Lecture 3

Multilingual modelling

Rochelle Choenni

Today’s topics

- Introduction
  - What are multilingual models?
  - Why do we need them?
- Earlier methods
  - Language transfer and joint learning
  - Word embeddings
- SOTA models and their limitations

INTRO: What are multilingual models?

A language model is called ‘multilingual’ when it can understand many (4+) different languages

Goal: Create a single model that captures **universal language structures** such that it can reason across all known languages

INTRO: In theory..

According to Noam Chomsky’s universal grammar theory:

Linguistic universals are patterns that occur systematically across natural languages. For example, (almost) all languages make a distinction between *nouns* and *verb*s and distinguish *function words* from *content words*.

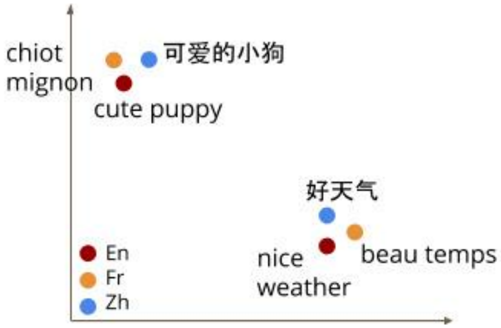
↳ Multilingual models can automatically find such commonalities between languages (on the lexical, syntactic and semantic level) and exploit them i.e. capturing language-agnostic information

INTRO: In practice..

Goal: phrases with similar meaning should obtain similar representations (distributional hypothesis)

Constraints:

- This should be done irrespective of the language



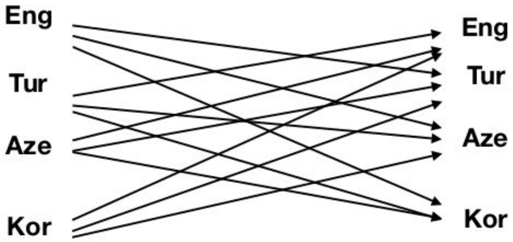
- And without affecting the monolingual semantic relations between the phrases within a language

  Example: the word *'table'* should appear close to its Italian translation *'tavola'* without losing the proximity to *'desk'* which should in turn be close to the Italian translation *'scrittoio'*. (Beinborn et al., 2020)

INTRO: Why do we need multilingual models?

Practical:

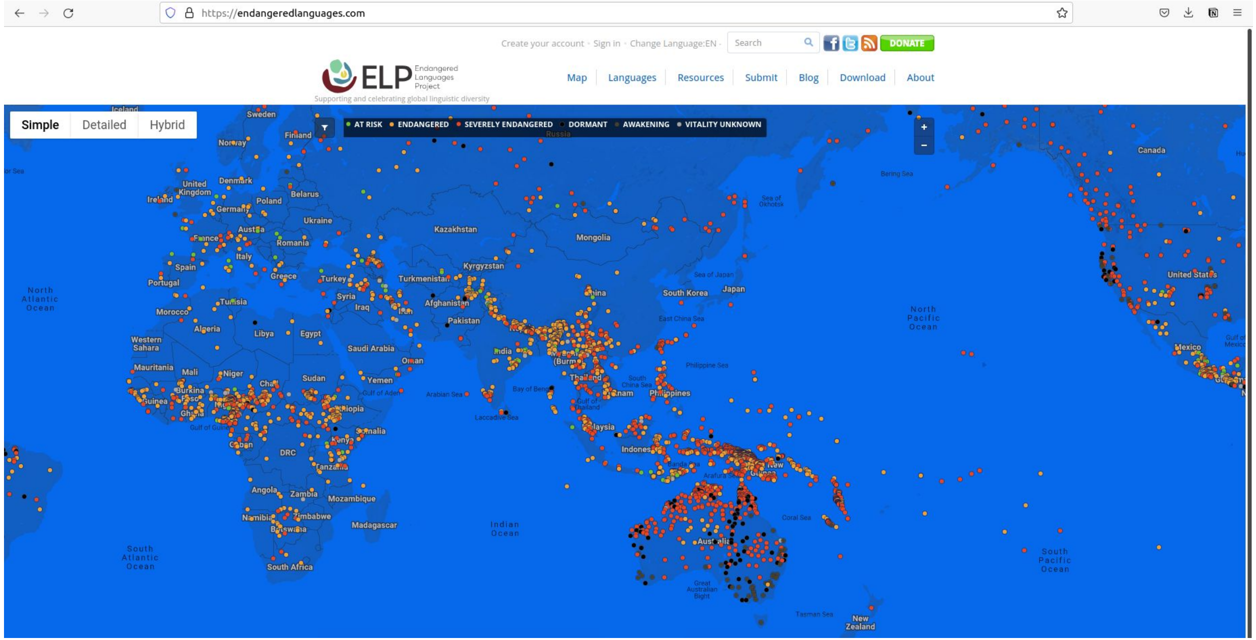
There are over 7K languages spoken in the world today, we don't want to train a model for each one..



