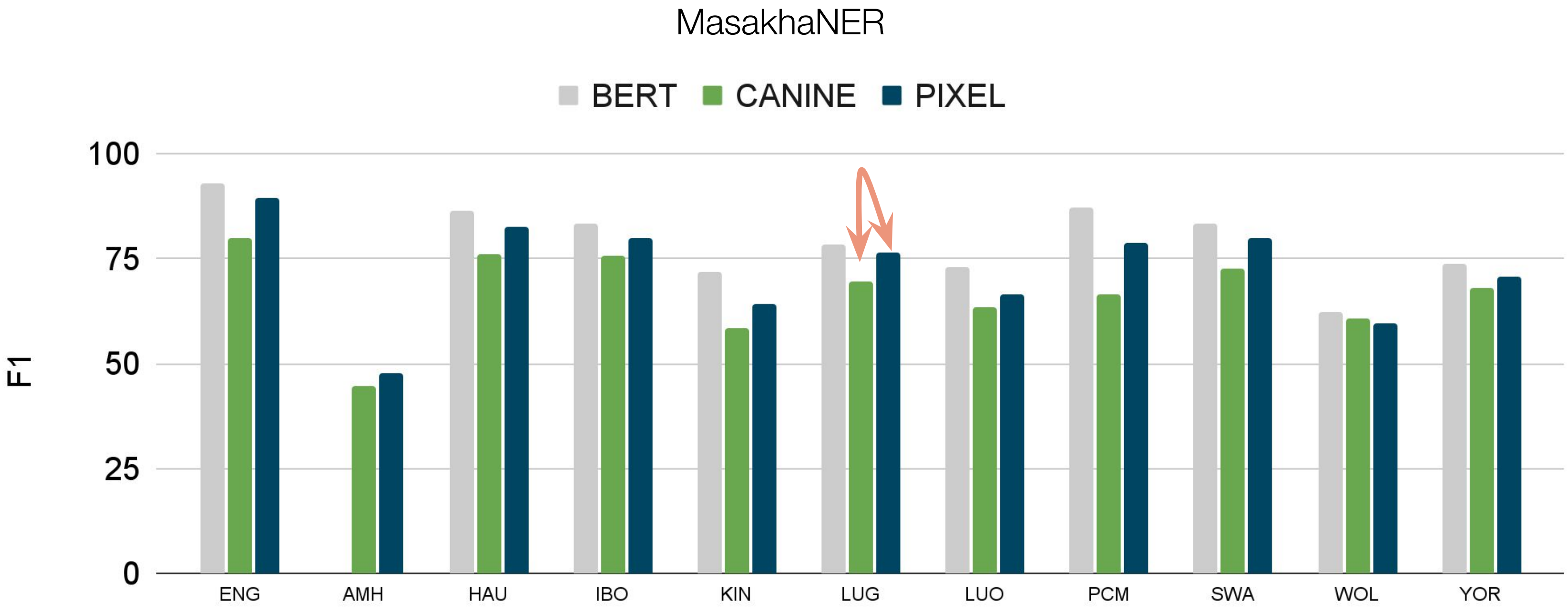


PIXEL outperforms BERT by a large margin on unseen scripts

PIXEL finetuned per language and task

DP: more global dependencies

Named Entity Recognition in African Languages



BERT outperforms PIXEL
on Latin scripts

PIXEL nearly always
outperforms CANINE-C

PIXEL finetuned per language and task

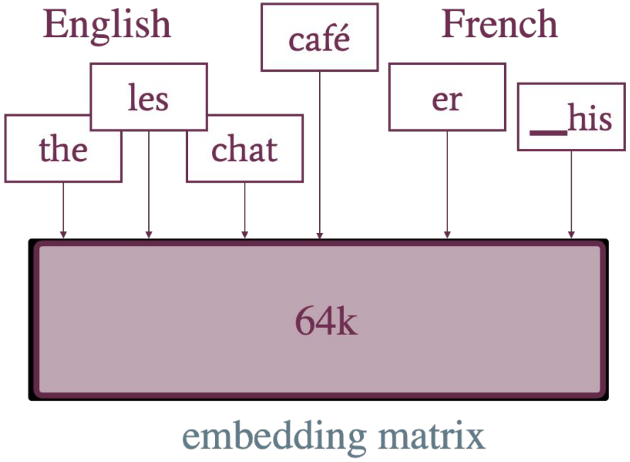
Multilingual modeling with visual representations

Vocabulary bottleneck in multilingual models

- Original vocabulary covers English and a fraction of languages with Latin, Cyrillic scripts
- Significant increase in parameters in order to increase language coverage (+30%)
  - Larger vocabulary increases MLM training time
  - Minimal parameter sharing across scripts

Vocabulary bottleneck in multilingual models

As we add languages to our models…   vocabulary capacity per language is reduced



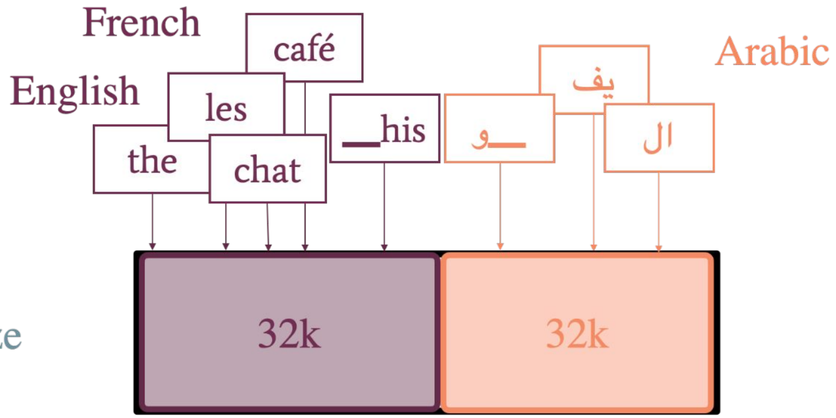
matrix of predetermined, fixed size

embeddings are disjoint by script

Vocabulary bottleneck in multilingual models

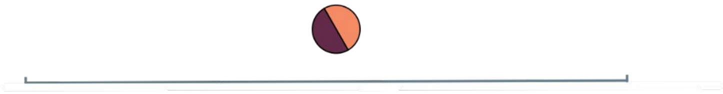
As we add languages to our models… vocabulary capacity per language is reduced

matrix of predetermined, fixed size



embedding matrix

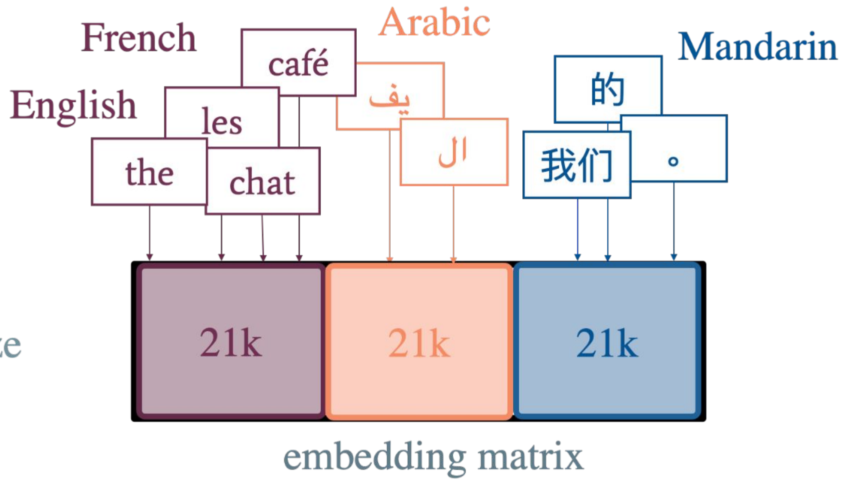
embeddings are disjoint by script



embedding matrix allocation

Vocabulary bottleneck in multilingual models

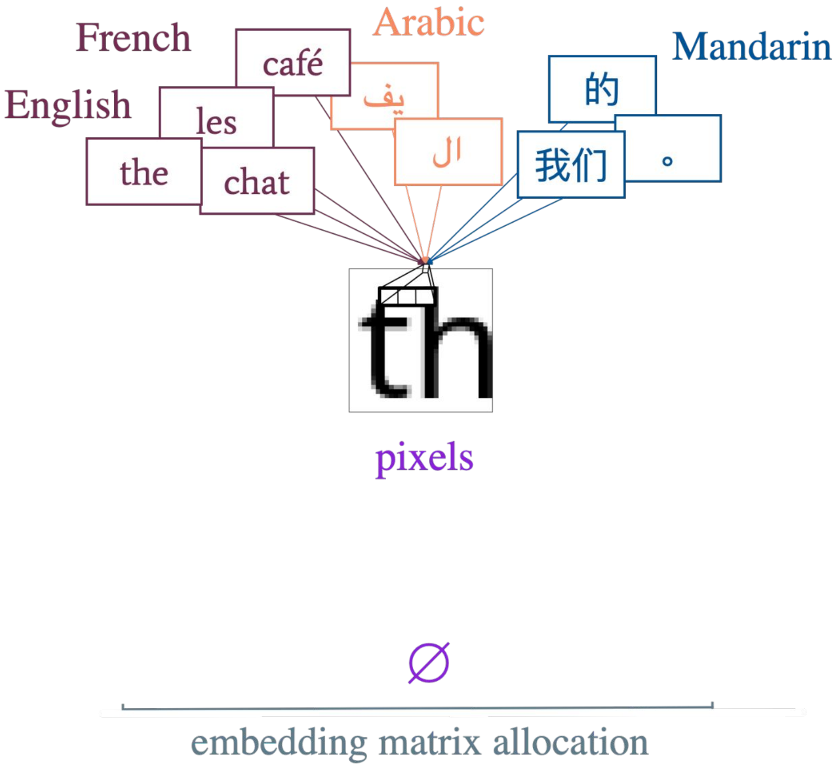
As we add languages to our models… vocabulary capacity per language is reduced



matrix of predetermined, fixed size

embeddings are disjoint by script

Pixels are shared between scripts in rendered text



Pixels are shared between scripts

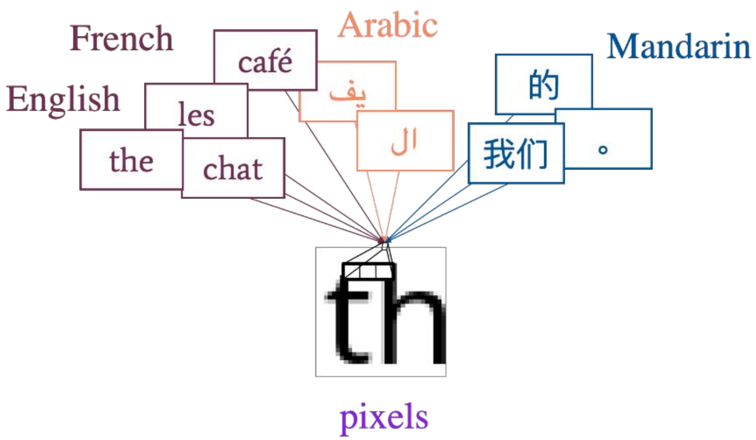
Models have dynamic, adaptable vocabularies

Our research questions

What is the impact of multilingual training with pixels?

Are architectural changes needed given the larger input space in a multilingual context?

A closer look at cross-lingual transfer:
are pixels more data-efficient than subwords?



Experimental setup

TED-7

- 7 source languages, 6 scripts (data from previous section)
- 1.2M training examples
- Balanced data across langs

TED-59

- 58 source languages, 17 scripts
- 5.1M training examples
- Imbalanced data across langs

Indic

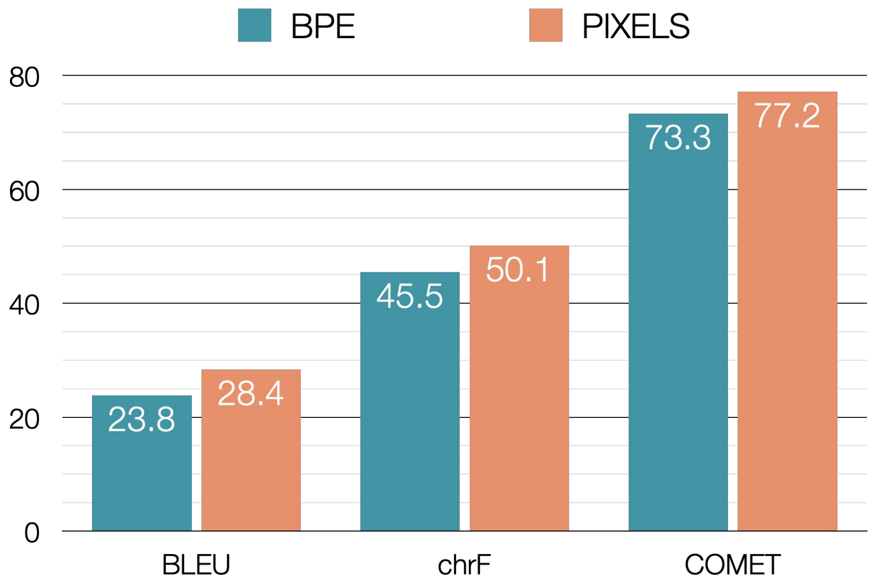
- 11 source languages, 9 scripts
- 50M training examples
- Imbalanced data across langs

All many-to-one machine translation (into English)
All models vary only by source representations

overall

Multilingual translation performance

Average improvement of +12% across 3 evaluation metrics



TED-59

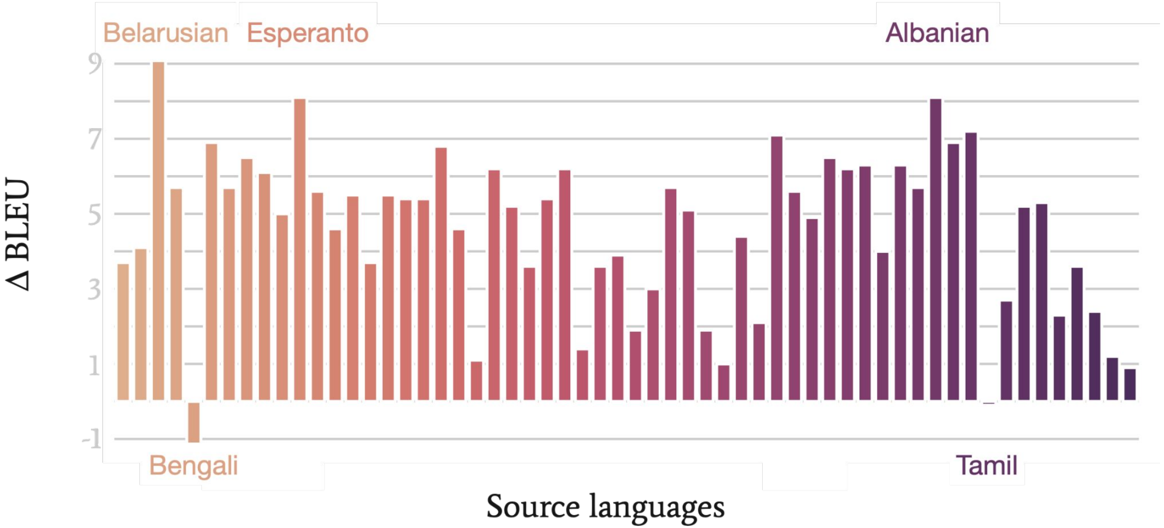
Many-to-one multilingual models

per language

Multilingual translation performance

Consistent improvements across all but 2 language pairs

➢ At least 5k examples required to visually learn a new script



TED-59

What is behind the improvements with pixels?

- Greater positive transfer across shared scripts
    - Complete parameter sharing gives stronger co-training benefits even without shared scripts

Amount of data per...

