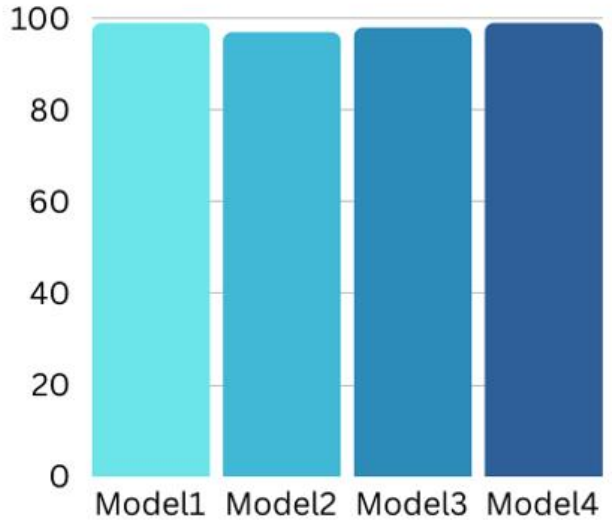
From Language Distance to Subwords: Navigating Variation in Multilingual NLP

Rob van der Goot

Identifying Open Challenges in Language Identification (ACL 2025)

- Task: Map text into language (ISO 639-3) codes

Wer jammerje jo no oer?	fry
Skal vi snakke lidt?	dan

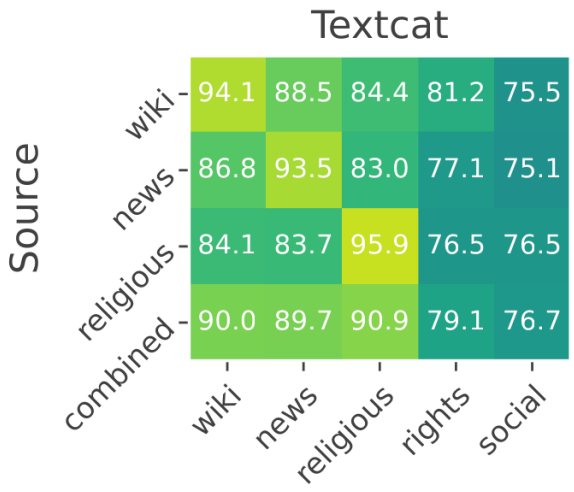


Collect

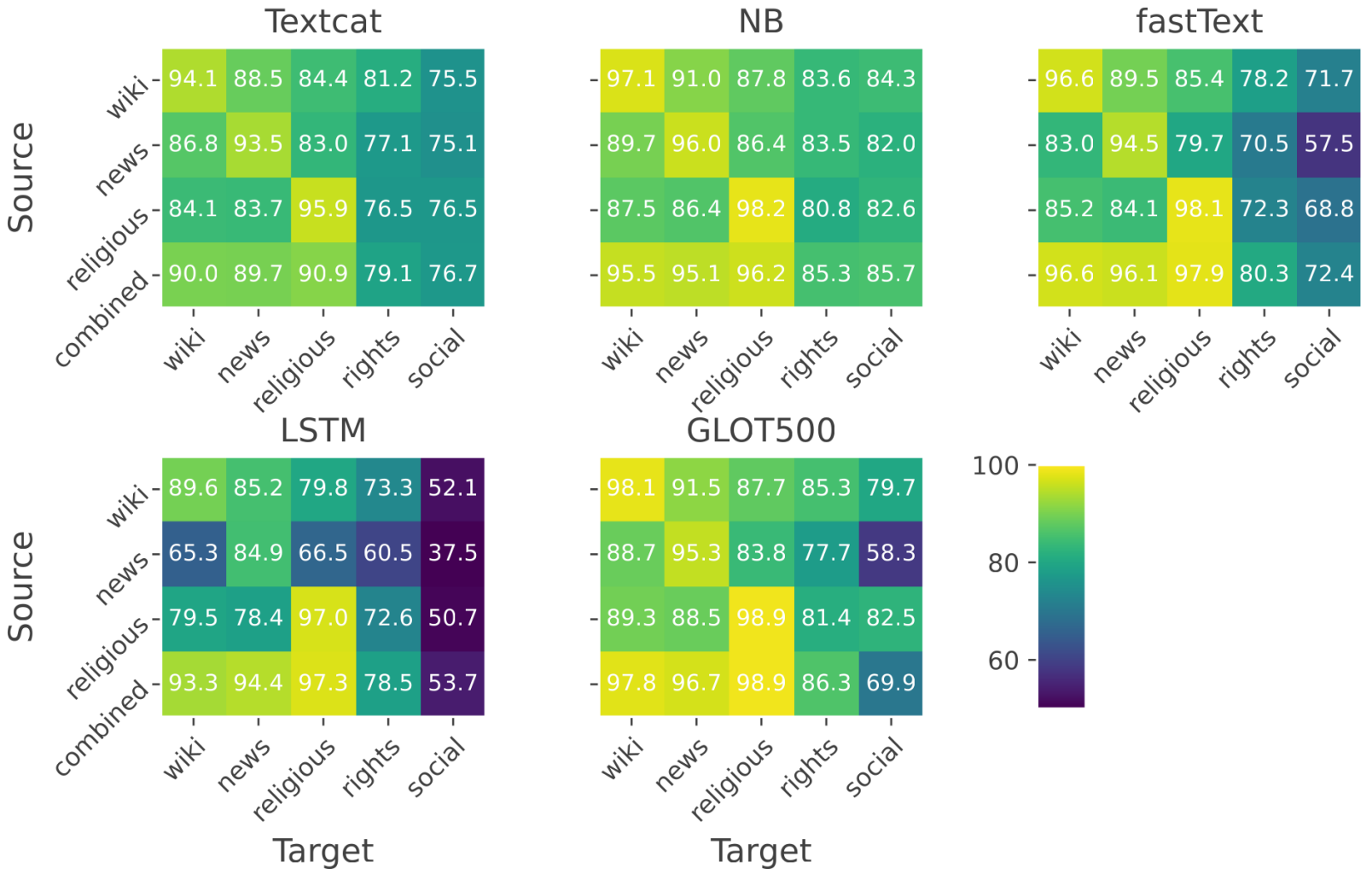
Dataset	langs	scripts	fams	domains
MIL-TALE	2,110	47	139	wiki, political, religious, grammar
UDHR	397	38	61	rights
OpenLID	139	25	16	literature, news, wiki, social, grammar, subtitles, spoken
MassiveSumm	77	24	13	news
TwitUser	59	20	13	social
UD	54	11	17	medical, news, academic, wiki, legal, nonfiction, learner-essays, fiction, social, grammar-examples, reviews, religious, spoken
Total	2,176/ 7,850	51/ 163	145/ 298	

Type	model	size	
N-gram overlaps	Textcat	40,000	
ML	NB	100,000	
Embeddings	FastText	4,434,860	
NN	LSTM	15,158,772	
Transformer LM	GLOT500	395,687,155	

Cross-domain evaluation



Cross-domain evaluation



Other findings

- 10 characters is too small, 100 is enough
- Need 100 samples per language
- Not so important: number of languages, scripts, language families