- Supporting translation across just 4 languages requires 4*3=12 models
- Across all languages requires us to build .. ~ 49 million models

Social:

We want to extend the benefits of NLP technology to more language communities + capture endangered languages



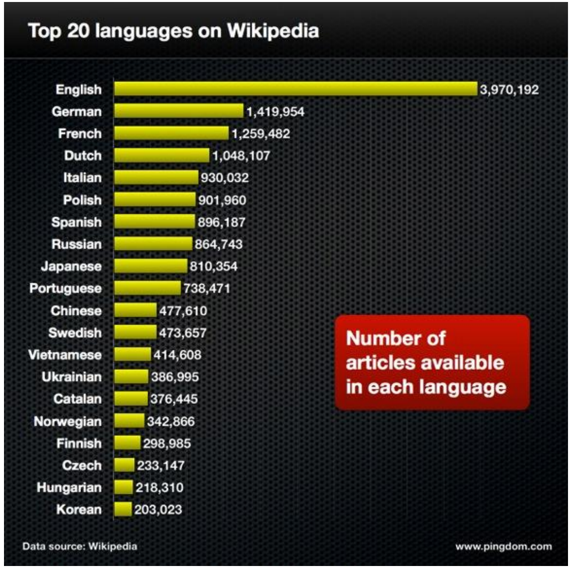
Technical:

SOTA methods are data hungry!

For many languages there’s simply too little data to train a monolingual model successfully

Transformer based models:

- BERT: 13GB (3.4 billion word text corpora)
- GPT2: 40GB…
- RoBERTa: 160GB…
- GPT-3: 45 TB…



Top 20 languages on Wikipedia

Number of articles available in each language

Data source: Wikipedia

www.pingdom.com

Many languages are left behind!

INTRO: Some terminology

In the NLP community we talk about:

- **High resource**: languages for which we have *‘much’* data available

  -> we can generally train good monolingual models

- **Low resource**: languages for which we have *‘too little’* data available (most languages)

Pay attention: each paper can use a different threshold to determine the categorisation!

Approaches: Two solutions to data-scarcity

- Language transfer (cross-lingual transfer):

  Transfer from **high-resource** to **low-resource** languages, hence leveraging information across languages

- Multilingual joint learning:

  Jointly learn from annotations in multiple languages to leverage language interdependencies

For further reading see Ponti et al., 2019

Approaches: Language transfer methods

To leverage useful information from a source language, it typically needs to be manipulated to better suit the properties of the target language first (Ponti et al., 2019)

Earlier methods include:

- **Data transfer** -> facilitate homogeneous use of data

  Annotation projection (Hwa et al., 2002)

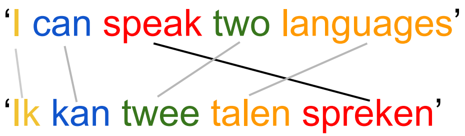
- **Model transfer** -> directly transfer trained model

  Delexicalization (Zeman and Resnik, 2008)

Data transfer – Annotation projection

1. Parse high resource language

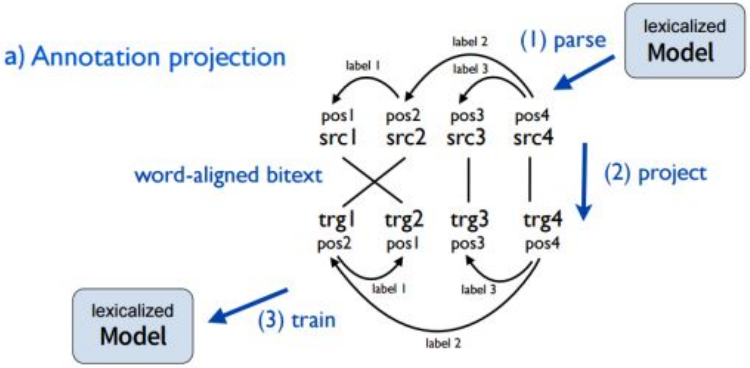
2. Extract word-alignments from parallel corpora:



‘I can speak two languages’

‘Ik kan twee talen spreken’

3. Use created data in the target language for supervised training (Ganchev et al., 2009; Hwa et al., 2005; Yarowsky et al., 2001)