Largest improvements for languages with shared scripts

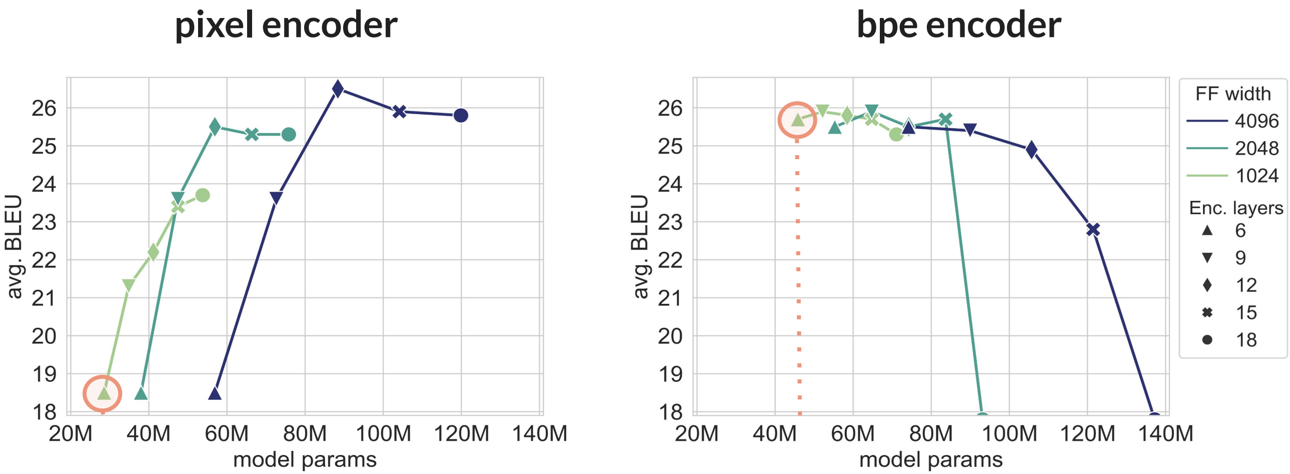
Low-resource languages with high-resource scripts benefit most from pixels

		Scores in BLEU				
		PIXEL	*BPE*	*Δ*	*Sents per Language*	*Sents per Script*
Best	*Belarusian (be)*	28.5	19.2	+9.3	4.5k	669k
	Esperanto (eo)	32.9	24.8	+8.1	6.5k	2.7M
	Albanian (sq)	40.0	31.9	+8.1	44.5k	2.7M
Worst	*Tamil (ta)*	7.6	7.7	-0.1	6.2k	6.2k
	Bengali (bn)	12.7	13.9	-1.2	4.6k	4.6k

Model capacity and parametrization

Without parameters from embeddings, model capacity lost

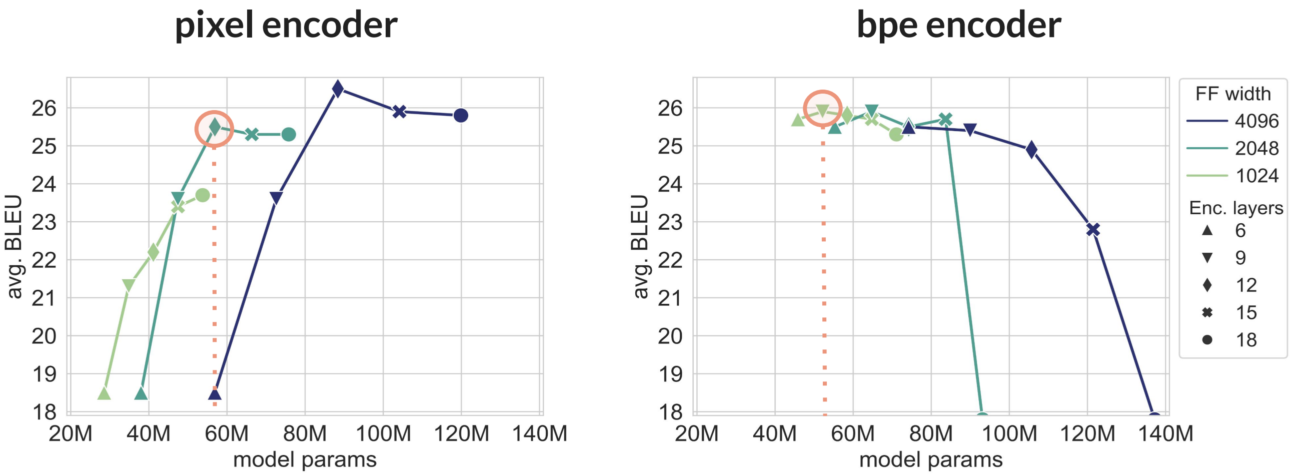
Embeddings comprise 33% of baseline model parameters



TED-7

Model capacity and parametrization

- Re-allocate parameters from the embedding matrix elsewhere for optimal performance
  - Required shift to deep-encoder + shallow-decoder for multilingual setting

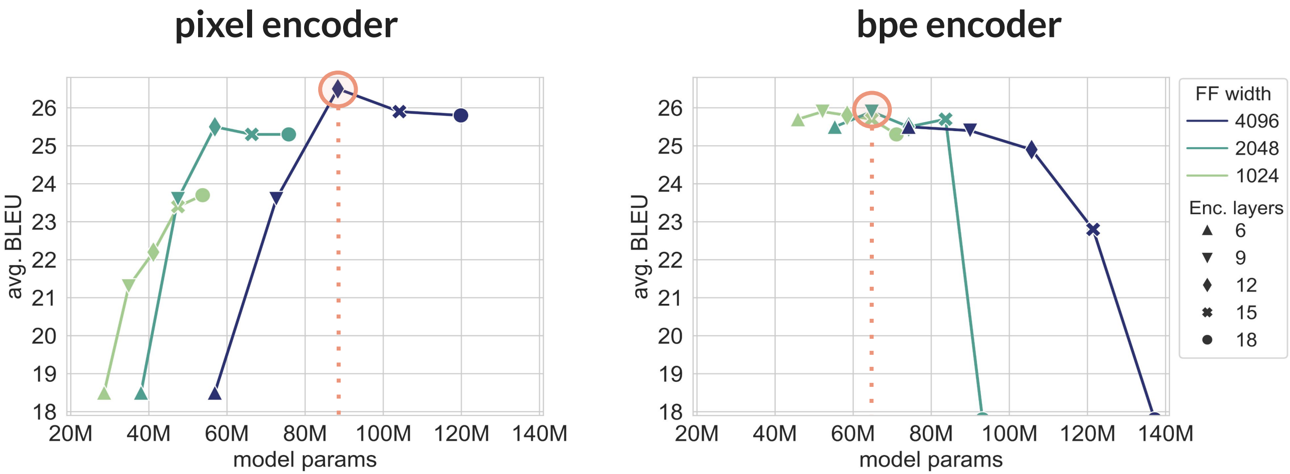


TED-7

Model capacity and parametrization

- Re-allocate parameters from the embedding matrix elsewhere for optimal performance
  - Required shift to deep-encoder + shallow-decoder for multilingual setting
  - Greater stability continuing to increase model capacity compared to baseline

Scale best architecture on TED-7 for larger datasets



TED-7

Indic case study

Language	ISO	Script	Unicode Range	Sample
English	eng	Latin	**0000-007F** ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	Universal Declaration of Human Rights
Assamese	asm	Bengali–Assamese	0000-007F ... 0900-097F **0980-09FF** 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	মানৱ অধিকাৰৰ সাৰ্বজনীন ঘোষণা
Bengali	ben	Bengali–Assamese	0000-007F ... 0900-097F **0980-09FF** 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	মানবাধিকারের সর্বজনীন ঘোষণা
Gujarati	guj	Gujarati	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F **0A80-0AFF** 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	માનવ અધિકારોની સાર્વત્રિક ઘોષણા
Hindi	hin	Devanagari	0000-007F ... **0900-097F** 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	मानव अधिकारों का सार्वजनिक घोषणापत्र
Kannada	kan	Kannada	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F **0C80-0CFF** 0D00-0D7F	ಮಾನವ ಹಕ್ಕುಗಳ ಸಾರ್ವತ್ರಿಕ ಘೋಷಣೆ
Malayalam	mal	Malayalam	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF **0D00-0D7F**	മനുഷ്യാവകാശങ്ങളുടെ സാർവത്രിക പ്രഖ്യാപനം
Marathi	mar	Devanagari	0000-007F ... **0900-097F** 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	मानवी हक्कांची सार्वत्रिक घोषणा
Odia (Oriya)	ory	Odia	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF **0B00-0B7F** 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	ମାନବିକ ଅଧିକାରର ସର୍ବଭାରତୀୟ ଘୋଷଣା |
Punjabi	pan	Gurmukhī	0000-007F ... 0900-097F 0980-09FF **0A00-0A7F** 0A80-0AFF 0B00-0B7F 0B80-0BFF 0C00-0C7F 0C80-0CFF 0D00-0D7F	ਮਨੁੱਖੀ ਅਧਿਕਾਰਾਂ ਦਾ ਵਿਸ਼ਵਵਿਆਪੀ ਐਲਾਨਨਾਮਾ
Tamil	tam	Tamil	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F **0B80-0BFF** 0C00-0C7F 0C80-0CFF 0D00-0D7F	மனித உரிமைகளின் உலகளாவிய பிரகடனம்
Telugu	tel	Telugu	0000-007F ... 0900-097F 0980-09FF 0A00-0A7F 0A80-0AFF 0B00-0B7F 0B80-0BFF **0C00-0C7F** 0C80-0CFF 0D00-0D7F	మానవ హక్కుల సార్వత్రిక ప్రకటన

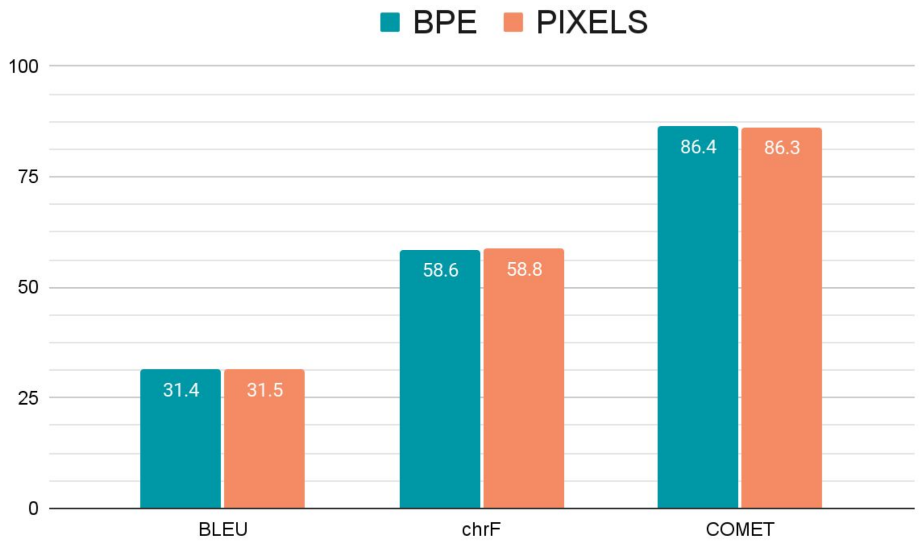
Samanantar corpus

Visually-similar scripts

Indic case study

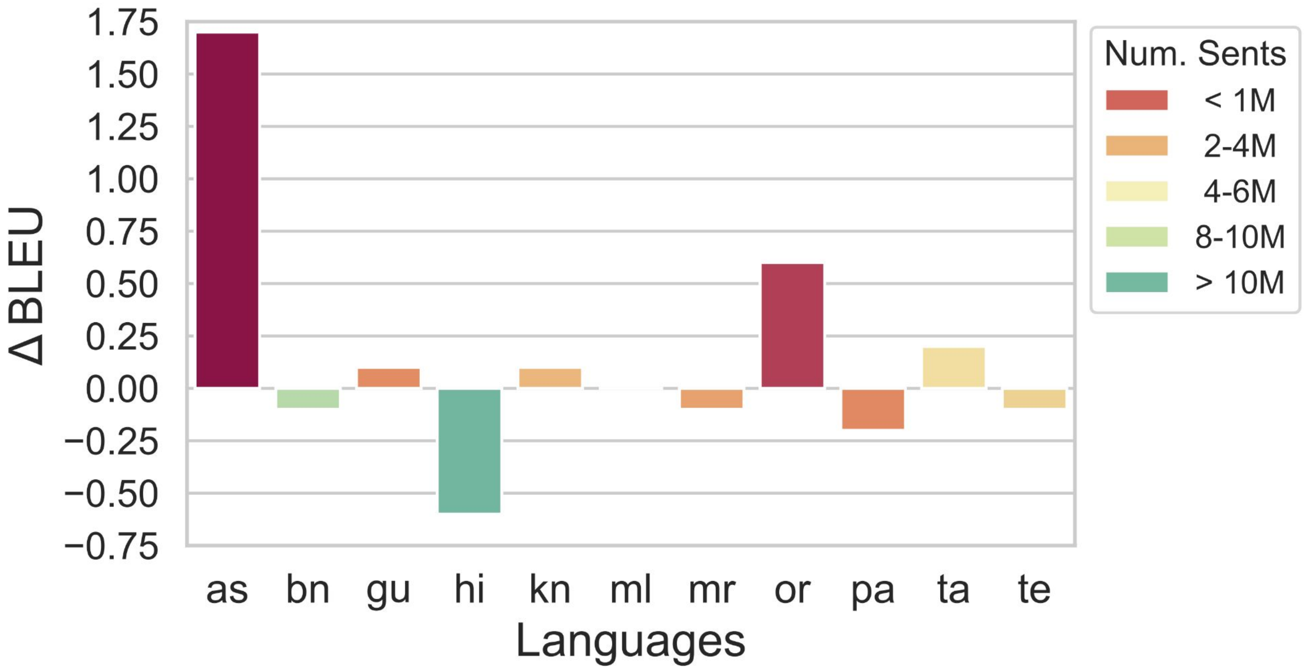
Matched performance across 3 evaluation metrics

BPE: comparing to IndicTrans (Day 1)



Indic

Indic case study



What about alternative representations?

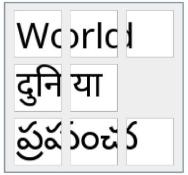
World — English, Latin script

5 characters	W	o	r	l	d
5 codepoints	H	e	l	l	o
5 bytes	57	6f	72	6c	64
1 BPE token	10603				

3 visual tokens

2 visual tokens

3 visual tokens



दुनिया — Hindi, Devanagari script

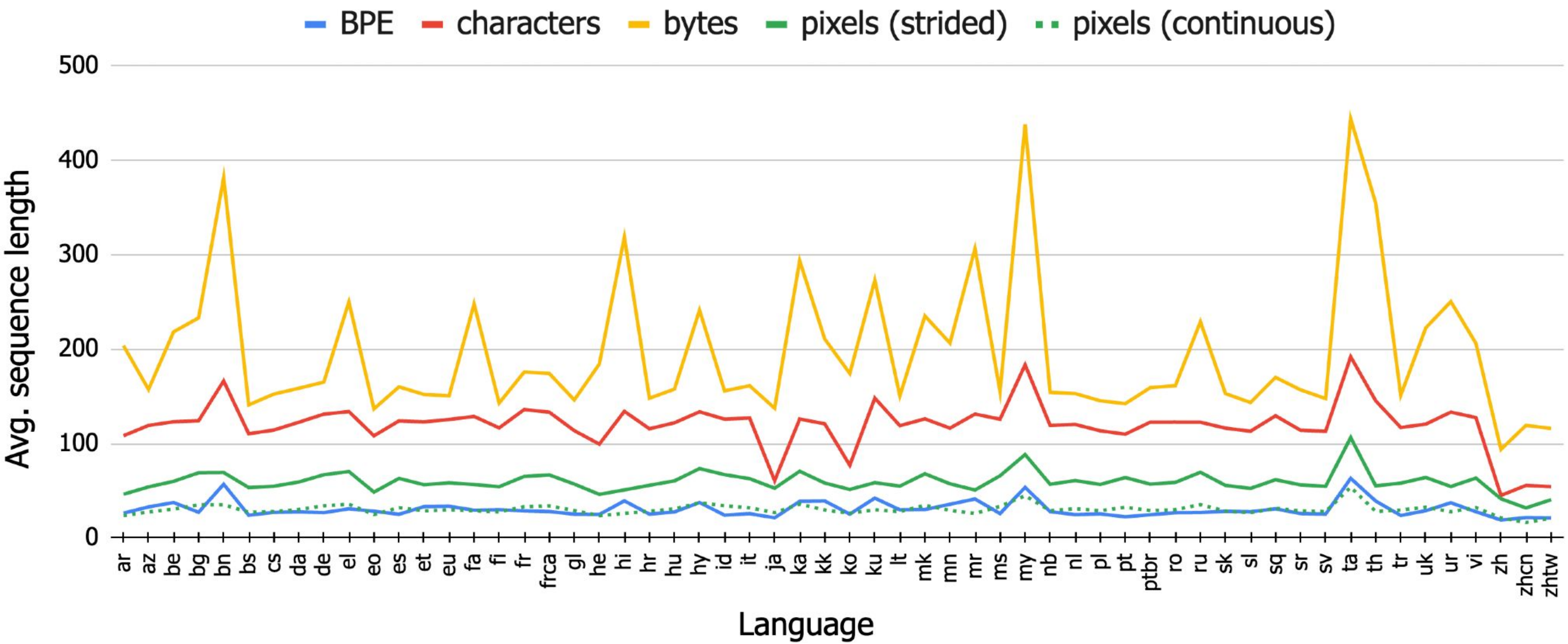
3 characters	दु						या						या					
6 codepoints	द			ु			न			ि			य			ा		
18 bytes	e0	a4	a6	e0	a5	81	e0	a4	a8	e0	a4	bf	e0	a4	bf	e0	a4	be
11 BPE tokens	11976		99	24231		223	11976		101	11976		123	11976		107	48077		

ప్రపంచం — Telugu, Telugu-Kannada script

3 characters	ప్ర									పం						చం					
7 codepoints	ప			్			ర			ప			ం			చ			ం		
21 bytes	e0	b0	aa	e0	b1	8d	e0	b0	b0	e0	b0	aa	e0	b0	82	e0	b0	9a	e0	b0	82
21 BPE tokens	156	108	103	156	109	235	156	108	108	156	108	103	156	108	224	156	108	103	156	108	224

OpenAI

What about alternative representations?