- Release best models with 2,034 language coverage

DistaLs: A Comprehensive Collection of Language Distance Measures

Rob van der Goot[1], Esther Ploeger[2], Verena Blaschke[3], Tanja Samardžić[4]
[1]IT University of Copenhagen, robv@itu.dk
[2]Aalborg University, espl@cs.aau.dk
[3]LMU Munich & Munich Center for Machine Learning, blaschke@cis.lmu.de
[4]University of Zurich, tanja.samardzic@uzh.ch

Category	Feature	Source	Coverage
Metadata	wiki_size	Wikipedia	7,856
	nlp_state	State and fate	2,269
	speakers	LinguaMeta	5,539
	AES	Glottolog	7,725
	loc	Glottolog	7,629
Typology	lang2vec	URIEL	3,910
	lang2vec_knn	URIEL	3,910
	PHOIBLE	PHOIBLE	2,078
	grambank_all	Grambank	2,326
	grambank_.*	Grambank	2,326
	glot_tree	Glottolog	7,856
	scripts	LinguaMeta, GlotScript	6,431
Wordlists	**ASJP**	ASJP	6,117
	concepts	Conceptualizer	1,274
Text-driven	whitespace	LTI LangID	2,525
	punctuation	LTI LangID	2,525
	char_distr.	LTI LangID	2,525
	textcat	LTI LangID	2,525

DistaLs

- https://distals.streamlit.app/
- pip package
- python interface

3CPO

- Diversity-first language model
- GLOT500/mmBERT: last phase is multi-lingual
- We propose to prioritize diversity over data availability