Drawback: noise coming from two sources – parser and word-alignment method
Quite successful: 70% accuracy between English and Spanish

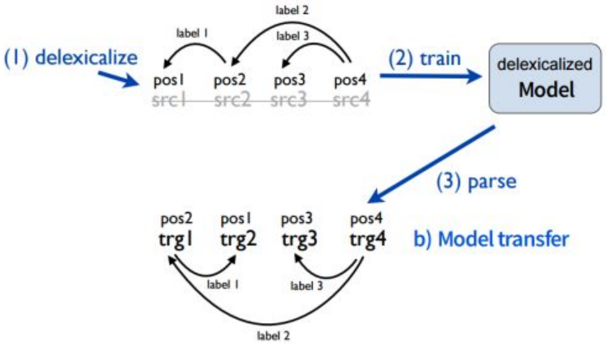
Image taken from Ponti et al., 2019

Model transfer – Delexicalization

1. Delexicalize data to solve for incompatible vocabularies

2. Train model on delexicalized model

3. Directly apply this model to the target language



Delexicalization: replace the words in a language by the corresponding POS tags

-> performance relies on the ability to find robust universal features

Image taken from Ponti et al 2019

Approaches: Limitations

- Doesn’t solve the practical problem -> methods remain inherently bilingual

- Doesn’t solve the social problem -> methods rely on the assumption that high quality resources exist at least for the source language.

  Suppose you want to transfer between:
  English -> Dutch
  ? -> Filipino

  Most languages do not have a suitable high-resource language for transfer

Approaches: Joint learning methods

Learn information from **multiple languages simultaneously** such that they can learn to support each other and thereby jointly enhance each others quality

For instance, French can benefit from Spanish, Italian etc.

Key strategy: **Parameter sharing**-> share (otherwise private) representations

This is still used in SOTA methods today as you will see in a bit!

Parameter sharing

Share (otherwise private) representations e.g., word embeddings (Guo et al., 2016), hidden layers (Duong et al., 2015) or attention mechanisms (Pappas and Popescu-Belis, 2017) across languages

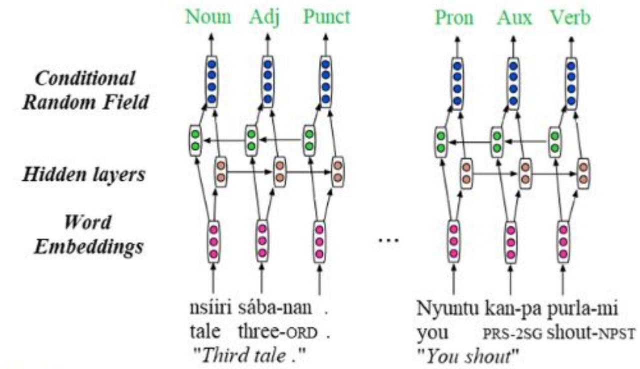


Figure 2
In multilingual joint learning, representations can be private or shared across languages. Tied parameters are shown as neurons with identical color. Image adapted from Fang and Cohn (2017), representing multilingual PoS tagging for Bambara (left) and Warlpiri (right).

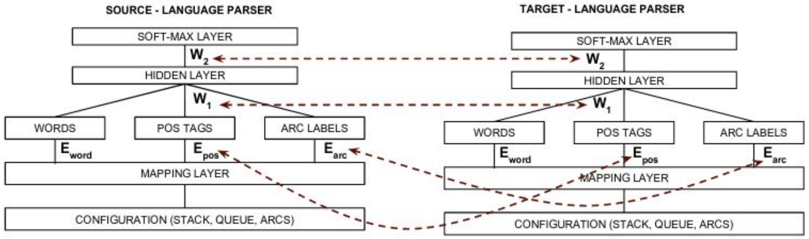


Figure 1: Neural Network Parser Architecture from Chen and Manning (2014) (left). Our model (left and right) with soft parameter sharing between the source and target language shown with dashed lines.

Full parameter sharing: parameter values are identical across languages

Soft parameter sharing: distance between parameters from different language-specific models is minimized

Left image from Ponti et al., 2019; right from Duong et al., 2015

Word embeddings: Different methods

1. **Monolingual mapping**:

   Learn linear mapping between monolingual representations in different languages

2. **Pseudo-cross-lingual**:

   Train a model on a corpus created by mixing contexts of different languages

3. **Cross-lingual training**:

   Optimize a cross-lingual constraint between embeddings of different languages

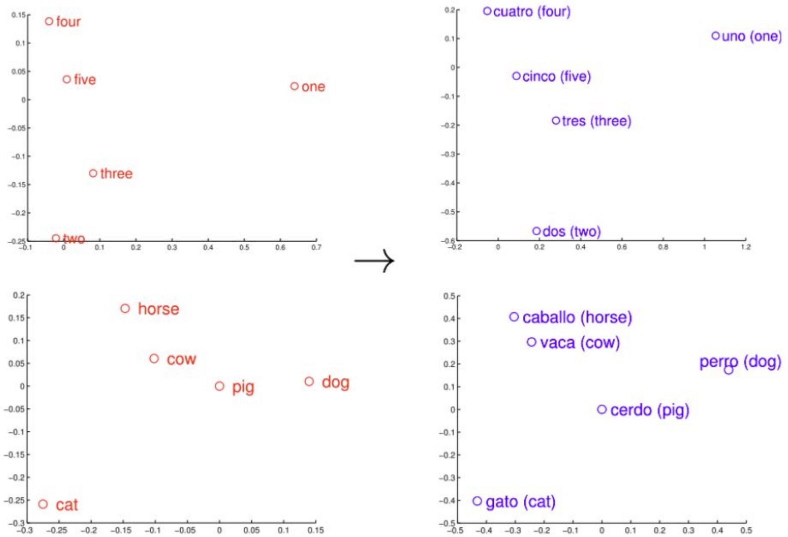
4. **Joint optimization**:

   Jointly optimise a combination of monolingual and cross-lingual losses

For further reading see Ruder et al., 2019a

Word embeddings: Mapping models

Linear projection



Learn a transformation between languages? (Mikolov et al., 2013):

- Use 5K translations as bilingual dictionary
- Learn transformation matrix W using SGD by minimising:

$$\min_{W} \sum_{i=1}^{n} |Wx_i - z_i|^2$$

Xi = monolingual representation of the source word wi

zi = monolingual representation of translation of wi

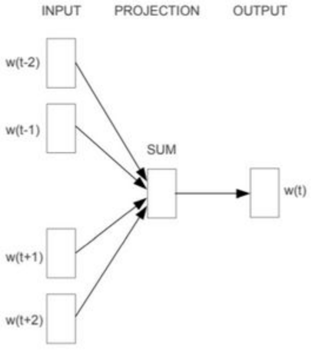
Word embeddings: Pseudo-cross-lingual

Random translation replacement (Gouws et al., 2015):

- Google Translate pairs of words in the source and target language
- Concatenate + shuffle source and target corpus
- Replace each word with its translation with a probability of 50% e.g.:

  ‘build the house’ -> construire the house, build la maison etc.

- Train CBOW on this corpus



CBOW

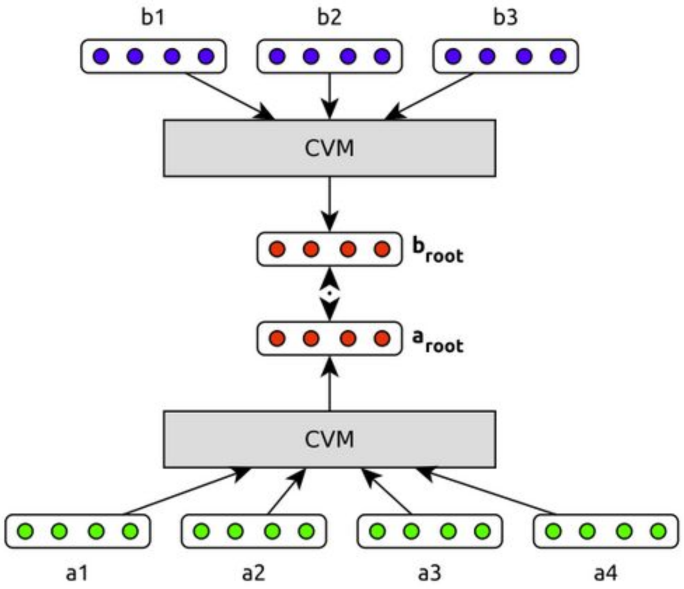
Word embeddings: Cross-lingual training

Bilingual compositional sentence model (Hermann et al., 2013):

- Train two models to produce sentence representations of parallel sentences in two languages
- Use the distance between the two sentence representations as objective
- Minimise the following loss:

$$E_{dist}(a,b) = |a_{\text{root}} - b_{\text{root}}|^2$$

where a_{root} and b_{root} are the representations of two aligned sentences from different languages