Recent work:
Ahia et al (2023)
Limisiewicz et al (2023)

Another look at cross-lingual transfer

Cross-lingual transfer

To adapt a pretrained multilingual model to a new language and script, we could…

- Simply finetune with the same vocabulary
  - With new scripts, we likely have a significant number of out-of-vocabulary tokens!
- Extend the model vocabulary (similarly to how we did this in the Section I)
  - With a strategic embedding initialization

Visual representations of text are 'vocabulary-free' — we can finetune on new languages and scripts without model extensions!

Cross-lingual transfer

We adapt our TED-7 multilingual models, which do not include all TED languages

- Finetune on 5 new language pairs with varying degrees of vocabulary coverage
- Though most scripts are ‘in-vocabulary’ there are unseen diacritics and character combinations

Language	ISO	Script seen?	Unigram Coverage	Bigram Coverage	Trigram Coverage
Romanian	ro	✓	96%	91%	84%
Polish	pl	✓	95%	88%	73%
Farsi	fa	✓	99%	79%	66%
Vietnamese	vi	✓	86%	66%	41%
Hebrew	he	✗	23%	5%	1%

TED-7

Arabic Chinese French German Japanese Korean Russian

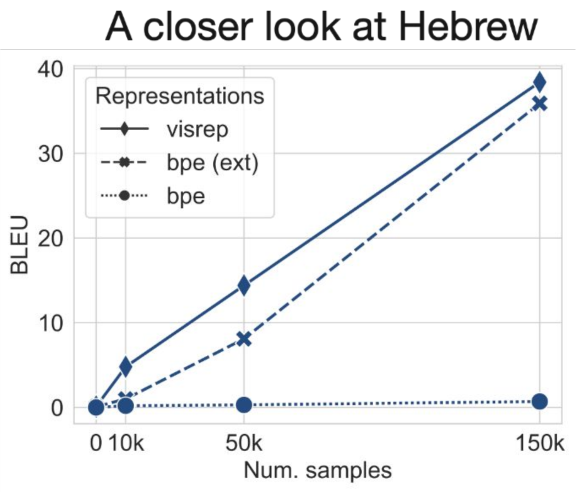
عربي, 中文, Français, Deutsch, 日本語, 한국어, русский

English

Cross-lingual transfer

More data-efficient transfer with pixels than extended-vocabulary BPE model

	BPE	*PIXEL*	*Δ*	*Script seen in training?*	*Trigram coverage*
Romanian	38.5	38.7	+0.2	✓	84%
Polish	26.4	27.4	+1.0	✓	73%
Vietnamese	25.8	27.2	+1.4	✓	66%
Farsi	23.9	26.0	+2.1	✓	41%
Hebrew	0.7	38.4	+37.7	x	1%

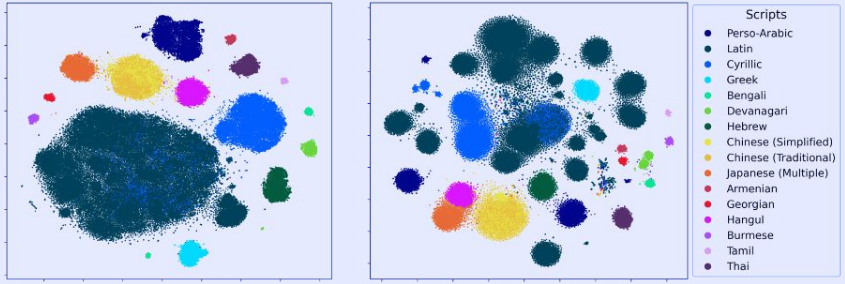


TED-7

Other analysis

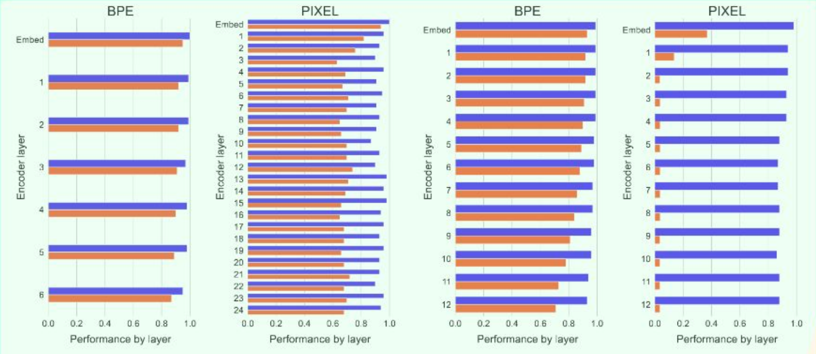
Ask at the end if interested!

Clustering by language family and script



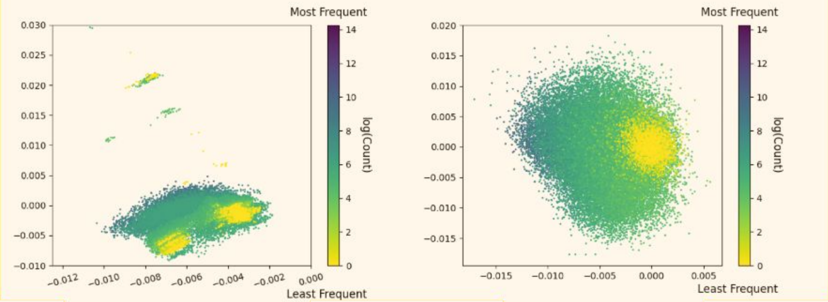
PIXEL representations | BPE embeddings

Layer-wise script and language ID



(a) Indic (b) TED-59

Reduced frequency-based representation degeneration



PIXEL representations | BPE embeddings

x

Conclusions

Conclusions

- This line of work renders text as images instead of tokenization, avoiding a fixed, finite vocabulary and the vocabulary bottleneck

- Pixel representations…
  - Are excellent on ***robustness*** tasks
  - Lead to more effective and efficient ***cross-lingual transfer***, particularly across scripts
  - Increase positive transfer in ***multilingual modeling***

- Not the end of tokenization *(or unicode!)* but perhaps a path towards more robust multilingual models for more languages

Collaborators








Questions?

Feel free to email!

elizabeth.salesky@gmail.com

Salesky et al. (2021/2023)

Rust et al. (2023)

EXTRA SLIDES

visrep monolingual

Unicode (UTF-8)

Codepoint	Byte 1	Byte 2	Byte 3	Byte 4
U+0000....007F	0xxxxxxx			
U+0080....07FF	110xxxxx	10xxxxxx		
U+0800....FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+100000..10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8 encodes codepoints in one to four bytes, determined by codepoint value.

Real-world data

“Language ID in the Wild” LREC 2020

Pred. Language	Mined “Sentence” purporting to be in this language	Noise class
Manipuri	🙋🏻‍♀️ 🙋🏽‍♂️ 🙋🏼‍♂️ 🙋🏼‍♀️ 🙋🏾‍♂️	General noise
Twi (Akan)	me: why you lyyyɪn, why you always lyyyɪn	General noise
Varhadi	Òÿáèè êè, áóðà- éÿòëÿðÿ ìàëèê îëàí Éàãóá ìöÿëëèì òÿïÿ- ÿäÿá-ÿðêàíû èëÿ áó íÿñë [...]	Misrendered PDF
Aymara	Orilyzewuhubys ukagupixog axiqyh asozasuh uxilutidobyq osoqalelohan [...]	Non-Unicode font
Balinese	As of now [illegible] is verified profile on Instagram.	Boilerplate
Cherokee	❝ALL mᎩ ᏆhᎾRᏁs ᏀREW bACK As fLᎾWERs❞ · · · SWEET ᏰᏘᏰIᏋᏕ n DᏌᏩᏕ	Creative use of Unicode
Oromo	My geology **essa**y introduction **essa**y on men authoring cro**ss**words	Unlucky frequent n-gram
Pular	MEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEOW	Repeated n-grams
Chechen	Ж и р н о в с к и й ... Ж и р н о в с к и й р а й о н н ы й Ф е с т и в а л ь Т О С о в	A N T S P E A K
Kashmiri	सा.	Short/ambiguous
Nigerian Pidgin	This new model features a stronger strap for a secure fit and increased comfort.	High-resource cousin
Uyghur	نۇرسۇلتان نازاربايەۆ قىتايدىڭ قازاقستانداعى ەلشىسىمەن	Out-of-model cousin
Dimli	The S<b class='b2'>urina<b class='b1'>m toa<b class='b3'>d is [...]	Deliberately Obfuscated

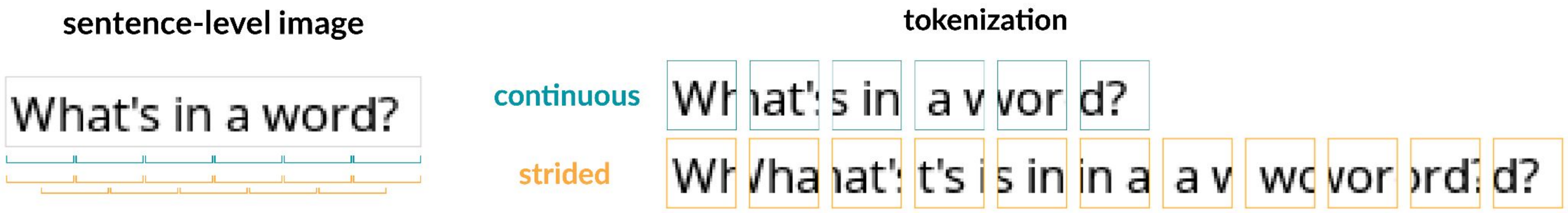
“Quality at a Glance” TACL 2022

Error Codes	
X: *Incorrect translation, but both correct languages*	
`en` A map of the arrondissements of Paris	`kg` Paris kele mbanza ya kimfumu ya Fwalansa.
`en` Ask a question	`tr` Soru sor Kullanıma göre seçim
WL: *Source OR target wrong language, but both still linguistic content*	
`en` The ISO3 language code is zho	`zza` Táim eadra bracach mar bhionns na frogannaidhe.
`en` Der Werwolf—sprach der gute Mann,	`de` des Weswolfs, Genitiv sodann,
NL: *Not a language: at least one of source and target are not linguistic content*	
`en` EntryScan 4 _	`tn` TSA PM704 _
`en` organic peanut butter	`ckb` ◆◆◆◆◆◆◆

Computational efficiency

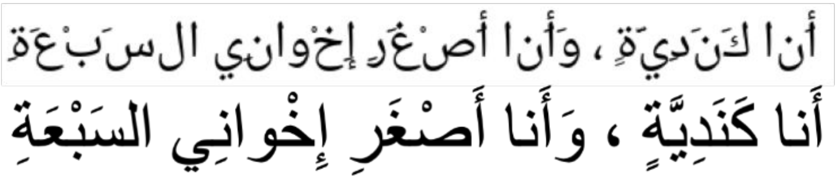
- Rendering and tokenization
  - Rendering with PangoCairo lies between Python and Rust BPE implementations
  - Time is ~1.2x time to learn and apply subword tokenization, though the Rust implementation can be scaled with batching
- Training time
  - Dependent on sequence lengths, model operations, vocab & softmax sizes
  - Training time is ~1.4x of equivalent subword models
- Inference time
  - No significant differences at inference time
- Disk storage
  - Raw and binarized images take significantly more disk space to store (400x)
  - Rendering on the fly preferable, toolkit allowing

Tokenizing rendered text



Why do we render the whole sentence?

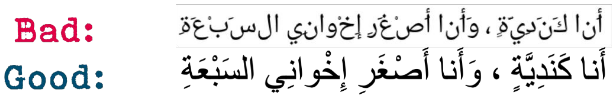
As opposed to by character or word



Arabic

Pixel representations: considerations

- Why do we render the whole sentence?
  - As opposed to say, rendering each character or word

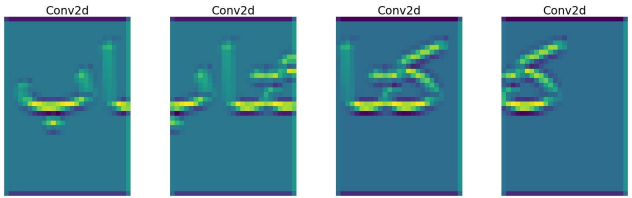


- Two reasons: ① ***rendering correctness*** and ② ***tokenization-free modeling***
  - ① Many scripts have contextual forms and require context to render correctly
    - For example, Arabic characters may appear differently in isolation than in context
    - Rendering diacritics individually would result in strange visual forms!
  - ② Avoids predetermining a discrete segmentation
    - What is the ‘correct’ segmentation for English newstext? For twitter? For Chinese or non-whitespace marking languages? For a morphologically rich language like Kinyarwanda?

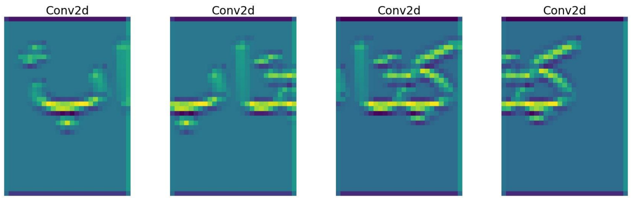
Convolutional filter visualization

- Visual representations have direct access to token components
  - Similar representations for word forms with and without diacritics
  - If a visual text model sees a partial match in training, both will be updated by backprop

كتاب



كِتَابٌ



Normalization

- **What about normalization as preprocessing?**
  - It helps text models, but selectively!
- **While spell-checking helps, it:**
  - is language-specific
  - is best suited to observed noise
  - relies on context to disambiguate:
    - noisy context hurts!

		Arabic		French		German		Korean		Russian	
		BPE	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*
	no noise	**32.1**	31.6	**36.7**	36.2	33.6	**35.1**	**17.0**	16.6	**25.4**	25.0
swap	induced noise	2.3	**9.3**	2.4	**22.0**	1.9	**25.9**	5.4	**8.9**	5.4	**18.8**
	+ *spellcheck*	7.9	**11.9**	23.8	**29.1**	1.9	**14.1**	5.1	**6.9**	10.8	**18.2**
cambridge	induced noise	7.8	**13.2**	6.9	**18.3**	6.5	**16.9**	12.6	**14.1**	4.5	**11.1**
	+ *spellcheck*	10.9	**12.6**	16.4	**21.1**	10.0	**14.9**	10.3	**11.8**	5.9	**11.1**
l33tspeak	induced noise	—	—	0.3	**0.7**	0.7	**1.2**	—	—	—	—
	+ *spellcheck*	—	—	0.3	**0.7**	0.7	**1.2**	—	—	—	—
diacritics	induced noise	1.7	**25.2**	—	—	—	—	—	—	—	—
	+ *spellcheck*	2.1	**25.3**	—	—	—	—	—	—	—	—
unicode	induced noise	—	—	—	—	—	—	—	—	1.6	**22.0**
	+ *spellcheck*	—	—	—	—	—	—	—	—	2.1	**20.4**

Table 11: Translation performance on five types of induced noise with spellchecking as preprocessing; all test sets have noise induced with $p = 1.0$. Both traditional text models *(BPE)* and visual text models *(visrep)* are shown. We bold the best model for each condition.

Appendix C

Normalization

Noise, with and without spellcheck

Not a perfect fix!

