



LREC 2022

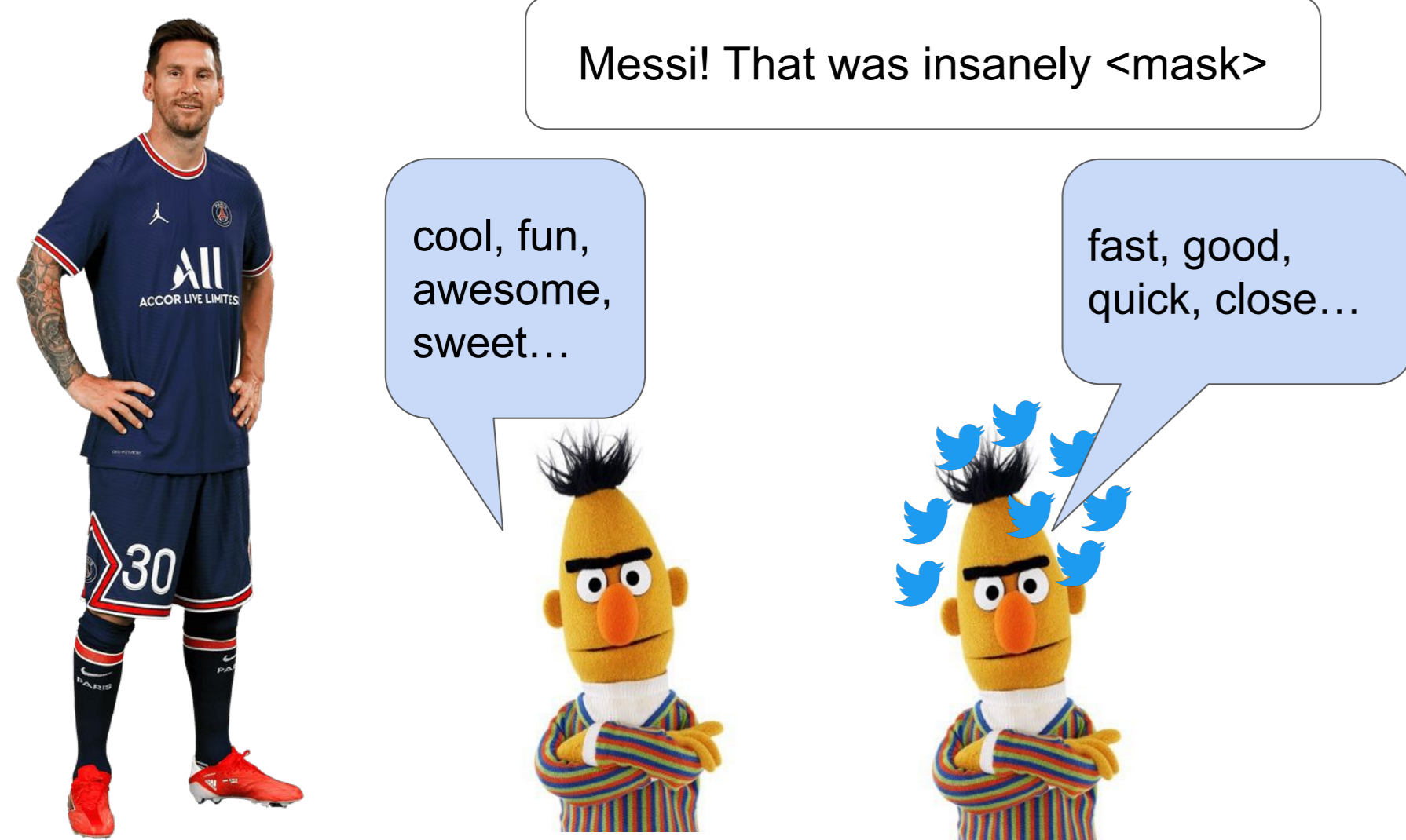
XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond

Francesco Barbieri♣, Luis Espinosa-Anke◇, Jose Camacho-Collados◇
♣Snap Inc., ◇Cardiff NLP, Cardiff University