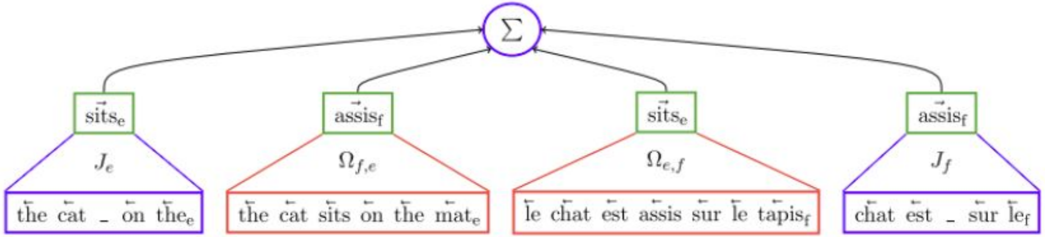


Word embeddings: Joint optimization

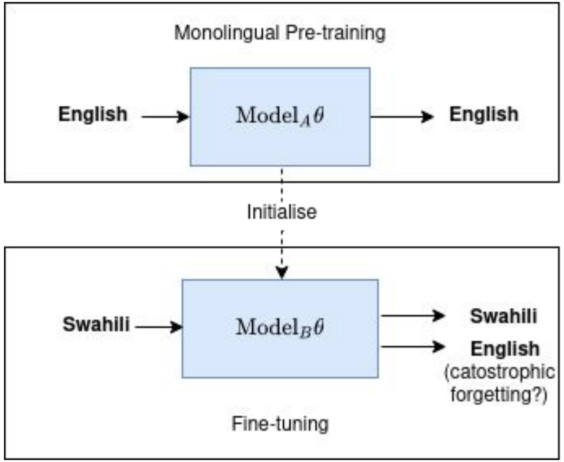
Trans-gram (Coulmance et al., 2015):

Train on a combination of monolingual and cross-lingual objectives

- 2 monolingual skip-gram losses: J_e (English) and J_f (French)
- and 2 cross-lingual trans-gram losses: $\Omega_{f,e}$ (French->English) and $\Omega_{e,f}$ (English->French)


Σ
$\overrightarrow{sits}_e$
J_e
$\overleftarrow{the}$ $\overleftarrow{cat}$ _ $\overleftarrow{on}$ $\overleftarrow{the}_e$
$\overrightarrow{assis}_f$
$\Omega_{f,e}$
$\overleftarrow{the}$ $\overleftarrow{cat}$ $\overleftarrow{sits}$ $\overleftarrow{on}$ $\overleftarrow{the}$ $\overleftarrow{mat}_e$
$\overrightarrow{sits}_e$
$\Omega_{e,f}$
$\overleftarrow{le}$ $\overleftarrow{chat}$ $\overleftarrow{est}$ assis $\overleftarrow{sur}$ $\overleftarrow{le}$ $\overleftarrow{tapis}_f$
$\overrightarrow{assis}_f$
J_f
$\overleftarrow{chat}$ $\overleftarrow{est}$ _ $\overleftarrow{sur}$ $\overleftarrow{le}_f$

SOTA approaches: Cross-lingual Transfer



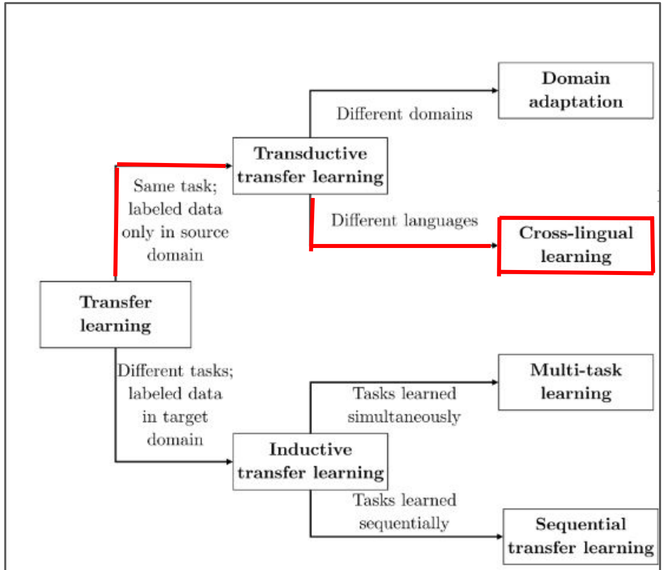
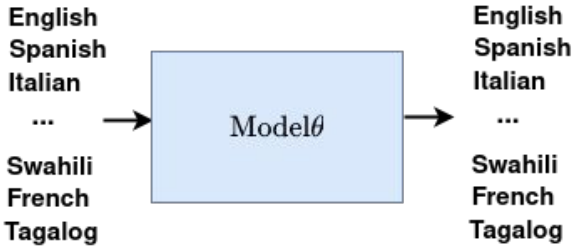