3CPO

- Full DistaLs matrix (7856, 7856, 10) features
- 64 clusters
- Sample heuristics:
  - Equal
  - Largest: top-100 + smoothing
  - OneCluster: largest lang. Of each cluster
  - Cluster: equal amount of each cluster
  - (68,719,476,736 characters)

3CPO

Strategy	Family		AES		NLP State		Speakers	
	entr.↑	cov.↑	entr.↑	cov.↑	entr.↑	cov.↑	entr.↑	cov.↑
`equal`	2.67	**1.00**	1.26	**1.00**	1.49	**1.00**	1.40	**1.00**
`largest`	1.79	0.12	0.86	0.71	1.78	**1.00**	0.42	0.70
`oneCluster`	2.12	0.13	1.16	0.71	**1.82**	**1.00**	0.96	0.80
`cluster`	**2.77**	**1.00**	**1.37**	**1.00**	1.43	**1.00**	**1.49**	**1.00**

Table 1: Entropy, coverage for each feature

3CPO

- Gemma3-1B
- Tokenizers from scratch (or not)
- 8 models

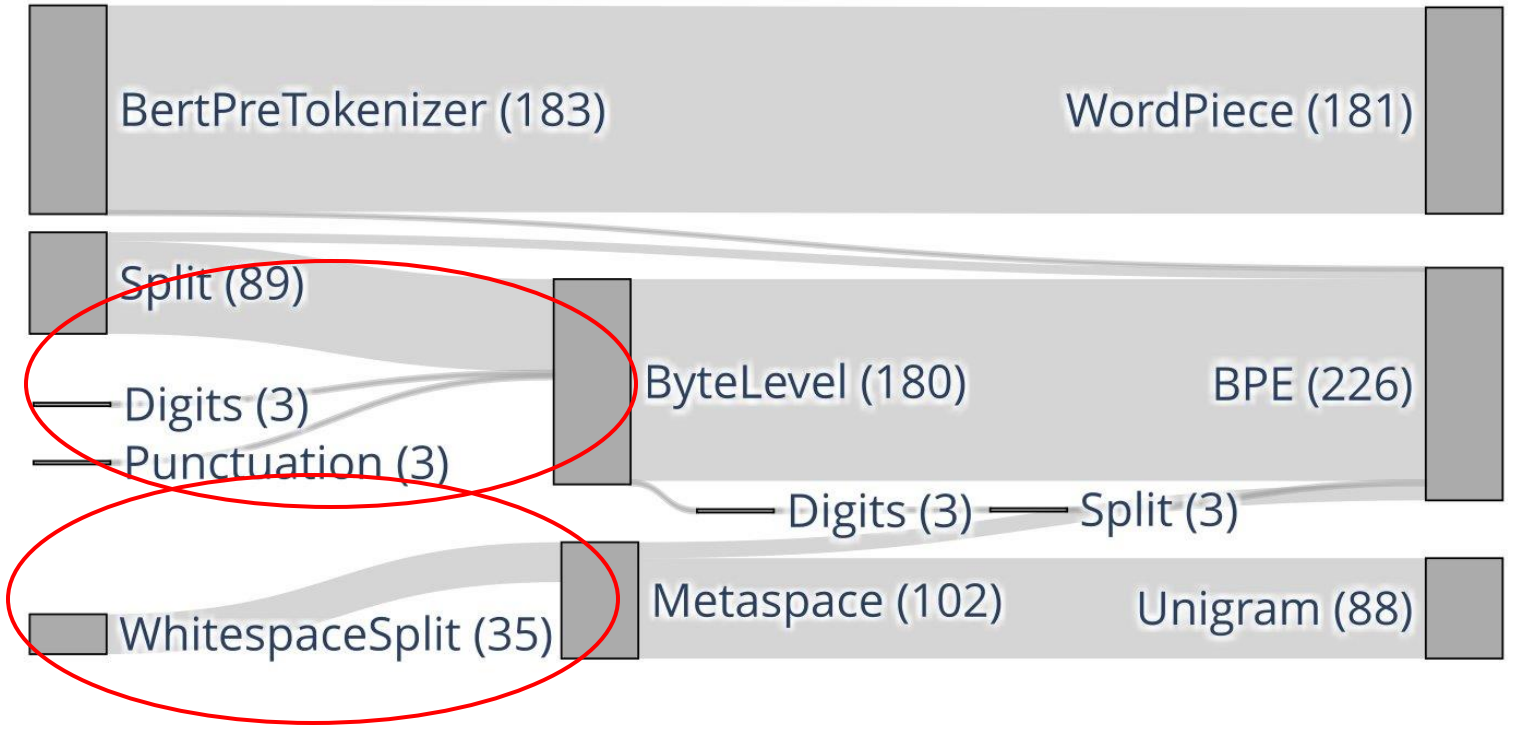
	gemma-3-1b-pt	largest	equal	oneCluster	cluster
Avg.	2.52	2.68	3.04	2.73	2.99