Motivation & Background

Specializing Language Models

- augmenting with external information
- pretraining on domain-specific corpora

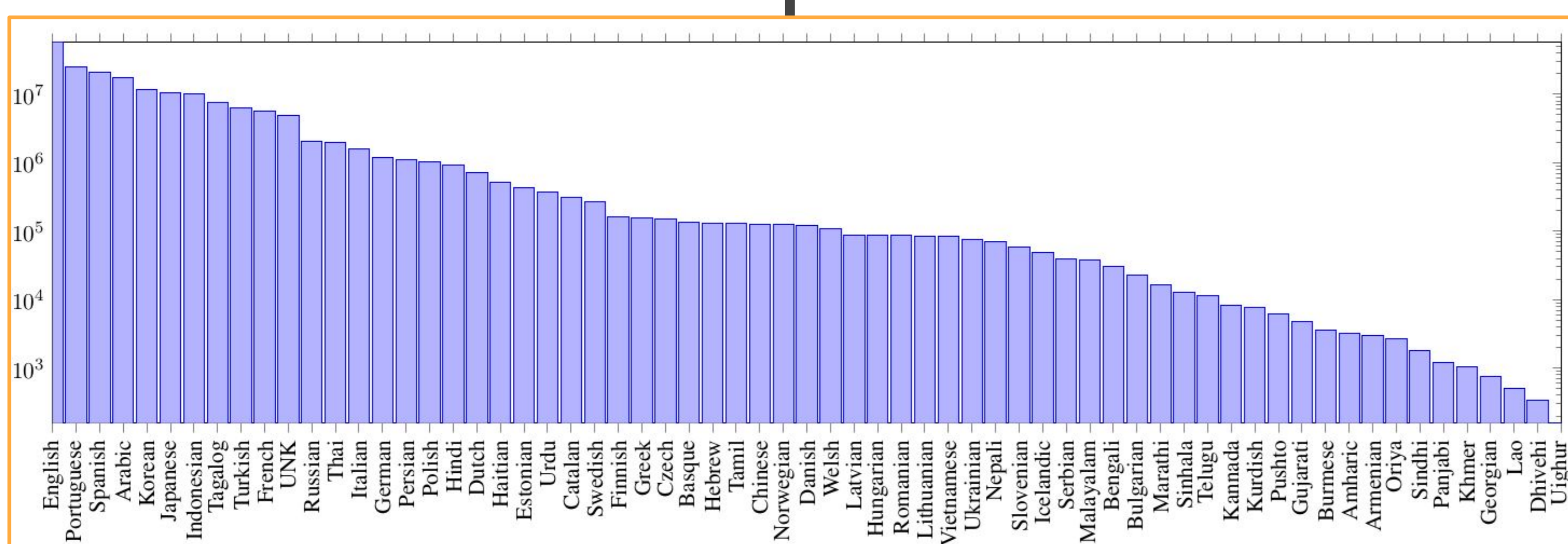


LMs for Twitter and English

- BerTweet (Nguyen et al., 2020), model and TweetEval (Barbieri et al., 2020), model and unified benchmark for tweet classification.
- However, not a similar set of models and frameworks for multilingual tweet classification

XLM-T: the Model

- **198M tweets** between May 2018 and March 2020
 - At least three tokens
 - No URLs
 - No language filtering
- **Continue pretraining** an XLM-R model
 - Start from a publicly available checkpoint
 - same MLM objective
 - pretrain until convergence in a validation set.
- **About 14 days on 8 NVIDIA V100 GPUs**



UMSAB: The Dataset

Lang.	Dataset	Time-Train	Time-Test
Arabic	SemEval-17 (Rosenthal et al., 2017)	09/16-11/16	12/16-1/17
English	SemEval-17 (Rosenthal et al., 2017)	01/12-12/15	12/16-1/17
French	Deft-17 (Benamara et al., 2017)	2014-2016	Same
German	SB-10K (Cieliebak et al., 2017)	8/13-10/13	Same
Hindi	SAIL 2015 (Patra et al., 2015)	NA, 3-month	Same
Italian	Sentipolc-16 (Barbieri et al., 2016)	2013-2016	2016
Portug.	SentiBR (Brum and Nunes, 2017)	1/17-7/17	Same
Spanish	Intertass (Díaz-Galiano et al., 2018)	7/16-01/17	Same

Details:

- Pruned all datasets to the smallest language (Hindi)
- Dataset size: 24,263 tweets (3,033 per language)