Domain adaptation
Different domains
Transductive transfer learning
Same task; labeled data only in source domain
Different languages
Cross-lingual learning
Transfer learning
Different tasks; labeled data in target domain
Multi-task learning
Tasks learned simultaneously
Inductive transfer learning
Tasks learned sequentially
Sequential transfer learning

Image adapted from Ruder 2019b

Different options:

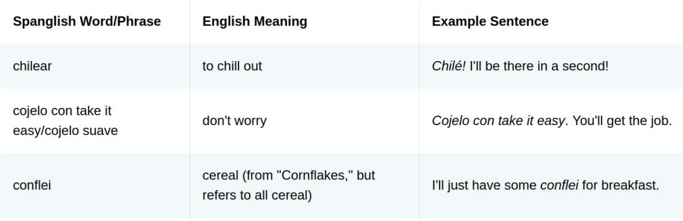
- Pre-training on large dataset and Fine-tuning on large dataset (regular)
- Pre-training on large dataset and Fine-tuning on small dataset (**few-shot**)
- Pre-training on large dataset no Fine-tuning on the test language (**zero-shot**)

SOTA approaches: Multilingual joint learning



Train one single model on a mixture of data from multiple languages

- Full parameter sharing
- Code switching!:



Spanglish Word/Phrase	English Meaning	Example Sentence
chilear	to chill out	*Chilé!* I'll be there in a second!
cojelo con take it easy/cojelo suave	don't worry	*Cojelo con take it easy*. You'll get the job.
conflei	cereal (from "Cornflakes," but refers to all cereal)	I'll just have some *conflei* for breakfast.

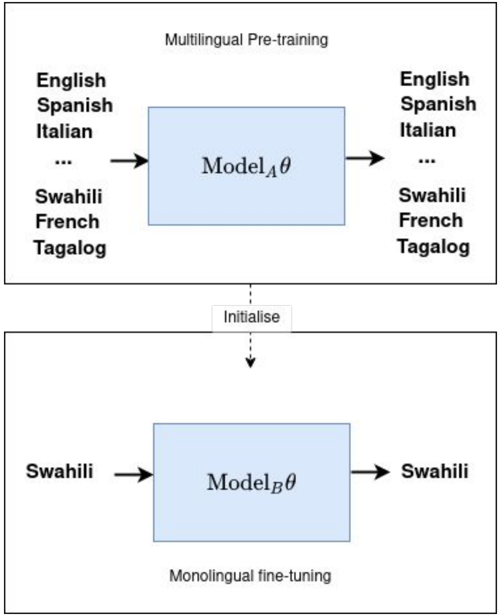
SOTA approaches: Best practice

SOTA sentence encoders commonly use:

A combination of **cross-lingual transfer** and **multilingual joint learning**

+

A **monolingual** or **cross-lingual training objective** in combination with **different architectures** (e.g. LSTMs or Transformers) + **tokenization!**



SOTA approaches: Pre-training objectives

- Monolingual: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP)

  -> Inexpensive, easier to expand the number of train languages

  -> No cross-lingual signal

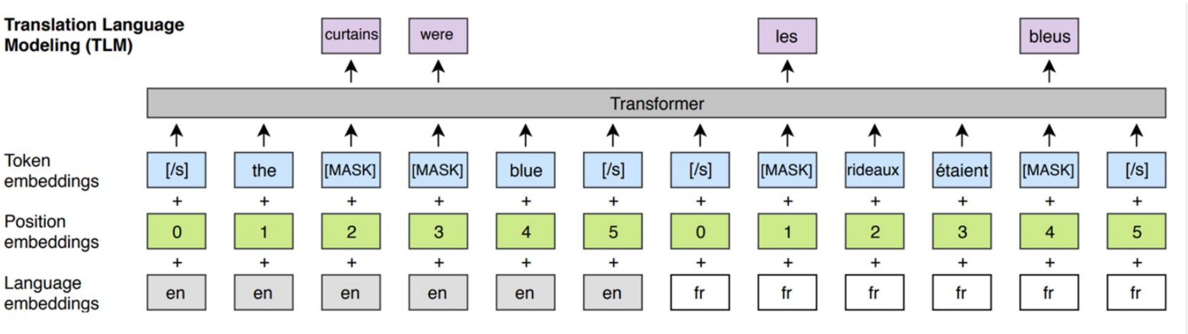
- Cross-lingual**:** Machine Translation (MT) and Translation Masked Modelling (TLM)

  -> Tasks designed to force the model to understand patterns across languages

  -> Requires parallel corpora

SOTA approaches: Translation Language Modelling (TLM)

The model can leverage information from the context in either language to predict the words, thereby encouraging the alignment of representations in both languages.