- What do we see?
  - Spellcheck generally helps BPE models…
    - but also visrep models!
- Spellcheck doesn't help all languages equally
  - See: German BPE vs French BPE, swap
- Spellcheck doesn't help all noise equally
  - See: l33tspeak
- Spellcheck can also *create* errors

		Arabic		French		German		Korean		Russian	
		BPE	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*	*BPE*	*visrep*
	no noise	**32.1**	31.6	**36.7**	36.2	33.6	**35.1**	**17.0**	16.6	**25.4**	25.0
swap	induced noise	2.3	**9.3**	2.4	**22.0**	1.9	**25.9**	5.4	**8.9**	5.4	**18.8**
	+ spellcheck	7.9	**11.9**	23.8	**29.1**	1.9	**14.1**	5.1	**6.9**	10.8	**18.2**
cambridge	induced noise	7.8	**13.2**	6.9	**18.3**	6.5	**16.9**	12.6	**14.1**	4.5	**11.1**
	+ spellcheck	10.9	**12.6**	16.4	**21.1**	10.0	**14.9**	10.3	**11.8**	5.9	**11.1**
l33tspeak	induced noise	—	—	0.3	**0.7**	0.7	**1.2**	—	—	—	—
	+ spellcheck	—	—	0.3	**0.7**	0.7	**1.2**	—	—	—	—
diacritics	induced noise	1.7	**25.2**	—	—	—	—	—	—	—	—
	+ spellcheck	2.1	**25.3**	—	—	—	—	—	—	—	—
unicode	induced noise	—	—	—	—	—	—	—	—	1.6	**22.0**
	+ spellcheck	—	—	—	—	—	—	—	—	2.1	**20.4**

Table 11: Translation performance on five types of induced noise with spellchecking as preprocessing; all test sets have noise induced with $p = 1.0$. Both traditional text models *(BPE)* and visual text models *(visrep)* are shown. We bold the best model for each condition.

Appendix C

Subword regularization

- Subword regularization techniques often improve performance and robustness
  - *Are the improvements similar to with visual text representations?*

- Recall BPE:

```
u-n-r-e-l-a-t-e-d
u-n re-l-a-t-e-d
u-n re-l-at-e-d
u-n re-l-at-ed
un re-l-at-ed
un re-l-ated
un rel-ated
un-related
unrelated
```

BPE-dropout:

```
u-n_r-e-l-a_t-e_d       u-n-r-e-l-a_t-e-d
u-n re-l_a-t-e_d        u_n re_l-a-t-e-d
u-n re_l-at-e_d         u_n re-l-at-e-d
un re-l-at-e-d          u_n re-l-ate_d
un re_l-at-ed           u_n rel-ate-d
un re-lat-ed            u_n relate_d
un relat_ed
```

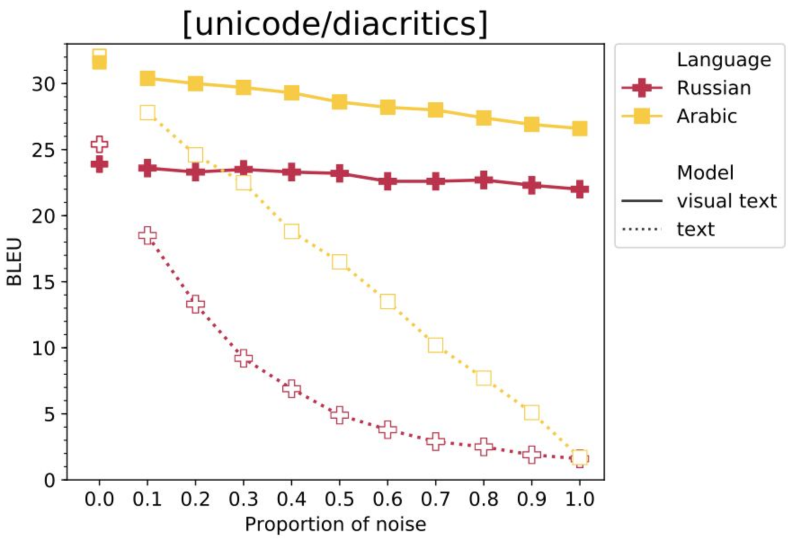
Different subword set with the same (overall) number of merges

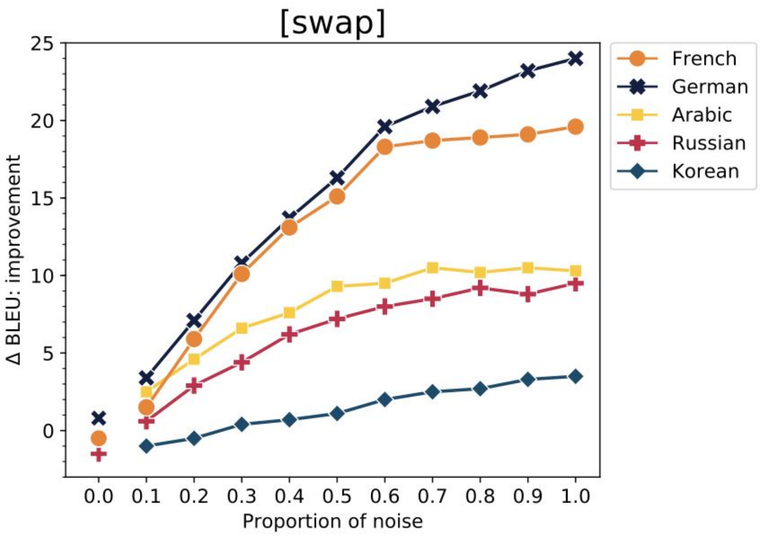
Subword regularization

- BPE-Dropout (Provilkov et al. 2020):
  - *Subword segmentation using BPE algorithm*
  - *'Drop' candidate merges with some probability, and train with different segmentations each epoch*
    - *NOTE: small number of resulting subwords will not be in the MT model's vocabulary*

- Subword Regularization (Kudo, 2018):
  - *Subword segmentation using unigram LM probabilities*
  - *Can draw a stack of ℓ candidates, and use different candidate segmentations each epoch*
    - *{ _hell o, _h ello, _he llo, _h e l l o, _h el l o }*

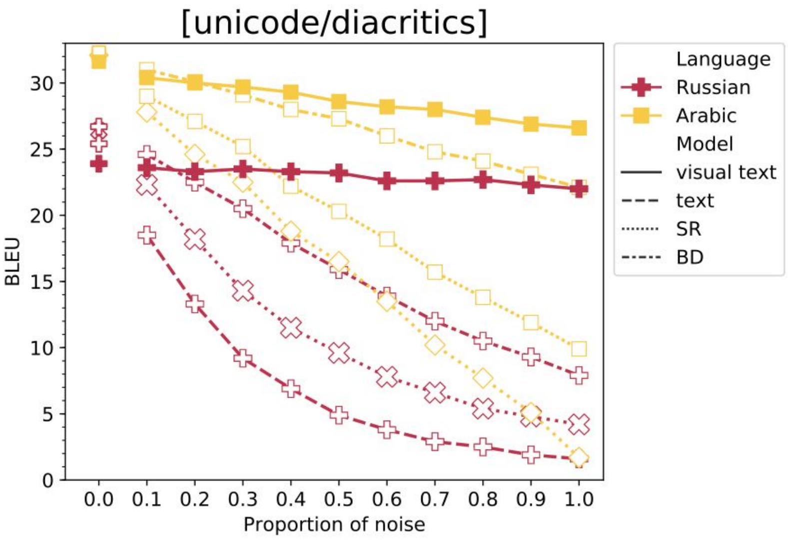
Subword regularization

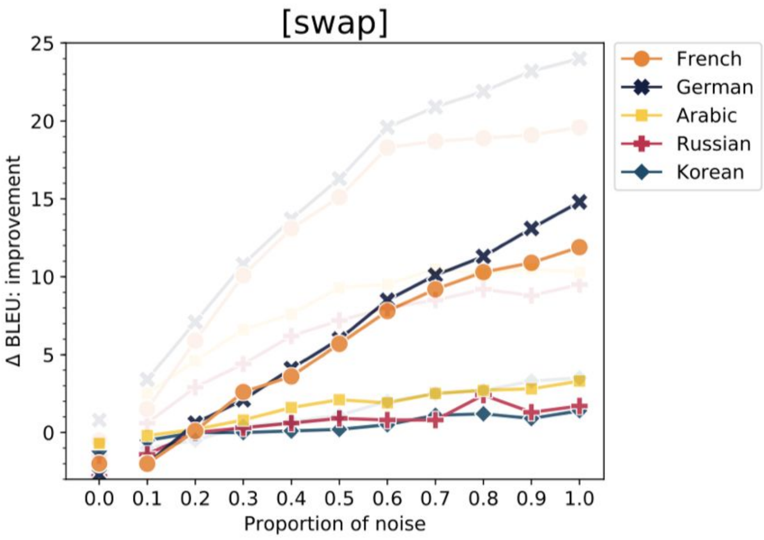




Improvement over standard BPE model

Subword regularization





Improvement over stronger BPE dropout baseline,
compared to over standard BPE model (background)

Hyperparameters

AR–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	30.8	30.0	30.6	31.3	30.4	30.5	29.6
10	30.3	28.5	31.0	31.6	31.4	31.4	30.4
15		25.2	30.2	30.3	30.6	29.4	29.3

DE–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	0.7	32.6	35.1	0.5	33.1	33.9	32.5
10	0.6	34.6	34.8	32.8	32.9	34.4	33.5
15		32.8	33.9	32.0	31.4	33.7	33.9

FR–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	35.4	35.7	35.7	35.5	0.7	0.6	0.8
10	35.6	36.2	36.1	36.1	34.7	34.7	35.0
15		35.7	35.8	35.6	34.4	34.3	34.6

JA–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	12.4	11.5	12.3	13.1	12.4	12.4	12.3
10	11.8	11.8	12.4	12.5	11.5	12.4	12.3
15		9.4	12.1	12.7	12.2	12.4	12.1

KO-EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	15.8	15.7	15.3	16.2	15.6	16.0	16.1
10	14.7	15.9	15.5	16.5	14.7	15.9	16.4
15		14.3	15.2	15.4	15.7	16.2	15.6

RU–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	0.6	22.7	23.8	0.5	23.6	23.0	0.5
10	2.0	23.2	25.0	23.2	23.2	23.9	23.2
15		21.1	24.4	23.7	24.5	24.2	22.0

ZH–EN	$c=1,\ font=10pt$						
$s\downarrow/w\rightarrow$	10	15	20	25	30	35	40
5	16.7	0.4	16.7	17.3	17.4	17.0	0.4
10	15.8	17.1	16.8	17.1	16.3	17.0	0.4
15		16.0	16.0	16.3	16.4	16.3	0.5

Varied conv. kernel size (note: 23=full window height).

$h \times w$	3×3	3×1	1×3	13×3	23×3	5×5
ZH–EN	17.4	17.1	16.9	16.7	0.6	16.6

w = window size, s = stride, c = number of convolutional blocks

Ablations: visrep without changing segmentation

subword-aligned tokenization

Ablations: visrep without changing segmentation

CLEAN:			ar	de	fr	ja	ko	ru	zh
Visual text			31.6	35.1	36.2	13.1	16.6	25.0	17.6
+ *subword-ali*			25.2	28.6	29.6	6.5	11.5	19.9	7.8
Text, subwords			32.1	33.6	36.7	14.4	17.0	25.4	18.3
NOISED:									
Visual text	`visual`	`p=0.5`	28.4	5.2	6.1	—	—	23.2	—
+ *subword-ali*	`visual`	`p=0.5`	16.7	5.1	5.3	—	—	16.6	—
Text, subwords	`visual`	`p=0.5`	16.5	2.7	2.5	—	—	4.9	—
Visual text	`perm`	`p=0.5`	21.7	29.4	28.4	—	11.5	18.3	—
+ *subword-ali*	`perm`	`p=0.5`	13.9	15.2	15.3	—	8.2	12.6	—
Text, subwords	`perm`	`p=0.5`	12.4	13.1	13.3	—	10.8	11.1	—

Ablations: sliding segmentation without visrep

Character trigrams contain approximately the same amount of text as a visrep sliding window

Here, we compare character n-grams to BPE and visrep, as approx. the same segmentation as visrep, but without a visual component

MODEL:	ar	de	fr	ja	ko	ru	zh
Visual text	31.6	35.1	36.2	13.1	16.6	25.0	17.6
w/o visrep (char n-grams)	31.5	34.6	36.4	1.4	1.3	24.6	5.5
Text, BPE	32.1	33.6	36.7	14.4	17.0	25.4	18.3
NOISED:							
Visual text; `swap p=0.5`	21.7	29.4	28.4	—	11.5	18.3	—
w/o visrep; `swap p=0.5`	11.2	10.8	11.9	—	1.1	9.5	—
Text, BPE; `swap p=0.5`	12.4	13.1	13.3	—	10.8	11.1	—

Ablations: sliding segmentation without visrep

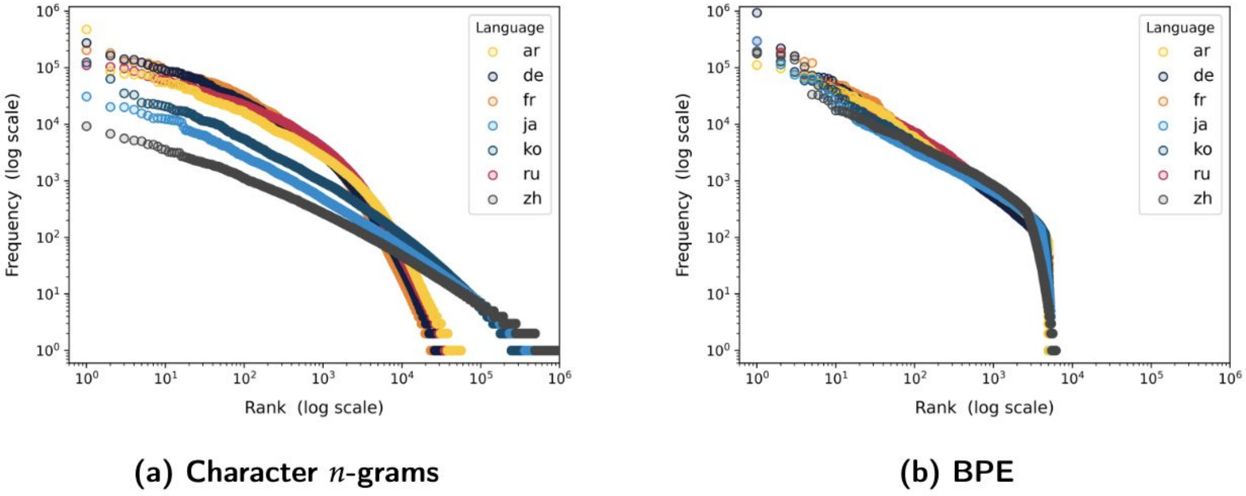


Figure 4-15. The rank–frequency distribution compared between sliding window segmentation (≈character n-grams) and BPE.

Hybrid representations: BPE & visrep

Hybrid representations: BPE & visrep

	Individual		Aligned	Multimodal		
Languages	Subword	Vistext	$\text{Vistext}_{pretrained}$	ADD	AVG	CONCAT
de-en	33.6	35.1	33.3	33.6	34.7	34.3
fr-en	36.7	36.2	35.3	36.1	36.7	36.4
ja-en	14.4	13.1	10.8	15.0	15.2	14.7
ko-en	17.0	16.6	15.8	17.8	18.0	17.0
zh-en	18.3	17.4	17.5	18.1	18.6	18.2

Table D-I. Comparing individual subword or visual text representations to multimodal inputs which combine both subword and visual text with various operations. Translation performance shown in BLEU on the TED dataset.