3CPO


lm loss
pretrain-run-equal
pretrain-run-cluster
12
10
8
6
4
Step
200
400
600
800
1k

From Bytes to Subwords: Challenges of Input Representations in NLP (ACL-findings 2026)

- Top-500 most downloaded language models (Huggingface)



From Bytes to Subwords: Challenges of Input Representations in NLP

- Regular expressions:

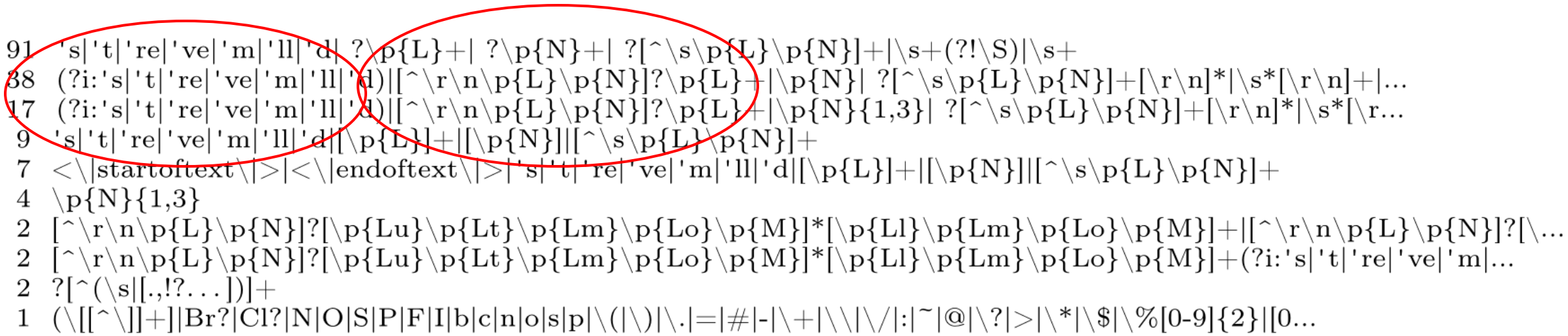


Figure 5: Frequency counts of regular expressions used to split tokens (first 100 characters shown

Diversity in subword segmenters

- There is not much diversity, arbitrary decisions propagate
- Internal diversity also limited