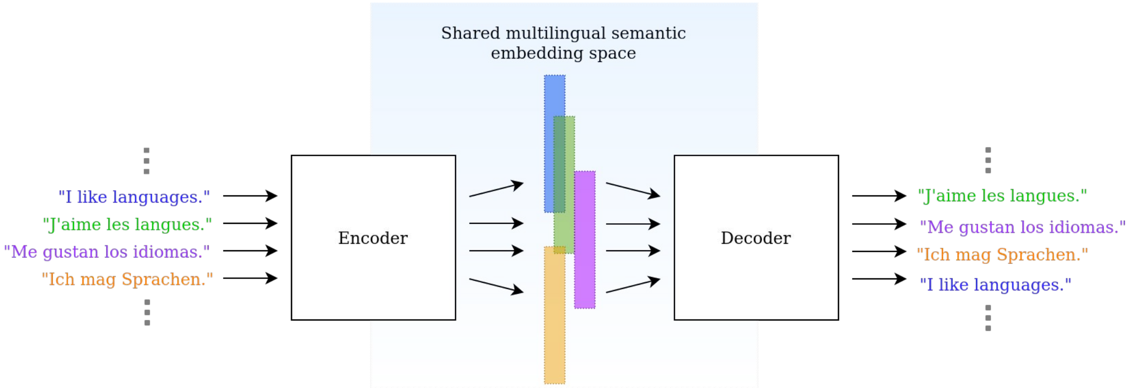
‘build the house’ -> construire the house, build la maison etc.

Image taken from Conneau et al., 2019

Models: LASER

The first multilingual sentence encoder! (93 languages - 30 families, 28 scripts) (Artexte et al., 2019)

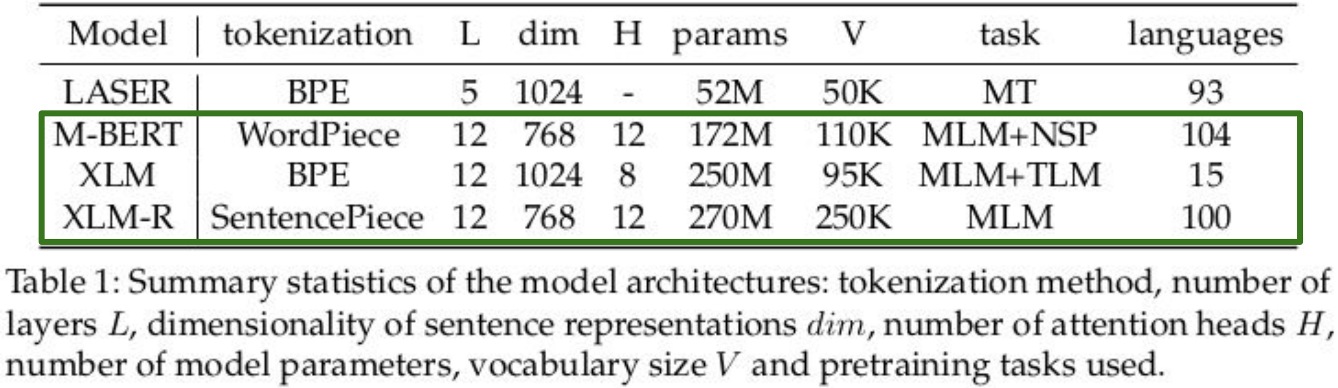
- Training objective: Machine translation
- Encoder/decoder type: BiLSTM
- Key: Encoder and decoder are jointly trained on parallel corpora (end-to-end)



The decoder functions as a feedback generator to the encoder

When training stabilizes, the decoder is discarded and the encoder can be used as multilingual model

Models: BERT-based models



Model	tokenization	L	dim	H	params	V	task	languages
LASER	BPE	5	1024	-	52M	50K	MT	93
M-BERT	WordPiece	12	768	12	172M	110K	MLM+NSP	104
XLM	BPE	12	1024	8	250M	95K	MLM+TLM	15
XLM-R	SentencePiece	12	768	12	270M	250K	MLM	100

Table 1: Summary statistics of the model architectures: tokenization method, number of layers L, dimensionality of sentence representations dim, number of attention heads H, number of model parameters, vocabulary size V and pretraining tasks used.

- M-BERT = BERT + more diverse data (Devlin et al., 2018)
- XLM = BERT-based architecture - NSP + TLM + data from 15 languages (Conneau et al., 2019)
- XLM-R = RoBERTa - NSP + more diverse data (Conneau et al,. 2020)

SOTA LLMs: GPT-3, LLAMA, BLOOM, PALM..