Subword regularization for visual text

Subword regularization for visual text

Method	Lang	Clean text$_{TED}$				Noisy text$_{MTNT}$			
		30	25	20	15	30	25	20	15
default	de	34.0	34.8	35.1	34.6	20.5	20.0	21.1	20.9
	ja	12.4	13.1	12.4	11.8	4.6	5.2	4.7	3.5
resizing	de	34.1	—	—	—	21.2	—	—	—
	ja	12.6	—	—	—	5.1	—	—	—
padding	de	31.8	—	—	—	16.6	—	—	—
	ja	0.4	—	—	—	0.1	—	—	—

Morphological generalization

Test suite for morphological phenomena in MT (Amrhein and Sennrich, 2021)

Compounding (German)	**Circumfixation** (Chicasaw)	**Infixation** (Bontoc)	**Vowel Harmony** (Turkish)	**Reduplication** (Itza')
Schild , Kröte 'shield' , 'toad'	lakna 'it is yellow'	fikas 'strong'	üzüldün 'you are sad'	tz'eek 'few'
Schildkröte 'turtle'	iklakno 'it isn't yellow'	fumikas 'to be strong'	mutlusun 'you are happy'	tz'eek-tz'eek 'very few'

Morphological generalization

word-level accuracy

Phenomena		Freq.	Tokenization / Representation				
			BPE 32K	BPE_{drop} 32K	BPE_{drop} 500	CHAR	VISREP
Compounding	#9	27	0	0	0	0	0
	#7	67	46.1	0	83.8	0	0
	#5	238	98.1	97.6	96.2	97.0	97.3
	#3	522	98.9	98.4	97.3	96.5	98.7
	#1	1,095	96.2	97.8	97.3	97.0	98.1
Circumfixation	#4	11,718	97.9	97.9	97.9	95.9	99.0
	#2	26,007	100.0	98.0	98.0	99.2	99.2
Infixation	#4	3,796	98.9	98.9	96.7	100.0	99.7
	#3	15,540	98.5	96.4	99.3	97.1	98.7
Vowel Harmony	#3	8,636	98.9	99.4	97.7	98.3	98.9
Reduplication	Triple	106	0	0	0	0	0
	Partial	34,783	94.2	95.0	95.9	95.0	97.1

visrep multilingual

Vocabulary and script coverage

Let’s take a closer look at what it means for a script to be ‘covered’ by a model…

- Though most scripts are ‘in-vocabulary’ there are unseen diacritics and character combinations

Language	ISO	Script seen?	Unigram Coverage	Bigram Coverage	Trigram Coverage
Romanian	ro	✓	96%	91%	84%
Polish	pl	✓	95%	88%	73%
Farsi	fa	✓	99%	79%	66%
Vietnamese	vi	✓	86%	66%	41%
Hebrew	he	✗	23%	5%	1%

Arabic Chinese French German Japanese Korean Russian

عربي, 中文, Français, Deutsch, 日本語, 한국어, русский

English

TED-7

Data-efficient cross-lingual transfer

- We see increased improvements compared to BPE the more distant the language and/or script is to those observed in training

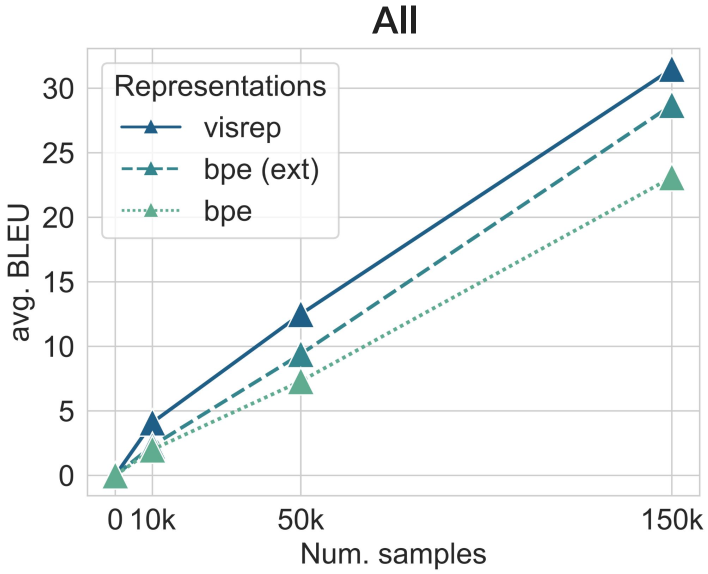
Language	ISO	Script seen?	Unigram Coverage	Bigram Coverage	Trigram Coverage	$\frac{\Delta}{\text{BPE}_{extend}}$	$\frac{\Delta}{\text{BPE}}$
Romanian	ro	✓	96%	91%	84%	+11%	+11%
Polish	pl	✓	95%	88%	73%	+13%	+14%
Farsi	fa	✓	99%	79%	66%	+18%	+20%
Vietnamese	vi	✓	86%	66%	41%	+22%	+21%
Hebrew	he	✗	23%	5%	1%	+30%	+4700%

Data-efficient cross-lingual transfer

- We see increased improvements compared to BPE the more distant the language and/or script is to those observed in training
- Better transfer performance with limited examples fine-tuning to new languages & scripts

	# Samples	ro	pl	fa	vi	he
PIXEL	0	0.3	0.1	0.2	0.4	0.1
	10k	5.5	3.9	3.7	3.9	4.8
	50k	16.6	11.7	11.1	11.7	14.4
	150k	38.7	27.4	26.0	27.2	38.4
BPE_{extend}	0	0.2	0.1	0.2	0.2	0.1
	10k	2.9	2.7	2.5	2.3	1.0
	50k	12.3	9.7	8.5	8.6	8.1
	150k	38.5	26.4	23.9	25.8	30.9
BPE	0	0.2	0.1	0.2	0.2	0.0
	10k	2.9	2.7	2.4	1.9	0.2
	50k	12.1	9.1	7.7	7.5	0.3
	150k	38.3	26.3	23.7	24.1	0.7

(Language spans columns ro–he.)

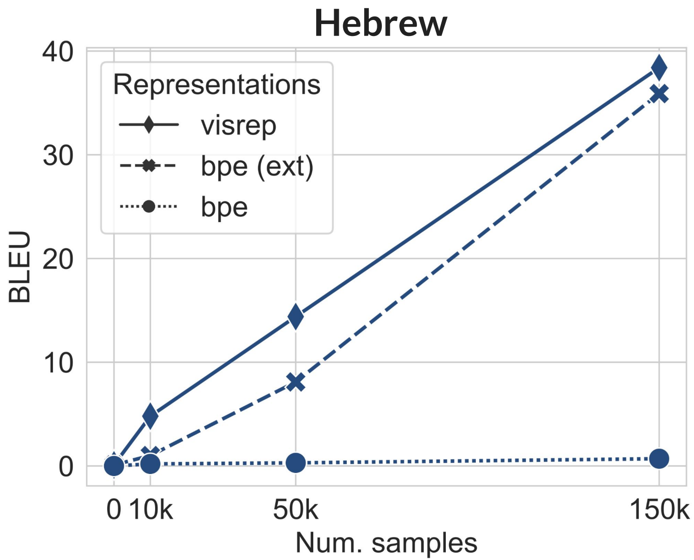


Data-efficient cross-lingual transfer

- We see increased improvements compared to BPE the more distant the language and/or script is to those observed in training
- Better transfer performance with limited examples fine-tuning to new languages & scripts

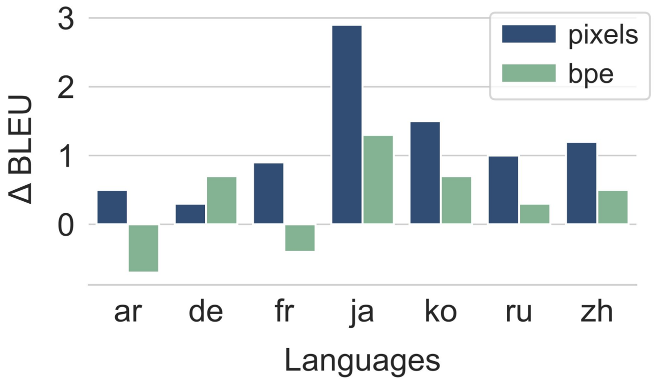
	# Samples	ro	pl	fa	vi	he
PIXEL	0	0.3	0.1	0.2	0.4	0.1
	10k	5.5	3.9	3.7	3.9	4.8
	50k	16.6	11.7	11.1	11.7	14.4
	150k	38.7	27.4	26.0	27.2	38.4
BPE_{extend}	0	0.2	0.1	0.2	0.2	0.1
	10k	2.9	2.7	2.5	2.3	1.0
	50k	12.3	9.7	8.5	8.6	8.1
	150k	38.5	26.4	23.9	25.8	30.9
BPE	0	0.2	0.1	0.2	0.2	0.0
	10k	2.9	2.7	2.4	1.9	0.2
	50k	12.1	9.1	7.7	7.5	0.3
	150k	38.3	26.3	23.7	24.1	0.7

Language (column group header over ro, pl, fa, vi, he)



Reduced interference across languages

- Comparing multilingual models to models for each language pair…
  - No degradation for any language pairs compared to bilingual models with pixels
  - *Have not run the equivalent for TED-59, Indic*



"Curse of multilinguality" ?

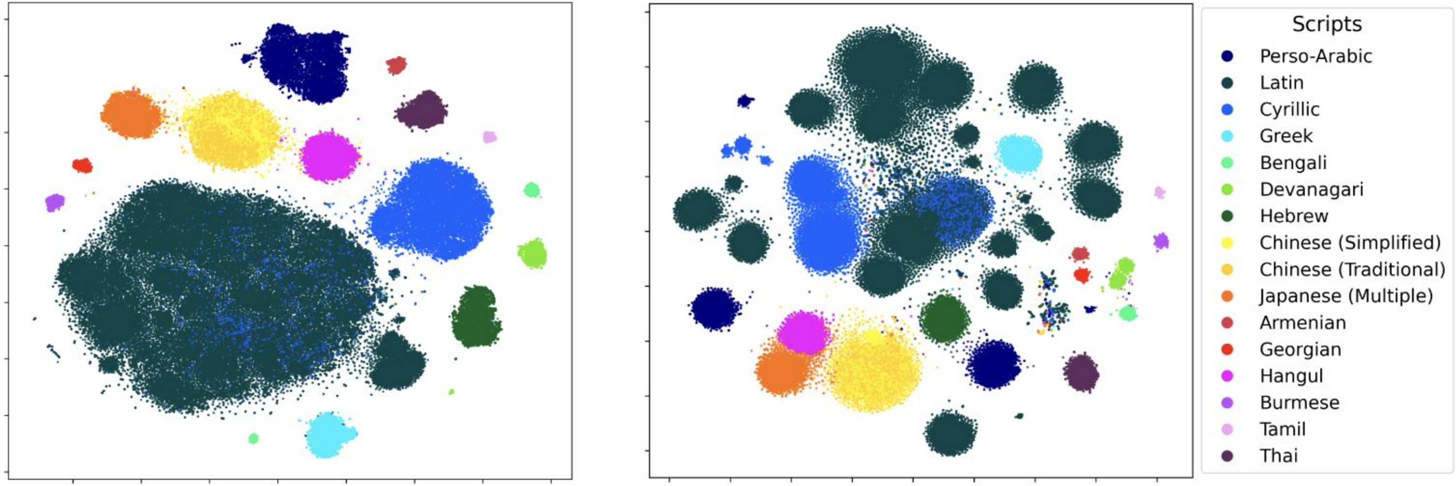
TED-7

Comparison to prior work

	Language	ISO	Script	# Sents	Aharoni.	BPE	PIXEL	Δ
LOW-RESOURCE	Azerbaijani	az	Latin	5,946	11.2	12.5	16.6	+5.4
	Belarusian	be	Cyrillic	4,509	18.3	19.2	28.5	+10.2
	Galician	gl	Latin	10,017	28.6	29.7	36.5	+7.9
	Slovak	sk	Latin	61,470	26.8	27.4	33.7	+6.9
				Avg:	21.2	22.2	28.8	+7.6
HIGH-RESOURCE	Arabic	ar	Arabic	214,111	25.9	26.1	29.8	+3.9
	German	de	Latin	167,888	28.9	30.0	36.1	+7.2
	Hebrew	he	Hebrew	211,819	30.2	30.7	35.3	+5.1
	Italian	it	Latin	204,503	32.4	32.3	38.5	+6.1
				Avg:	29.4	29.8	34.9	+5.6

Results in BLEU for 4 high-resource and 4 low-resource language pairs reported in prior work

Clustering by script

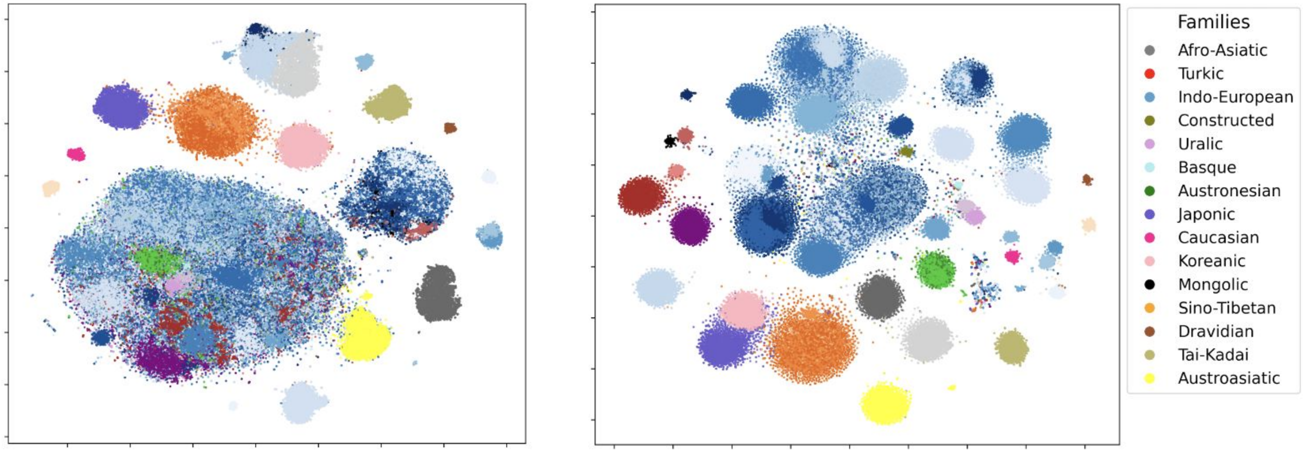


(a) PIXEL REPRESENTATIONS, clustered by script

(b) SUBWORD EMBEDDINGS, clustered by script

TED-59

Clustering by language family



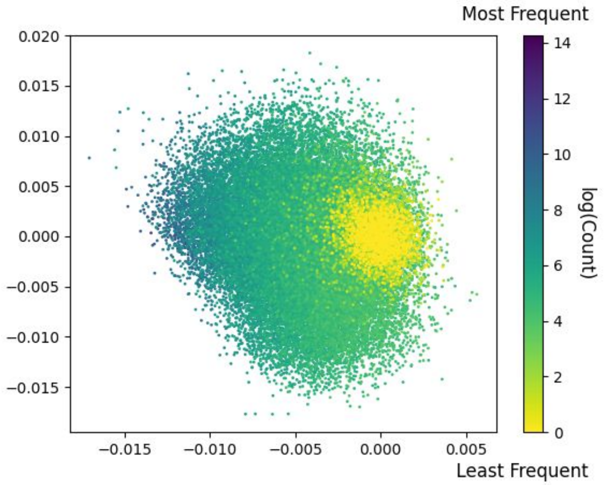
(c) PIXEL REPRESENTATIONS, clustered by family