Sentiment Analysis Experiments

X-Lingual, zero-shot experiments

	XLM-R								XLM-Twitter									
	Ar	En	Fr	De	Hi	It	Pt	Es	All-I	Ar	En	Fr	De	Hi	It	Pt	Es	All-I
Ar	63.6	64.1	54.4	53.9	22.9	57.4	62.4	62.2	59.2	67.7	66.6	62.1	59.3	46.3	63.0	60.1	65.3	64.3
En	64.2	68.2	61.6	63.5	23.7	68.1	65.9	67.8	68.2	64.0	66.9	60.6	67.8	35.2	67.7	61.6	68.7	70.3
Fr	45.4	52.1	72.0	36.5	16.7	43.3	40.8	56.7	53.6	47.7	59.2	68.2	38.7	20.9	45.1	38.6	52.5	50.0
De	43.5	64.4	55.2	73.6	21.5	60.8	60.1	62.0	63.6	46.5	65.0	56.4	76.1	36.9	66.3	65.1	65.8	65.9
Hi	48.2	52.7	43.6	47.6	36.6	54.4	51.6	51.7	49.9	50.0	55.5	51.5	44.4	40.3	56.1	51.2	49.5	57.8
It	48.8	65.7	63.9	66.9	22.1	71.5	63.1	58.9	65.7	41.9	59.6	60.8	64.5	24.6	70.9	64.7	55.1	65.2
Pt	41.5	63.2	57.9	59.7	26.5	59.6	67.1	65.0	65.0	56.4	67.7	62.8	64.4	26.0	67.1	76.0	64.0	71.4
Es	47.1	63.1	56.8	57.2	26.2	57.6	63.1	65.9	63.0	52.9	66.0	64.5	58.7	30.7	62.4	67.9	68.5	66.2

Zero-shot > monolingual: En->Ar, It->Hi

XLM-T > XLM-R: More robust, best in 6 out of 8 langs.

Training with all langs: Most beneficial for Hindi, also for English (maybe due to shared vocabs?).

With target language training data

	Monolingual		Bilingual		Multilingual		
	FT	XLM-R	XLM-R	XLM-T	XLM-R	XLM-T	
Ar	45.98	63.56	67.67	63.63 (En)	67.65 (En)	64.31	66.89
En	50.85	68.18	66.89	65.07 (It)	67.47 (Es)	68.52	70.63
Fr	54.82	71.98	68.19	73.55 (Sp)	68.24 (En)	70.52	71.18
De	59.56	73.61	76.13	72.48 (En)	75.49 (It)	72.84	77.35
Hi	37.08	36.60	40.29	33.57 (It)	55.35 (It)	53.39	56.39
It	54.65	71.47	70.91	70.43 (Ge)	73.50 (Pt)	68.62	69.06
Pt	55.05	67.11	75.98	71.87 (Sp)	76.08 (En)	69.79	75.42
Sp	50.06	65.87	68.52	67.68 (Po)	68.68 (Pt)	66.03	67.91
All	51.01	64.80	66.82	64.78	69.06	66.75	69.35

Findings

- More training data is better, even if different languages
 - Better off XLM-T than XLM-R
- However, smart lang selection is better in ~50% of cases
 - Typological proximity?
- Obvious trade-offs

Release

Language Models at

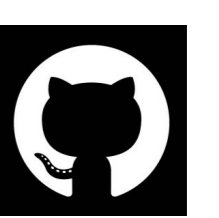
<https://huggingface.co/cardiffnlp>



- Multilingual Twitter-specific Language Model
 - <https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base>
- The above, fine-tuned on sentiment analysis
 - <https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment>

Repository

<https://github.com/cardiffnlp/xlm-t>



Fine-tuning interface

- LM fine-tuning
- Adapters!



Notebooks with starter code

- End-to-end pipeline on UMSAB
 - Download, predict, evaluate
- Extract embeddings from tweets
- Sentiment prediction
- Fine-tuning on custom data