Normalization is lossy

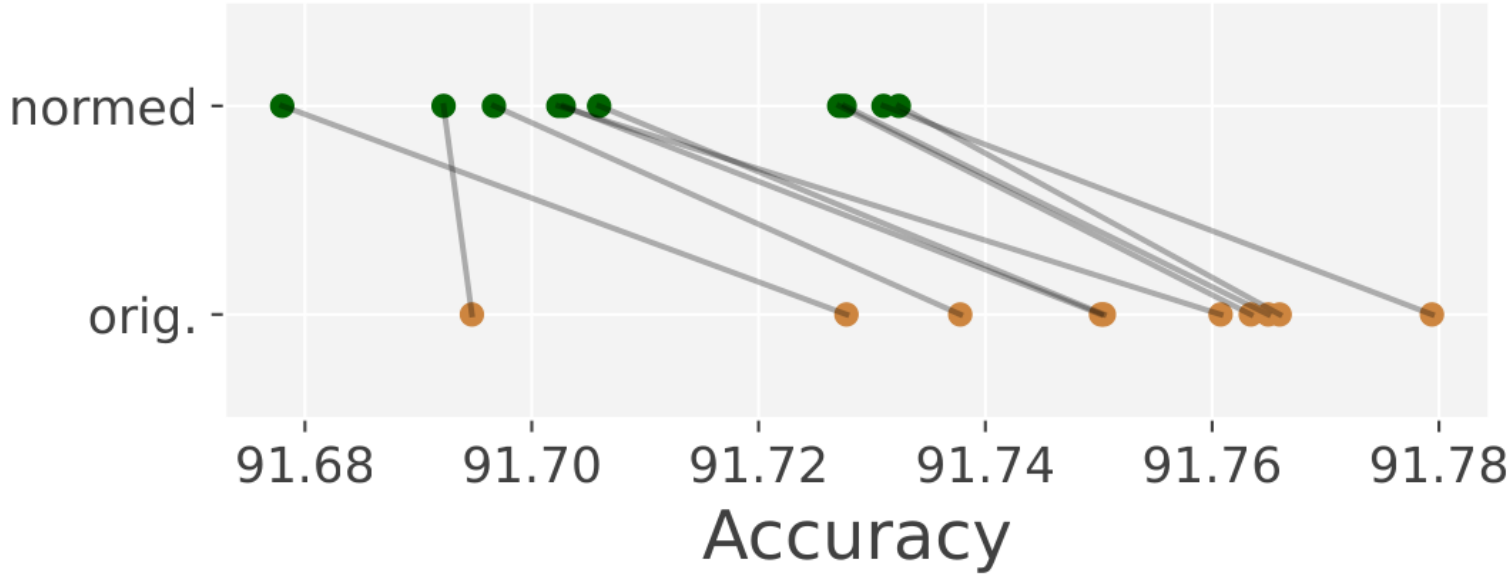
Norm	Count
NFC	264
NFD	29
NFKC	5
NFKD	1

- Previous work has shown that normalization can be beneficial
- Is it always?
- Decode-encode test fails: (decode(encode(text))) == text

(Kudo and Richardson, 2018; Dagan et al., 2024; Jang et al., 2025)

Normalization is lossy

- Naïve Bayes, mixed-domain, 2,307 languages, 1,000 instances per language



UTF-8 is neither Fair nor Efficient

Characters	b		ø			r		n			
Bytes (UTF-8)	62		C3	B8		72		6E			
Characters	д			i			т			и	
Bytes (UTF-8)	D0	B4		D1	96		D1	82		D0	B8
Characters	孩				子				们		
Bytes (UTF-8)	E5	AD	A9		E5	AD	90		E4	BB	AC

Desiderata

UTF 8

- Coverage
- Backwards compatibility

LM, at least:

- Coverage
- Efficiency
- Fairness (across scripts/languages)
- Relevant overlap for multi-byte characters

Redesigning UTF-8

A bit more previous work than I initially found:

- Back to Bytes: Revisiting Tokenization Through UTF-8
- Bit-level BPE: Below the byte boundary
- BPE Stays on SCRIPT: Structured Encoding for Robust Multilingual Pretokenization

Proof of concepts

- Efficiency:
  - 10,000 characters for the 1,742 language-script combinations of Fineweb2
  - Save 10% bytes by simple remapping
- Fairness:
  - Script/category indices followed by character indices
  - Overlap across character indices?
  - Script-start/end "characters"?
- Unknown characters
  - Our sample only contains 8,739/159,866 Unicode characters
  - Bring back unknown token(s)?

Takeaways

- Subword segmentation strategies are not diverse
- Normalization should be used with care
- UTF-8 is suboptimal for language modeling