(d) SUBWORD EMBEDDINGS, clustered by family

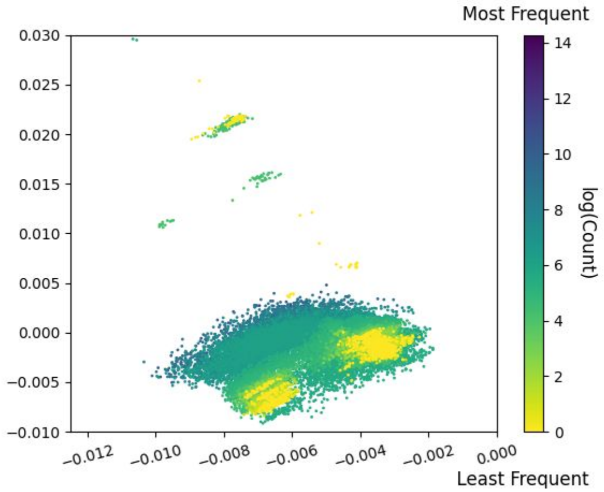
TED-59

Frequency effects

BPE

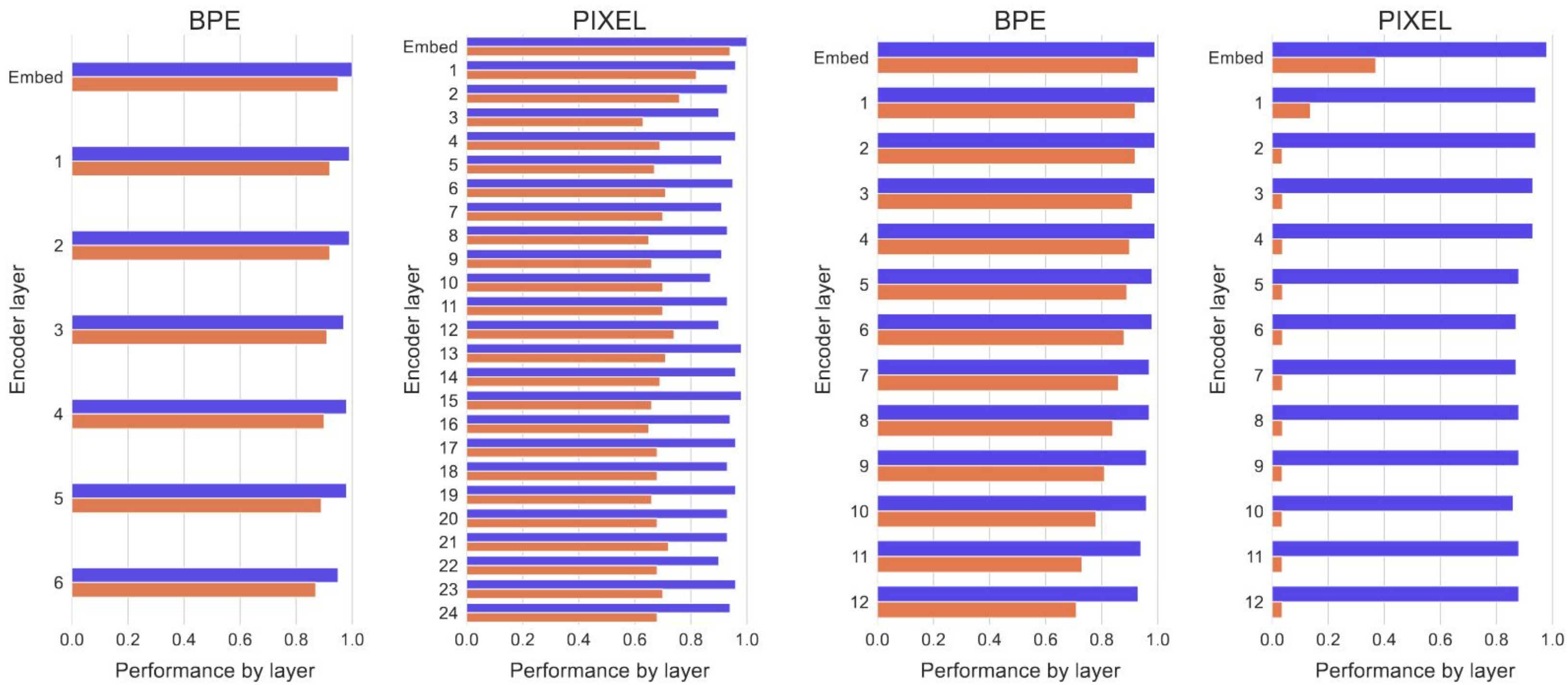


Pixels



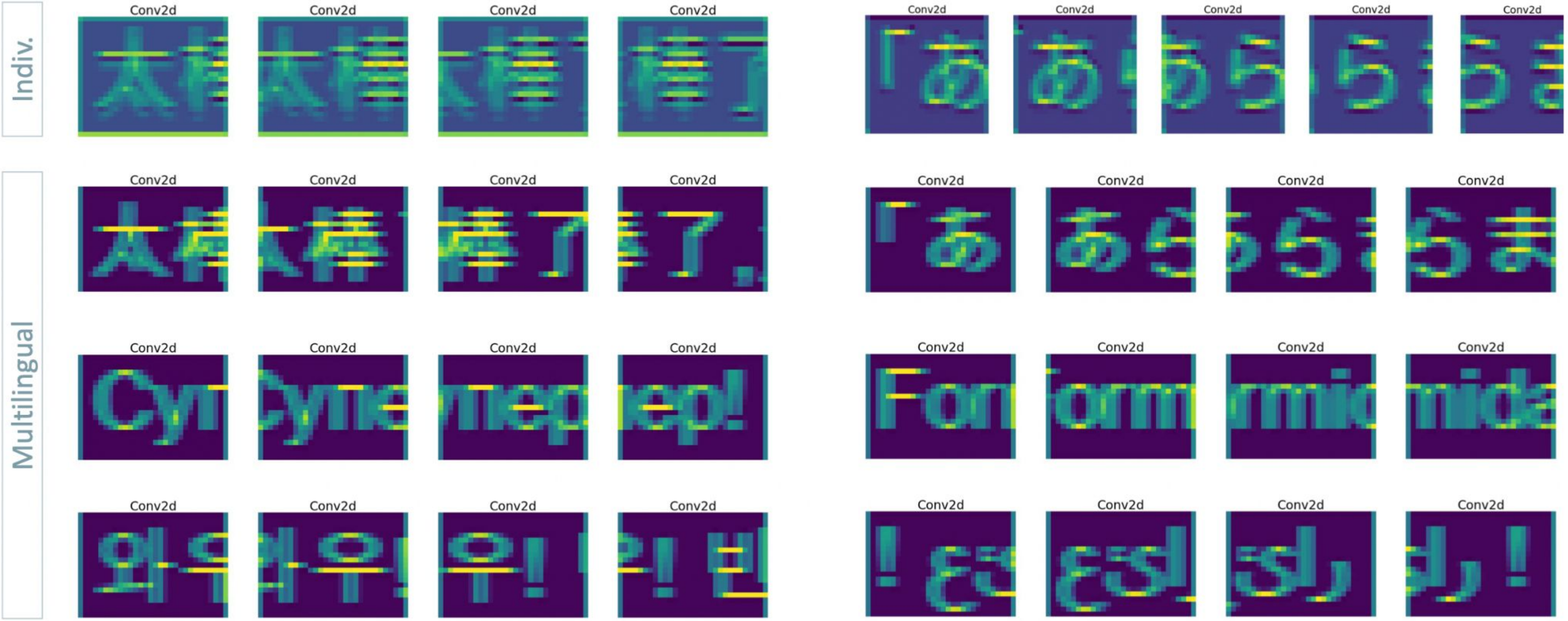
TED-7

Layer-wise script and language information



(a) Indic (b) TED-59

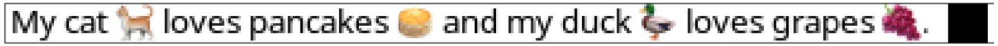
Convolutional filter visualization



PIXEL

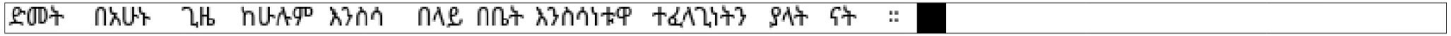
Flexible rendering

- Emojis and mixed font ranges:



- Left-to-right, right-to-left, and logosyllabic writing systems:

- Word-level rendering



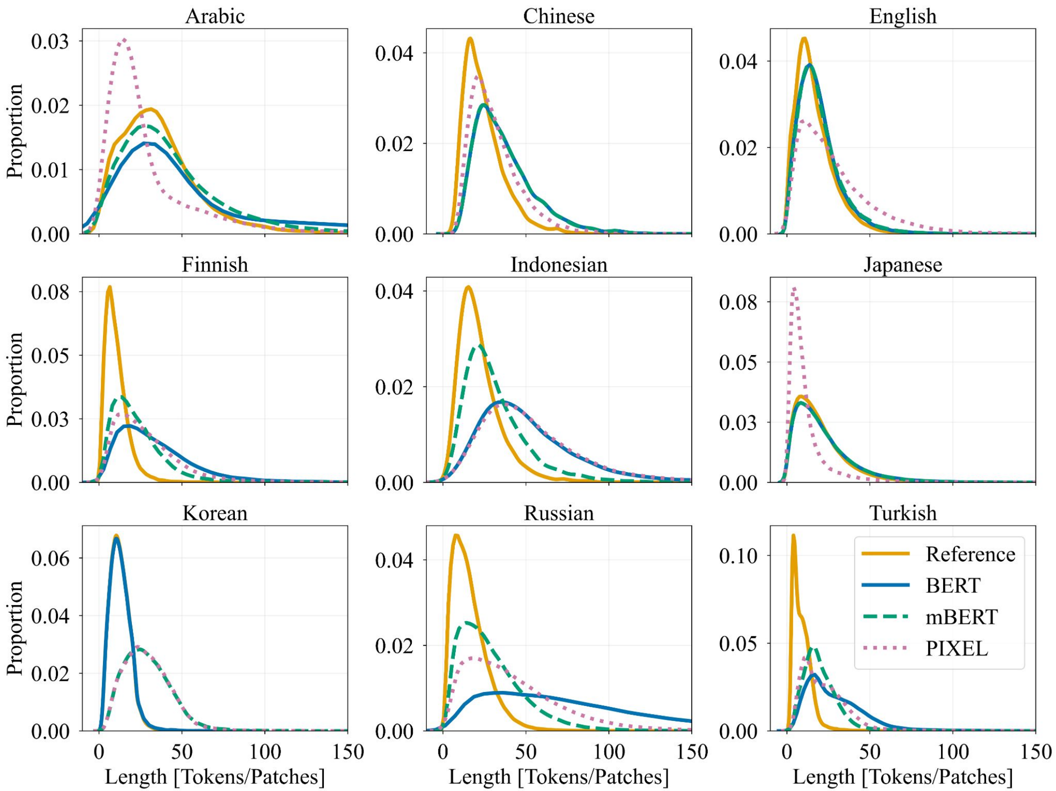
PangoCairo renderer: can mix fonts within a sequence, rendering speed comparable to HuggingFace BPE implementations

Rendering speed

Processor	Batched	Throughput [ex / s] ENG	Throughput [ex / s] ZHO
Renderer (Grayscale)	✗	3944.1	6309.0
Renderer (RGB)	✗	3615.1	6849.5
Tokenizer (Rust)	✓	**19128.9**	**18550.5**
	✗	4782.9	5684.4
Tokenizer (Python)	✓	1286.6	2637.1
	✗	1286.8	2580.9

PangoCairo renderer: can mix fonts within a sequence, rendering speed comparable to HuggingFace BPE implementations

PIXEL sequence lengths



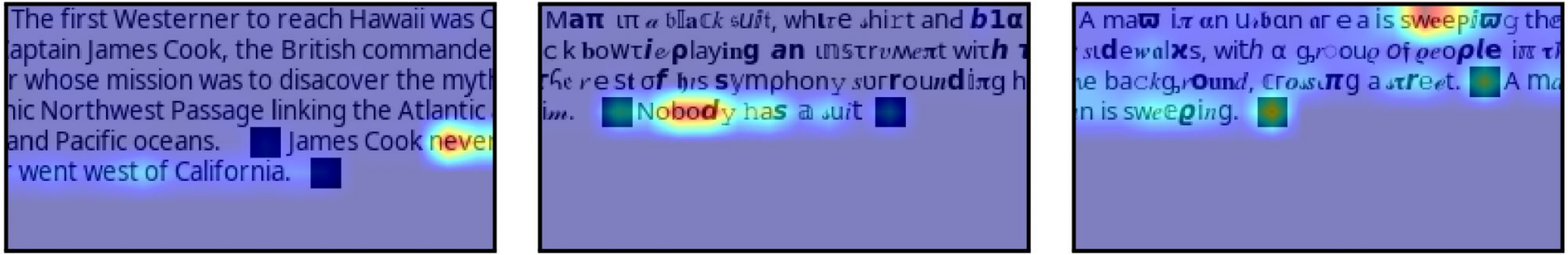
Evaluating against Äḋʊɜrsarīał aṯtacḳṣ

How well does PIXEL deal with visually similar attacks?

Attack	Sentence
NONE	Penguins are designed to be streamlined
CONFUSABLE	Peπguɯns aɾe ɗesignɛɗ to be ꞩtreaMlɪnɛd
SHUFFLE (INNER)	Pegnuins are dnesiged to be sieatrnmled
SHUFFLE (FULL)	ngePnius rae dsgednei to be etimaslernd
DISEMVOWEL	Pngns r dsgnd to be strmlnd
INTRUDE	Pe‘nguins a{re d)esigned t;o b*e stre<amlined
KEYBOARD TYPO	Penguinz xre dwsigned ro ne streamllned
NATURAL NOISE	Penguijs ard design4d ti bd streamlinfd
TRUNCATE	Penguin are designe to be streamline
SEGMENTATION	Penguinsaredesignedtobestreamlined
PHONETIC	Pengwains’s ar dhiseind te be storimlignd

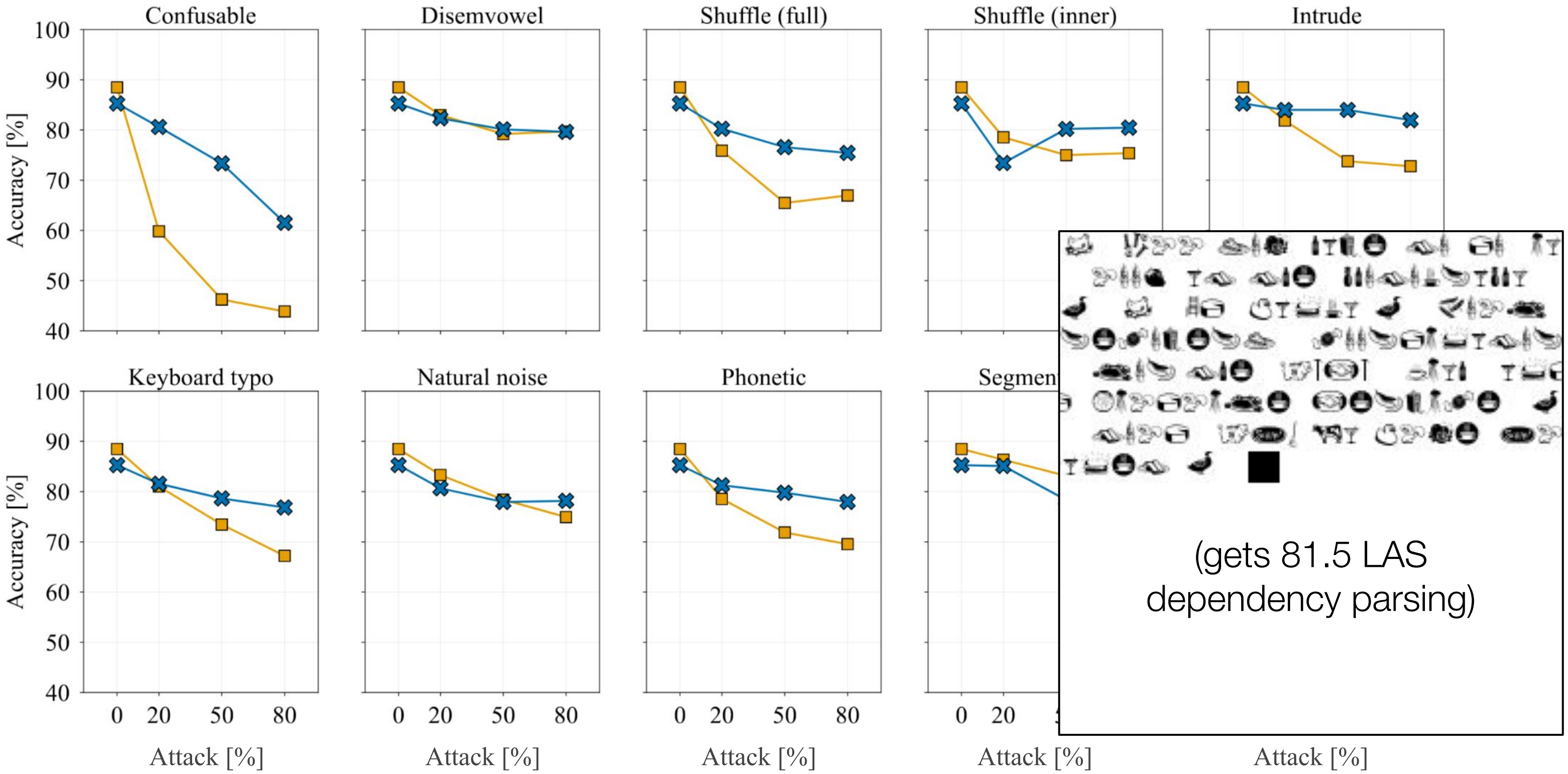
Evaluating against Ädʊɜrsarīał attacks

Saliency visualization in NLI tasks with % substitutions



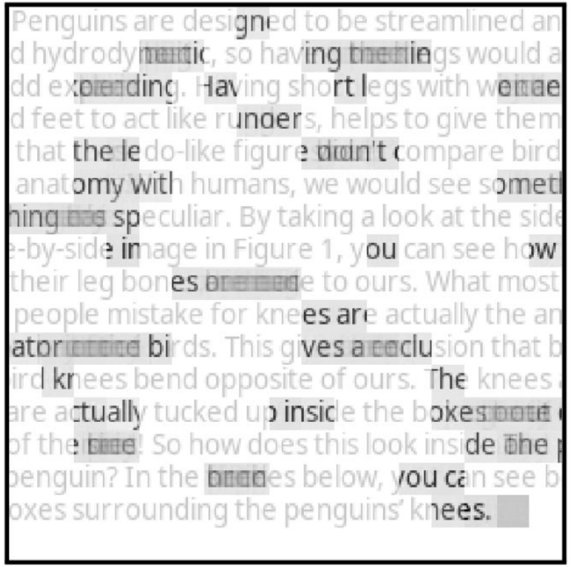
(a) 0%, *contradiction* (b) 80%, *contradiction* (c) 80%, *entailment*

Results on Zeroé (SNLI)