Leaderboard

	FT Mono	XLM-R Mono	XLM-Tw Mono	XLM-R Multi	XLM-Tw Multi
Arabic	46.0	63.6	67.7	64.3	66.9
English	50.9	68.2	66.9	68.5	70.6
French	54.8	72.0	68.2	70.5	71.2
German	59.6	73.6	76.1	72.8	77.3
Hindi	37.1	36.6	40.3	53.4	56.4
Italian	54.7	71.5	70.9	68.6	69.1
Portuguese	55.1	67.1	76.0	69.8	75.4
Spanish	50.1	65.9	68.5	66.0	67.9
All lang.	51.0	64.8	66.8	66.8	69.4

```
def preprocess(text):
    new_text = []
    for t in text.split(" "):
        t = '@user' if t.startswith('@') and len(t) > 1 else t
        t = 'http' if t.startswith('http') else t
        new_text.append(t)
    return " ".join(new_text)

def get_embedding(text):
    text = preprocess(text)
    encoded_input = tokenizer(text, return_tensors='pt')
    features = model(**encoded_input)
    features = features[0].detach().numpy()
    features_mean = np.mean(features[0], axis=0)
    return features_mean
```

```
query = "Acabo de pedir pollo frito 🍗" #spanish

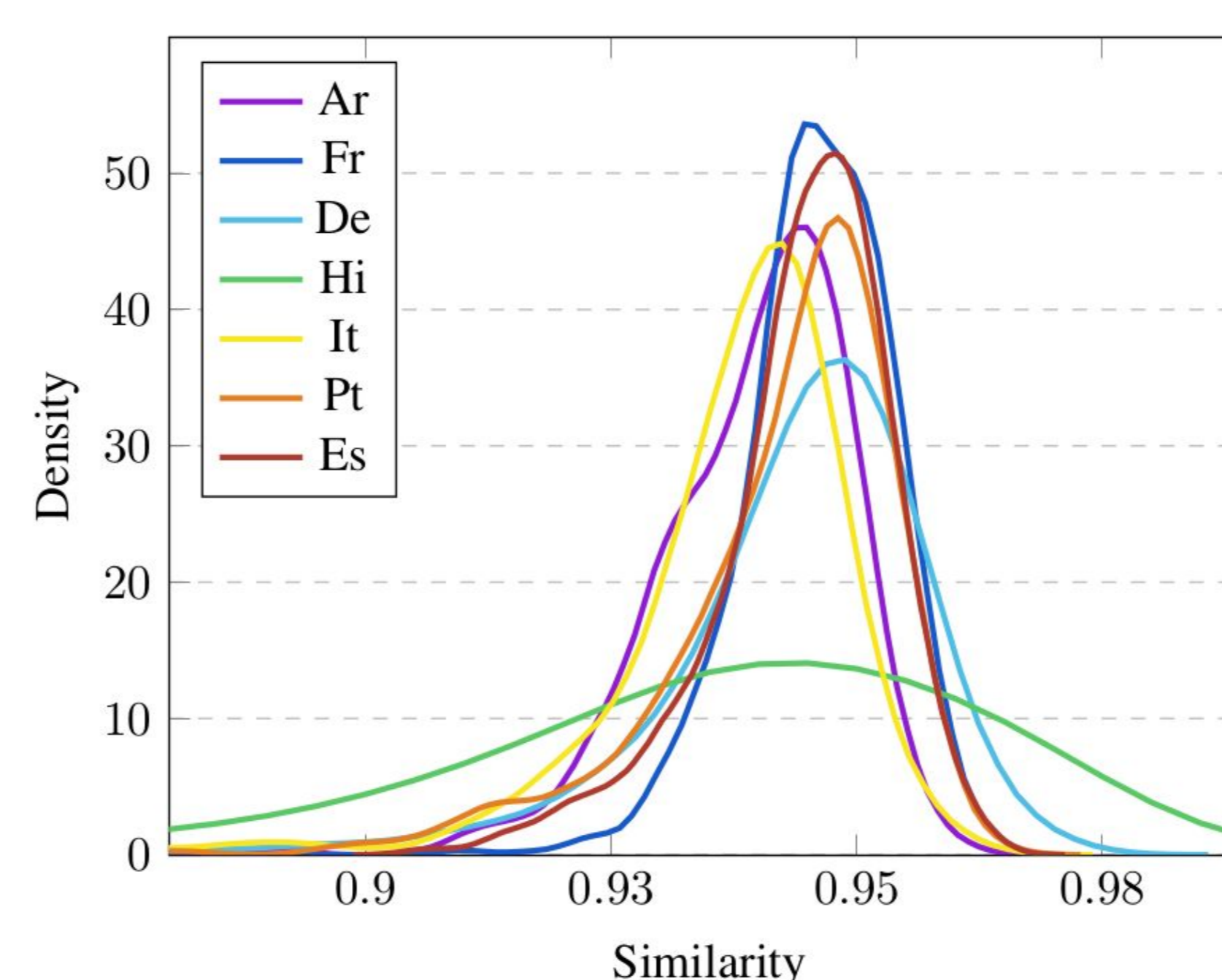
tweets = ["We had a great time! 🎉", # english
          "We hebben een geweldige tijd gehad! 🍷", # dutch
          "Nous avons passé un bon moment! 🍷", # french
          "Ci siamo divertiti! 🍷"] # italian

d = defaultdict(int)
for tweet in tweets:
    sim = 1 - cosine(get_embedding(query), get_embedding(tweet))
    d[tweet] = sim

print('Most similar to: ', query)
print('-----')
for idx, x in enumerate(sorted(d.items(), key=lambda x: x[1], reverse=True)):
    print(idx+1, x[0])
```

```
Most similar to: Acabo de pedir pollo frito 🍗
-----
1 Ci siamo divertiti! 🍷
2 Nous avons passé un bon moment! 🍷
3 We had a great time! 🎉
4 We hebben een geweldige tijd gehad! 🍷
```

Similarity with English



- In addition to lang proximity, topic overlap is important
 - Arabic similar to English, compiled at around same time