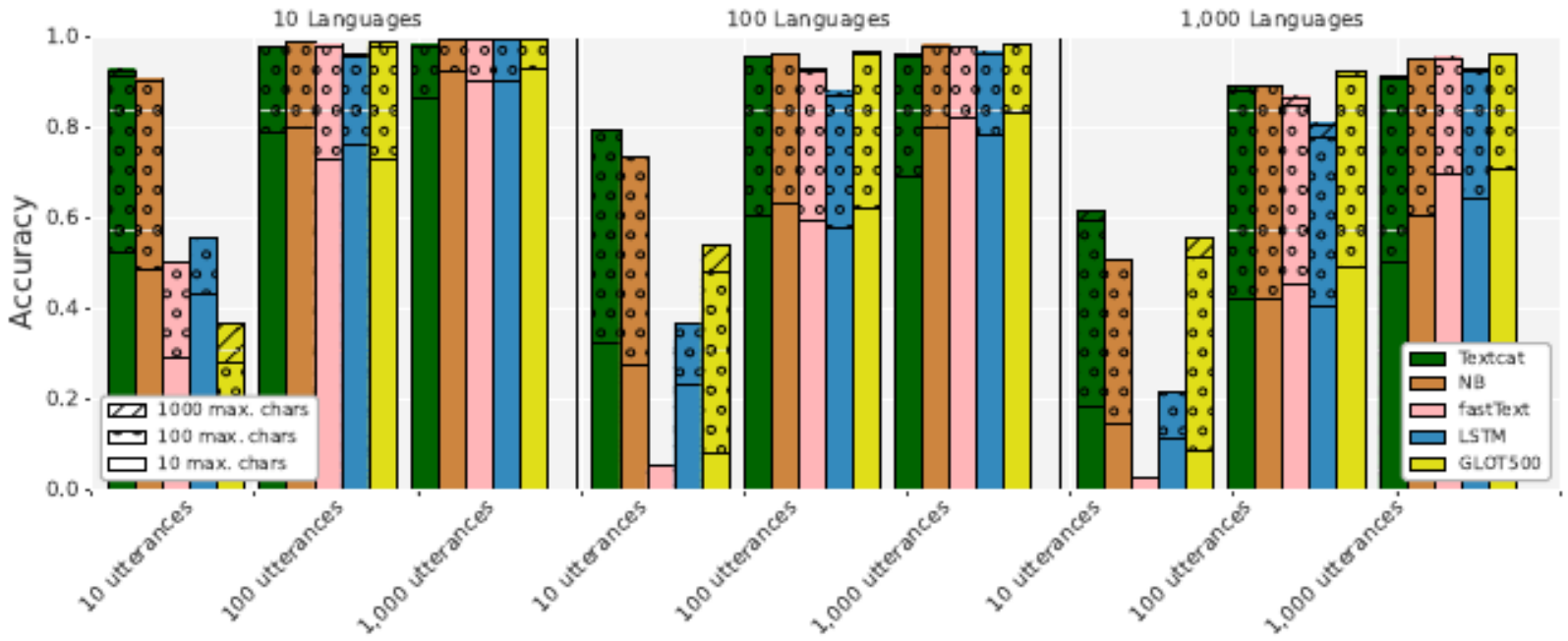
Thousand thanks!