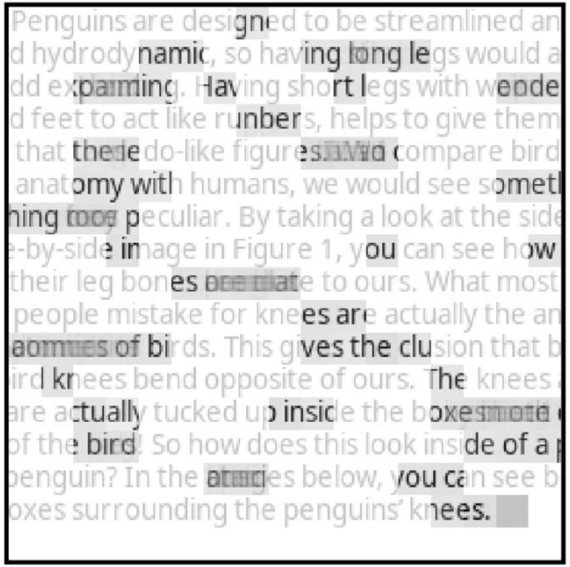


(gets 81.5 LAS dependency parsing)

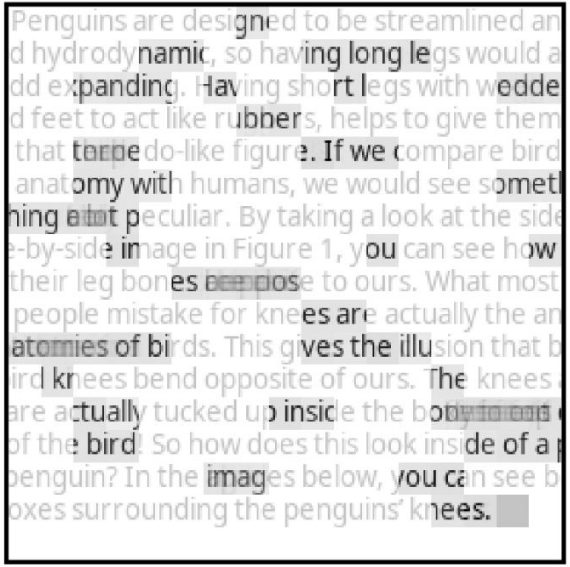
PIXEL dynamics across training



100K training steps



500K training steps



1M training steps

PIXEL structured rendering

Structured rendering

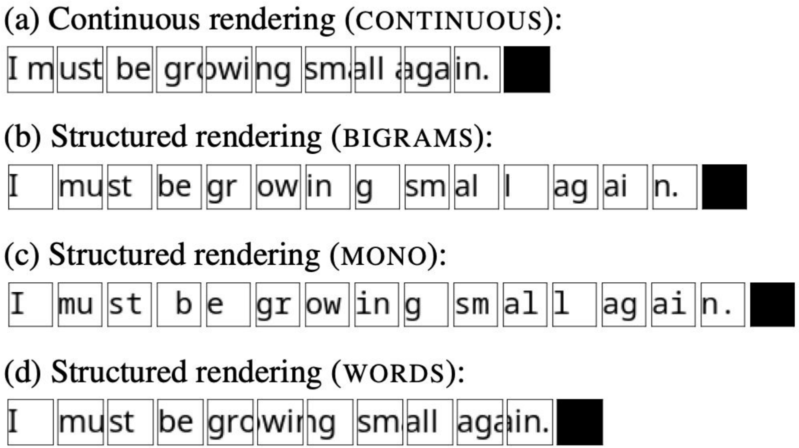


Figure 1: Examples of rendering strategies for the sentence "*I must be growing small again.*" from Carroll (1865). We use black patches to mark the end of a sequence, following Rust et al. (2023).

Structured rendering

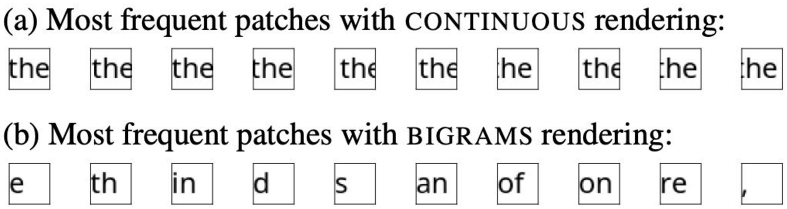
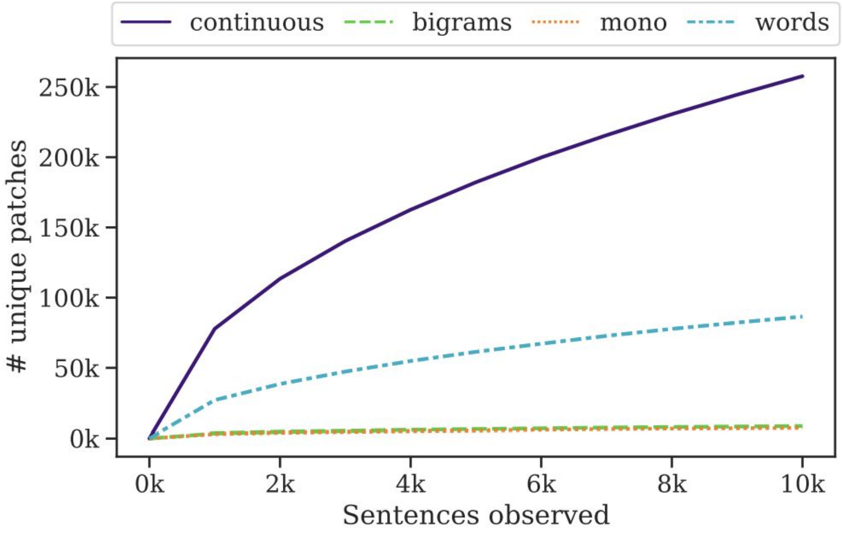


Figure 2: A continuous rendering strategy results in many uniquely-valued image patches for similar inputs, while structured rendering (here, BIGRAMS) regularises and compresses the potential input space.

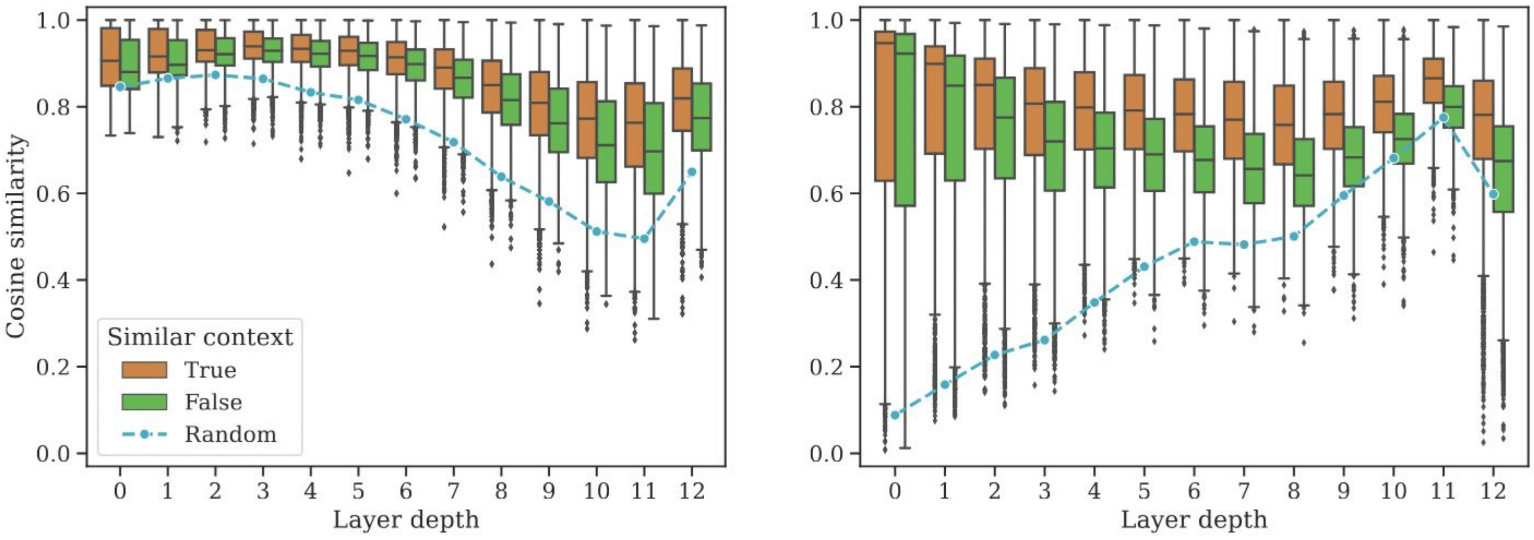


Structured rendering results

	Structure			Scale						
	UDP	GLUE			UDP		GLUE		TyDiQA-GoldP	
Renderer	*Avg.*	*Avg.*	*Variant*	$\|\theta\|$	*Avg.*	$\Delta\mu$	*Avg.*	$\Delta\mu$	*Avg.*	$\Delta\mu$
CONTINUOUS	76.2	71.0	TINY	5.5M	72.0	−0.3	66.5	+12.7	41.6	+4.9
BIGRAMS	76.1	75.4	SMALL	22M	76.1	−0.1	75.4	+4.4	50.8	+2.0
MONO	75.9	74.4	BASE	86M	75.5	−0.6	78.0	+3.9	52.8	+0.5
WORDS	76.6	74.7	BERT	110M	50.5	—	80.0	—	51.5	—

Table 2: **Structure** (left): averaged results for SMALL-models comparing downstream performance on UDP and GLUE following the different rendering strategies. **Scale** (right): averaged results across model scales using the BIGRAMS rendering structure. $\Delta\mu$ is the difference in average performance between BIGRAMS and CONTINUOUS rendering for a given model scale.

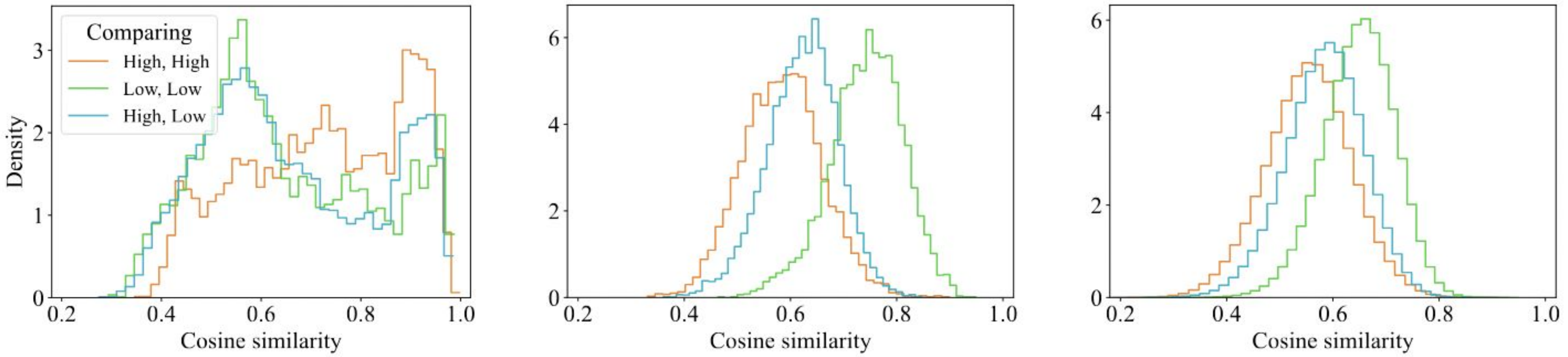
Distributions of cos similarities across layers



(a) BASE-BIGRAMS (b) BERT

Distributions of cosine similarities for verbs and nouns from the WiC dataset across model layers 0-12, layer 0 being the input layer. Every example presents a target word in either a similar or different context across a sentence pair. The representation of the target word is computed as the mean hidden state output over the corresponding tokens. We generally see that BASE-BIGRAMS encodes target words in a similar context as more similar. The median cosine similarity between random words from random sentences are shown as a baseline.

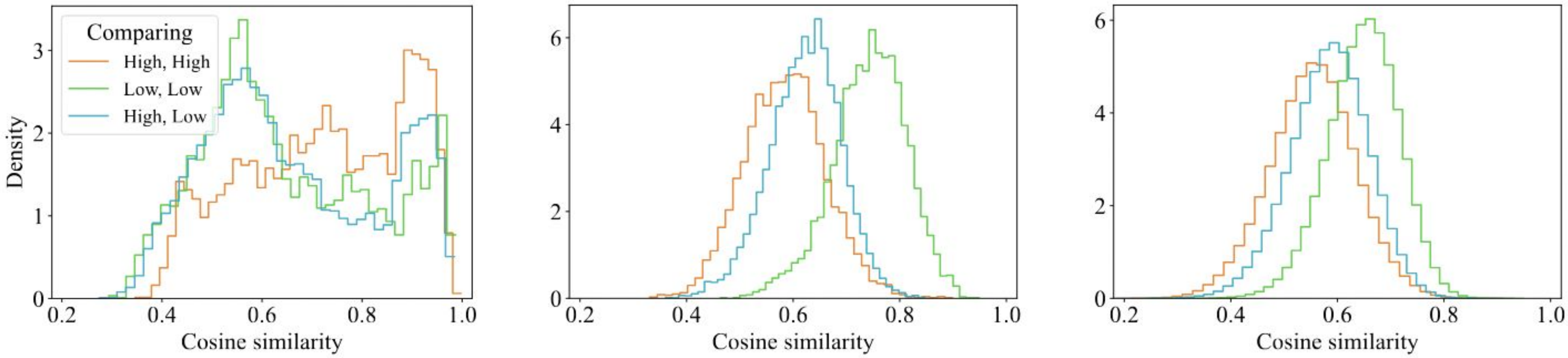
Words in Context



(a) Words in isolation, PIXEL (b) Words in isolation, BASE-BIGRAMS (c) Words in context, BASE-BIGRAMS

Distributions of cosine similarities for verbs and nouns from the WiC dataset across model layers 0-12, layer 0 being the input layer. Every example presents a target word in either a similar or different context across a sentence pair. The representation of the target word is computed as the mean hidden state output over the corresponding tokens. We generally see that BASE-BIGRAMS encodes target words in a similar context as more similar. The median cosine similarity between random words from random sentences are shown as a

Words in Context

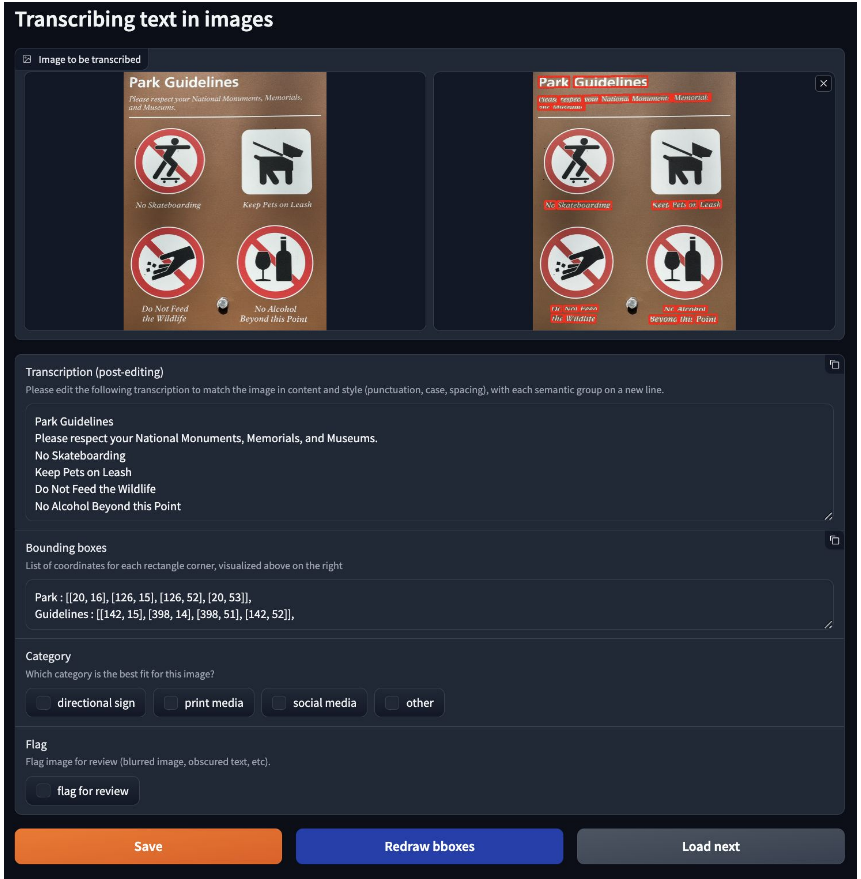


(a) Words in isolation, PIXEL (b) Words in isolation, BASE-BIGRAMS (c) Words in context, BASE-BIGRAMS

Distributions of cosine similarities within samples of high-frequency words (High), low-frequency words (Low), or between the two samples. Rendering with BIGRAMS structure leads to less directionally aligned vector representations of frequent words that have seen more updates during pretraining compared to infrequent words.