IT UNIVERSITY OF COPENHAGEN

CARLSBERG FONDET

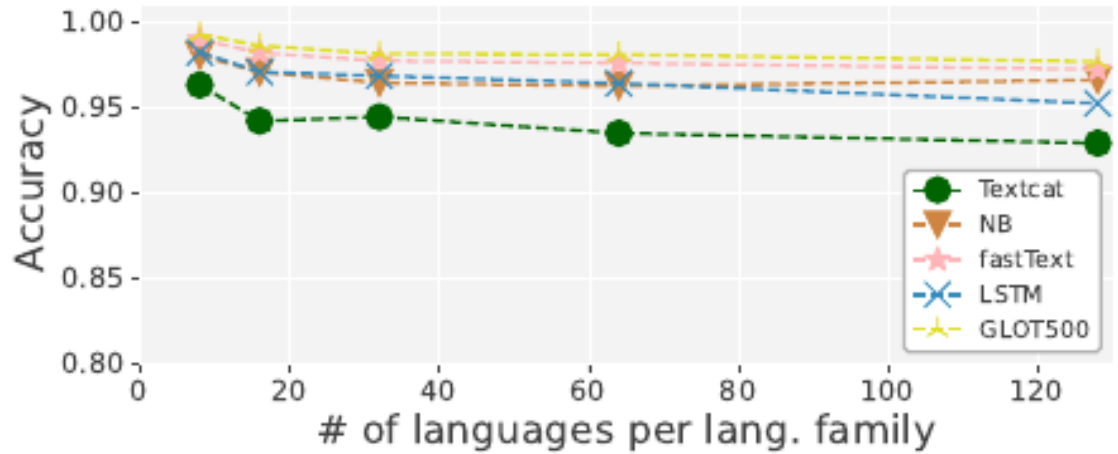
Size

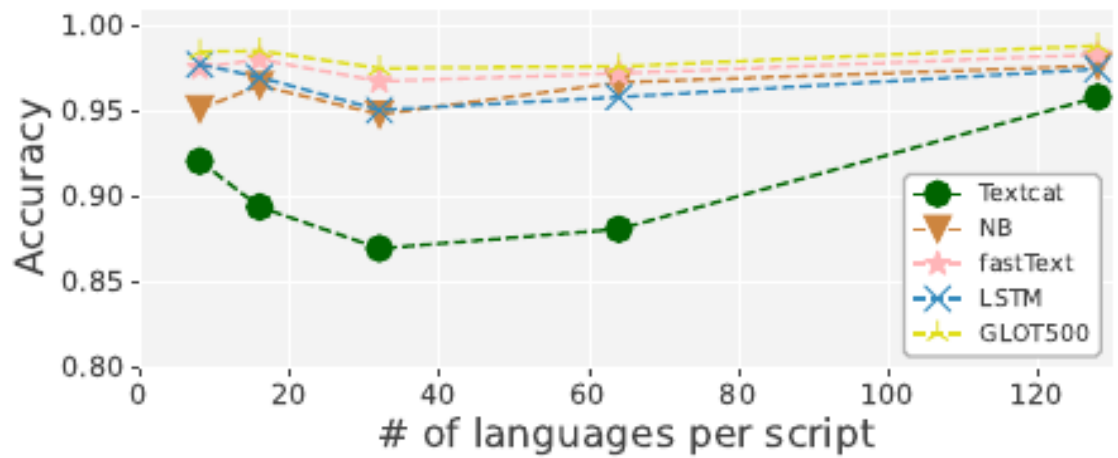
- Number of languages



Family/script

- Make subsets with N languages per family/script





Model	# params
Textcat	40,000
NB	100,000
fastText	4,434,860
LSTM	15,158,772
GLOT500	395,687,155

Table 3: Number of learned weights (# params) per model when training on 100 characters, 1,000 instances, and 100 languages.

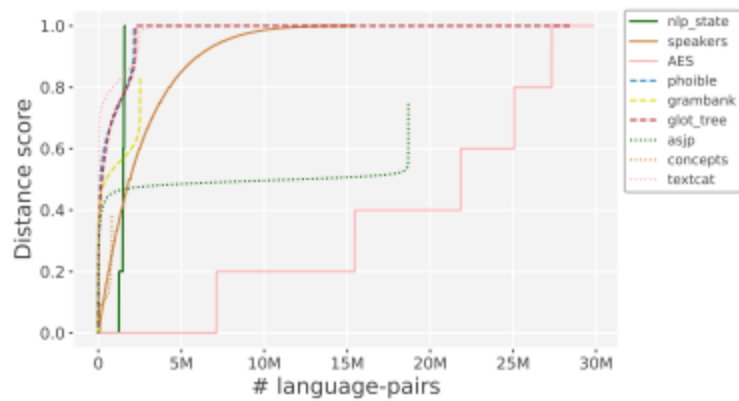


Figure 5: The cumulative probabilities of selected features.

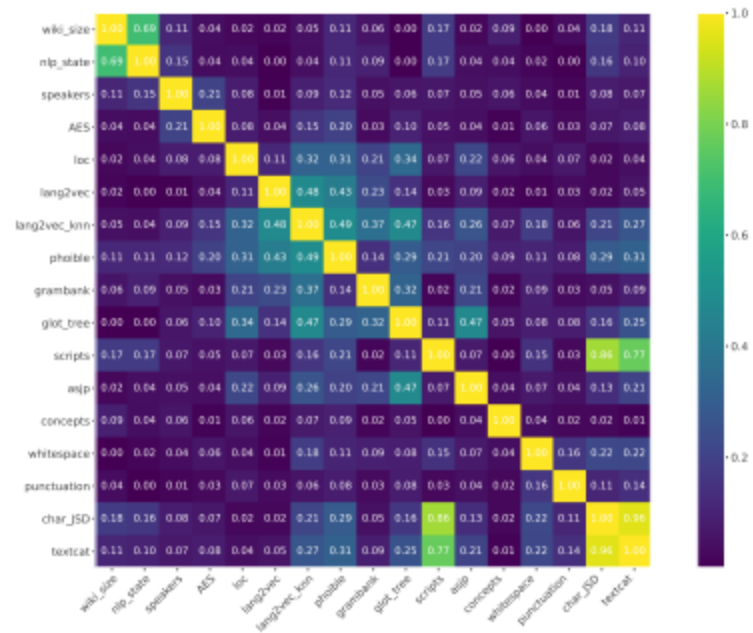


Figure 6: Pearson correlations across all features (except the grambank sub-categories). P-values in Appendix D.

What are characters?

- In NLP: Unicode/UTF-8
  - 159,866 characters

What is Unicode?

- A database of characters
  - "High" coverage
  - Backwards compatible (ASCII)

What is UTF-8?

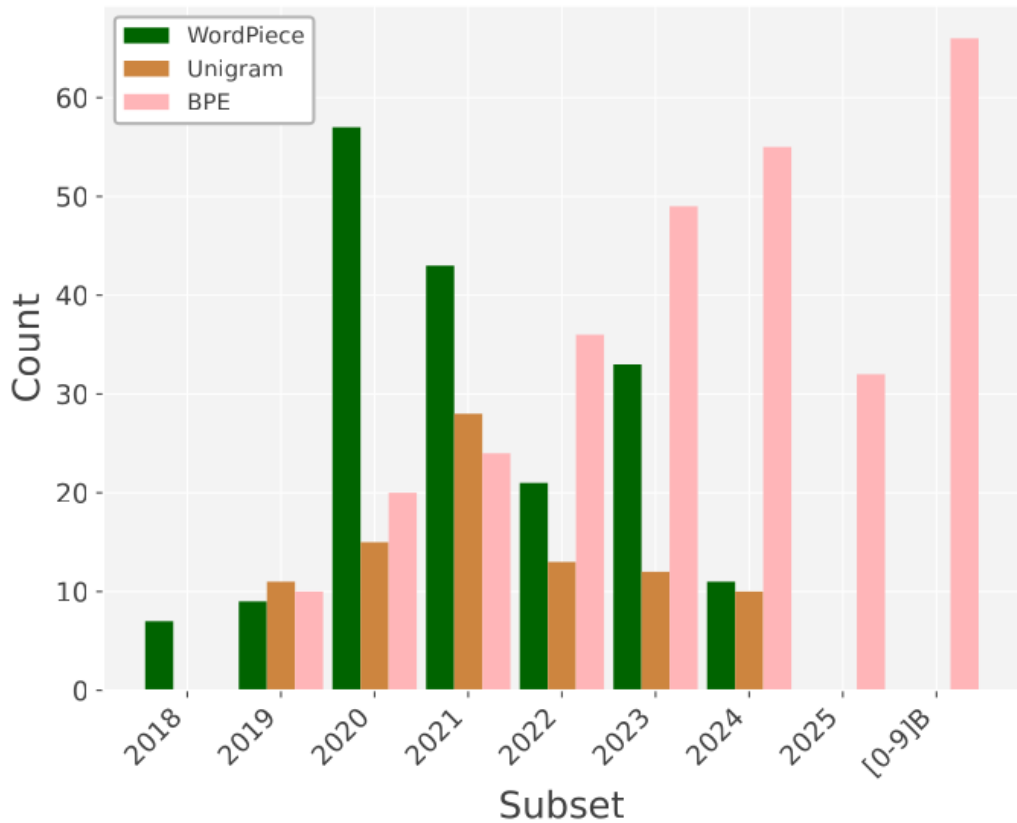
- Encoding of Unicode characters to bytes
  - Used on 98.3% of the websites indexed by W3C
  - Uses variable byte length, ASCII is represented in 1 byte, but many characters use 2-4 bytes
  - Consists of sequences of blocks that contains 1-n characters

Byte indexing and continuation bytes

00000001	1	0-128:	ASCII
00000010	2	128-256:	Continuation/unused
00000011	3		
…			
11111111	256		

ASCII

```
rob@rob-itu:~$ ascii -d
   0 NUL    16 DLE    32      48 0    64 @    80 P    96 `   112 p
   1 SOH    17 DC1    33 !    49 1    65 A    81 Q    97 a   113 q
   2 STX    18 DC2    34 "    50 2    66 B    82 R    98 b   114 r
   3 ETX    19 DC3    35 #    51 3    67 C    83 S    99 c   115 s
   4 EOT    20 DC4    36 $    52 4    68 D    84 T   100 d   116 t
   5 ENQ    21 NAK    37 %    53 5    69 E    85 U   101 e   117 u
   6 ACK    22 SYN    38 &    54 6    70 F    86 V   102 f   118 v
   7 BEL    23 ETB    39 '    55 7    71 G    87 W   103 g   119 w
   8 BS     24 CAN    40 (    56 8    72 H    88 X   104 h   120 x
   9 HT     25 EM     41 )    57 9    73 I    89 Y   105 i   121 y
  10 LF     26 SUB    42 *    58 :    74 J    90 Z   106 j   122 z
  11 VT     27 ESC    43 +    59 ;    75 K    91 [   107 k   123 {
  12 FF     28 FS     44 ,    60 <    76 L    92 \   108 l   124 |
  13 CR     29 GS     45 -    61 =    77 M    93 ]   109 m   125 }
  14 SO     30 RS     46 .    62 >    78 N    94 ^   110 n   126 ~
  15 SI     31 US     47 /    63 ?    79 O    95 _   111 o   127 DEL
rob@rob-itu:~$
```


WordPiece
Unigram
BPE
Count
0
10
20
30
40
50
60
2018
2019
2020
2021
2022
2023
2024
2025
[0-9]B
Subset