BENCHMARKING img-translation

OCR Annotation Interface



MT Annotation Interface

Task
John_Hopkinsuniversity_20240701_DE_test

Hit
509595993

SKIP HIT

Please provide translations as they would appear on corresponding traffic signs locally where possible

If corresponding sign exists in your location, but with significantly different text, staying true to the text in image takes priority

If sign does not exist in your location, then provide a translation which stays true to text in image while being an appropriate translation for a sign.

If for some reason it is impossible to provide a translation which works in local language, while staying true to source please raise a query and we will confirm with client.

Transcript directional sign	Translation de
CAUTION	
WET FLOOR	

Did the visual context influence the translation? *

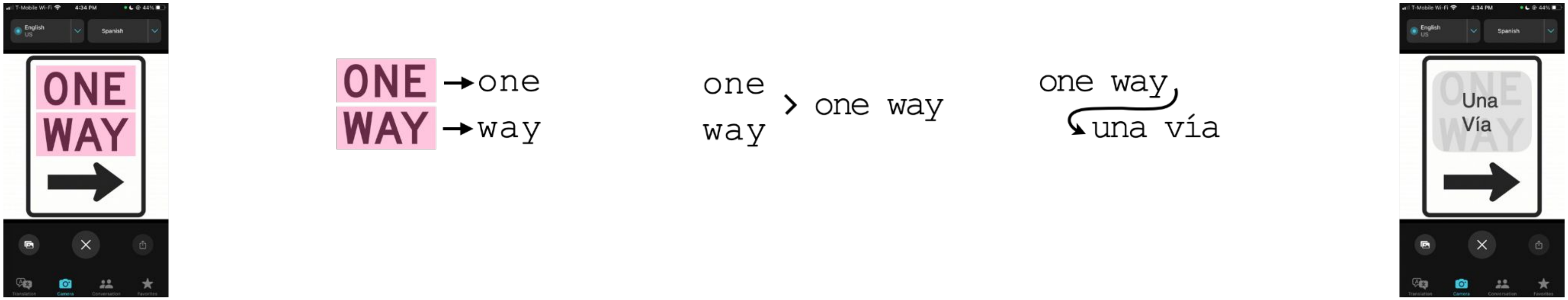
- Yes
- No

VST (OCR→MT) on Vistra

	mBART			Google Translate			GPT-4o		
OCR Model	**chrF**	**BLEU**	**COMET**	**chrF**	**BLEU**	**COMET**	**chrF**	**BLEU**	**COMET**
[Direct]							36.9	9.1	60.1
Tesseract-OCR	2.3	0.1	28.8	3.5	0.1	30.4			
Paddle-OCR	26.8	9.0	46.1	36.0	16.7	57.0			
GPT-OCR	28.1	6.9	48.0	36.4	13.2	58.2			
Google-OCR	31.1	9.1	47.3	37.4	14.9	55.3			
[Direct]									
Tesseract-OCR	2.4	0.1	30.1	3.6	0.3	31.7	54.0	21.4	73.4
Paddle-OCR	17.5	3.1	44.4	60.8	33.8	75.1			
GPT-OCR	23.3	4.2	50.4	60.8	24.6	75.0			
Google-OCR	22.0	4.0	45.5	62.2	29.9	71.3			
[Direct]									
Tesseract-OCR	1.7	0.1	25.3	2.6	0.1	27.3	35.6	10.7	70.2
Paddle-OCR	13.0	5.8	42.4	46.5	20.0	73.0			
GPT-OCR	16.0	7.5	42.4	48.1	18.4	71.0			
Google-OCR	14.8	5.1	43.1	47.1	15.1	74.4			
[Direct]									
Tesseract-OCR	0.3		32.6	0.4		34.4	33.6		85.5
Paddle-OCR	18.2		62.0	40.2		82.0			
GPT-OCR	19.7		63.1	40.1		82.5			
Google-OCR	18.7		59.2	41.6		77.7			

Translating text in natural images

Combining separate modeling stages



Detection Recognition Grouping Translation Overlay

Error propagation

WAY → way

way › way

way

forma

Detection

Recognition

Grouping

Translation

Overlay

Error propagation

Detection

ONE → on
WAY → way

Recognition

on
way
> on way

Grouping

on way
en camino

Translation

Overlay

Error propagation

Detection

Recognition

one way > one way

Grouping

one way

uno forma

Translation

Overlay

Error propagation

Agreement error

Contextual error

Correct translation:
UNA VÍA

Error propagation

Translating text in natural images

Text English | **Translation** German

EXIT

?

Ausfahrt

Ausgang

Image English | **Translation** German

Ausfahrt

Ausgang

Creating a benchmark dataset: Vistra

772 images containing English text, with metadata, transcripts, and translations to 4 target languages (German, Spanish, Russian, and Chinese)


```
{
  "image_file": "3c2b0778.png",
  "height": 1024, "width": 768,
  "category": "directional sign",
  "transcript": ["EXIT ONLY", "ONE WAY"],
  "translation":
      "de": ["NUR AUSFAHRT", "EINBAHNSTRAßE"],
      "es": ["SOLO SALIDA", "UNA VÍA"],
      "ru": ["ТОЛЬКО ВЫЕЗД", "ОДНОСТОРОННЕЕ ДВИЖЕНИЕ"],
      "zh": ["仅用作出口", "单向"],
  "bounding_boxes": {'EXIT': [[0.4701, 0.2565], …},
  "requires_image_context":
      "de":true, "es":true, "ru":false, "zh":true
}
```

Creating a benchmark dataset: Vistra

772 images containing English text, with metadata, transcripts, and translations to 4 target languages (German, Spanish, Russian, and Chinese)

Examples marked context-sensitive:

German: 99%
Spanish: 54%
Russian: 6%
Chinese: 96%


```
{
  "image_file": "3c2b0778.png",
  "height": 1024, "width": 768,
  "category": "directional sign",
  "transcript": ["EXIT ONLY", "ONE WAY"],
  "translation":
      "de": ["NUR AUSFAHRT", "EINBAHNSTRAßE"],
      "es": ["SOLO SALIDA", "UNA VÍA"],
      "ru": ["ТОЛЬКО ВЫЕЗД", "ОДНОСТОРОННЕЕ ДВИЖЕНИЕ"],
      "zh": ["仅用作出口", "单向"],
  "bounding_boxes": {'EXIT': [[0.4701, 0.2565], …},
  "requires_image_context":
      "de":true, "es":true, "ru":false, "zh":true
}
```

Models evaluated

Model	OCR	MT	VST	Release	OCR level	Returns bboxes?	Multilingual
PaddleOCR	✓			OPEN-SOURCE	line|word	yes	
TesseractOCR	✓			OPEN-SOURCE	word	yes	
Google Cloud Vision	✓			COMMERCIAL	word	yes	
mBART		✓		OPEN-SOURCE	—	—	✓
Google Translate		✓		COMMERCIAL	—	—	
GPT-4o	✓	✓	✓	COMMERCIAL	unknown	no	✓

OCR Error Taxonomy

text detection (I-III) errors

recognition (IV-VIII) errors

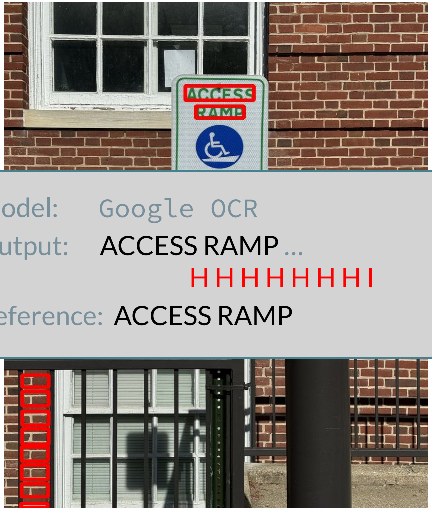
Class	Description
I	Undetected text: missing text and bounding boxes
II	Text hallucination: text detected where no text present
III	Bounding box misplaced: text clipped, cropping would affect recognition
IV	Grouping error: text from different groups intermixed in output text
V	Punctuation error
VI	Spacing error
VII	Character-level substitution
VIII	Word-level substitution

Examples of Errors by Class

I: Undetected text

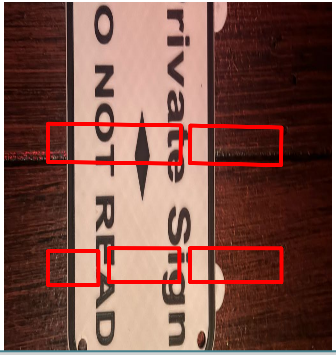
Model: `Google OCR`
Output: ESPASSING
Reference: NO TRESPASSING STATE HIGHWAY ADMINISTRATION

II: Text hallucination



Model: `Google OCR`
Output: ACCESS RAMP ...
HHHHHHHHI
Reference: ACCESS RAMP

III: Bounding box error



Model: `Paddle-OCR`
Output: Private Sign DONOTREAD
Reference: Private Sign DO NOT READ

Examples of Errors by Class

IV: Grouping error

Model: `Paddle OCR`

Output: ... I'M THINKING OF HAVE YOU GOT ANY DRAWING A NEW GOOD IDEAS? COMIC STRIP

Reference: ... I'M THINKING OF DRAWING A NEW COMIC STRIP HAVE YOU GOT ANY GOOD IDEAS?

V: Punctuation error

Model: `Tesseract-OCR`

Output:
| (NO OUTSIDE)!
|; FOOD OR |!
|| DRINKS |
] | ALLOWED |,

Reference: NO OUTSIDE FOOD OR DRINKS ALLOWED

VI: Spacing error

Model: `Paddle-OCR`

Output: PULLTOOPEN|PUSHTOCLOSE

Reference: PULL TO OPEN |
PUSH TO CLOSE

Examples of Errors by Class

VII: Character-level substitution

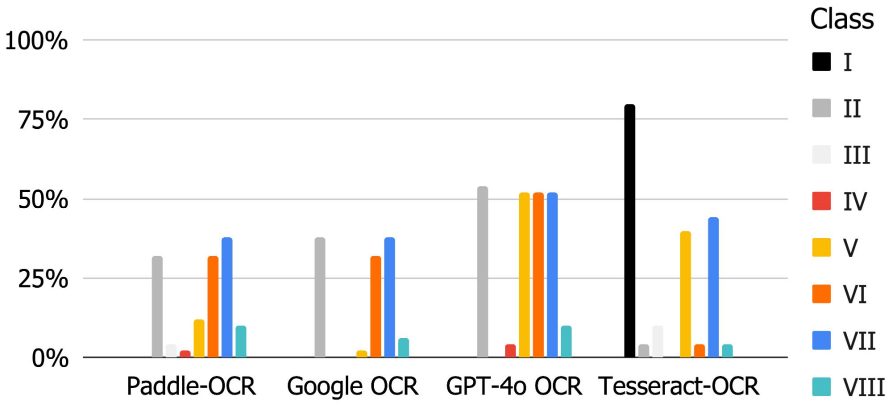
Model: `Google OCR`
Output: NO QVERNIGHT PARKING
Reference: NO OVERNIGHT PARKING

VIII: Word-level substitution

Model: `Google OCR`
Output: TOWN OF FEAST LYME ...
Reference: TOWN OF EAST LYME ...

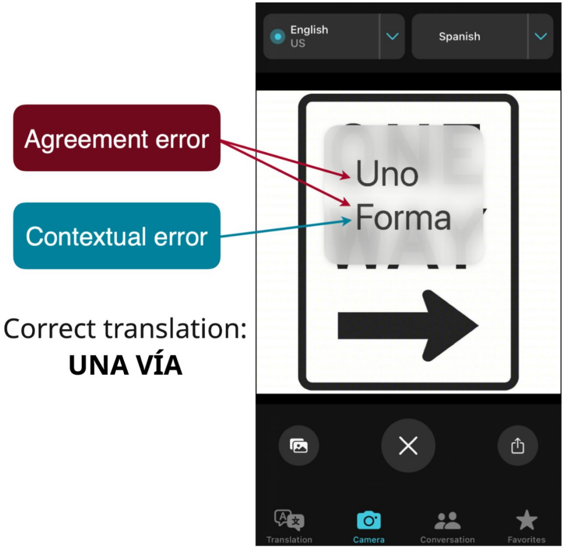
OCR Results on Vistra

Model	CER↓	TER↓	Sub.	Del.	Ins.
Paddle-OCR	13.0	21.5	963	2824	2851
Google OCR	18.0	32.0	186	381	8496
GPT-4o	23.8	36.0	1132	1277	9728
Tesseract-OCR	124.0	134.3	9597	37081	16477



Motivational OCR → MT Error Examples

Correct translation:
UNA VÍA



A **grouping error** causes each word to be translated individually, resulting in agreement errors `(Apple Translate)`

`mBART`

MT output:

| | | | | | | | | | | | | | | |

`Tesseract-OCR`

OCR output:

| (NO OUTSIDE)!
|; FOOD OR |!
| | DRINKS |
]| ALLOWED |,

Reference translation:

NO SE PERMITEN ALIMENTOS Y BEBIDAS AJENOS A ESTE ESTABLECIMIENTO

Inserted punctuation breaks up the text sequence, resulting in translation errors despite correctly recognized text `(mBART)` open-source

Cascaded visually-situated translation (OCR→MT) on Vistra

		mBART		
Target Language	**OCR Model**	**chrF**	**BLEU**	**COMET**
German	Tesseract-OCR	2.3	0.1	28.8
	Paddle-OCR	26.8	9.0	46.1
Spanish	Tesseract-OCR	2.4	0.1	30.1
	Paddle-OCR	17.5	3.1	44.4
Russian	Tesseract-OCR	1.7	0.1	25.3
	Paddle-OCR	13.0	5.8	42.4
Chinese	Tesseract-OCR	0.3	—	32.6
	Paddle-OCR	18.2	—	62.0

commercial

Cascaded visually-situated translation (OCR→MT) on Vistra

		Google Translate		
Target Language	**OCR Model**	**chrF**	**BLEU**	**COMET**
German	GPT-OCR	36.4	13.2	58.2
	Google Cloud	37.4	14.9	55.3
Spanish	GPT-OCR	60.8	24.6	75.0
	Google Cloud	62.2	29.9	71.3
Russian	GPT-OCR	48.1	18.4	71.0
	Google Cloud	47.1	15.1	74.4
Chinese	GPT-OCR	40.1	—	82.5
	Google Cloud	41.6	—	77.7

commercial

Direct visually-situated translation with a multimodal model

		Google Translate			GPT-4o		
Target Language	**OCR Model**	**chrF**	**BLEU**	**COMET**	**chrF**	**BLEU**	**COMET**
German	GPT-OCR	36.4	13.2	58.2	36.9	9.1	60.1
	Google Cloud	37.4	14.9	55.3			
Spanish	GPT-OCR	60.8	24.6	75.0	54.0	21.4	73.4
	Google Cloud	62.2	29.9	71.3			
Russian	GPT-OCR	48.1	18.4	71.0	35.6	10.7	70.2
	Google Cloud	47.1	15.1	74.4			
Chinese	GPT-OCR	40.1	—	82.5	33.6	—	85.5
	Google Cloud	41.6	—	77.7			

Can Multimodal LLMs resolve contextual ambiguity?

References:

English transcript:

EXIT ONLY
ONE WAY

German translation:

Nur Ausfahrt
Einbahnstraße

GPT-4o Cascade:

English OCR:

****EXIT ONLY** **ONE WAY** →**

German translation:

NUR AUSGANG EINWEG →

GPT-4o Direct:

German translation:

****AUSFAHRT NUR****
****EINEN WEG****

Can Multimodal LLMs resolve contextual ambiguity?

14/14 examples of "EXIT" are translated as "AUSGANG" in a cascade

GPT-4o Cascade:

English OCR:

****EXIT ONLY** **ONE WAY** →**

German translation:

NUR AUSGANG EINWEG →

4 examples of "EXIT" are translated as "AUSFAHRT" with a multimodal model

GPT-4o Direct:

German translation:

****AUSFAHRT NUR****
****EINEN WEG****

*and 5 as AUSGANG,
and 6 are fully incorrect

Cautionary note on evaluation metrics

Translation
German

`The exit is over there`

`Die Ausfahrt ist dort drüben`

`Der Ausgang ist dort drüben`

Cautionary note on evaluation metrics

With COMET, all combinations of these as hyp and ref score exactly the same!

`Die Ausfahrt ist dort drüben` → `Die Ausfahrt ist dort drüben`

`Der Ausgang ist dort drüben` → `Der Ausgang ist dort drüben`

Lexical metrics may (for now) better check use of context-sensitive terms