SOTA models: Data collection

How to add data from e.g. 104 languages?

Exponentially smoothed weighting:

- P(en) -> 21% of data is English
- Exponentiate each prob by factor *S* -> re-normalize -> sample from new distribution

Under-sampled English, Oversample Icelandic:

Old: English sampled 1000x more than Icelandic

After smoothing it's ***only*** sampled 100x more!

In practice: data is still very skewed!

Shouldn’t this result in an exploding vocabulary size?

1 language BERT: ~30K Vocab -> 100 languages: ~3M vocab?

SOTA: Subword Tokenization

Split rare words into frequent subwords: e.g. “*reconstructing*” -> “re” - “construct” - “ing”

BERT: ~30K Vocab - 1 language -> M-BERT: **only** ~110K Vocab - 104 languages!

Byte Pair Encoding (BPE) (Sennrich et al., 2016):

1. Init *base_vocab* using unique symbols and characters + set vocab size *V* (hyperparameter)
2. Split each word into the base vocabulary characters e.g.: **[(‘c’,’a’,’r’ , 5), (‘c’,’a’,’b’,’l’,’e’, 3), (‘w’,’a’,’t’,’c’,’h’, 2), (‘c’,’h’,’a’,’i’,’r’, 5)]**
3. While len(*base_vocab*) < *V*:
   a. Count the occurrence of every symbol pair and pick the one with the highest frequency
   b. Add symbol pair to *base_vocab* + merge all occurences of the symbol pair

E.g.: The pair “*ca*” occurs 5 x in *car* + 3 x in *cable* = 8 occurrences

-> *base_vocab* += [“ca”] + **[(‘ca’,’r’ , 5), (‘ca’,’b’,’l’,’e’, 3), (‘w’,’a’,’t’,’c’,’h’, 2), (‘c’,’h’,’a’,’i’,’r’, 5)]**

The pair “ch” is occurs 2 x in *watch* and 5 x in *chair* = 7 occurrences

-> *base_vocab* += [“ch”] + **[(‘ca’,’r’ , 5), (‘ca’,’b’,’l’,’e’, 3), (‘w’,’a’,’t’,’ch’, 2), (‘ch’,’a’,’i’,’r’, 5)]**

SOTA: Subword Tokenization

WordPiece (Schuster et al., 2012):

1. Init *base_vocab* using unique symbols and characters + set vocab size *V* (hyperparameter)
2. Train language model *M* on *base_vocab*
3. While len(*base_vocab*) < *V*:
    a. Pick the pair that maximizes the likelihood of the train data
    b. Add symbol pair to *base_vocab* + merge all occurences of the symbol pair

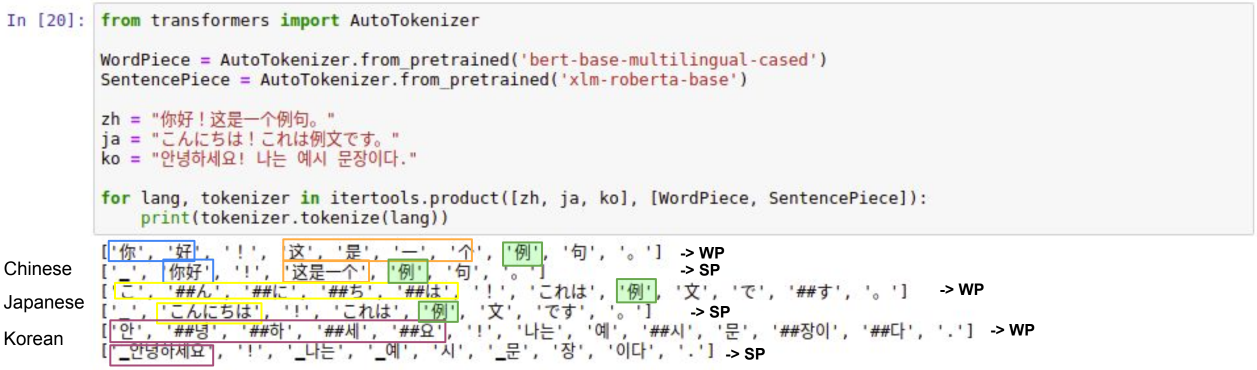
E.g.: Pick "*ca*" if p(ca)/p(c)p(a) > any other symbol pair in vocab

SOTA: Subword Tokenization

BPE and WordPiece are created for English-> Some languages do not split words by spaces (e.g. Chinese)!!

Solutions:

- WordPiece: add white space around characters and perform character tokenization for corner cases – Quick fix
- SentencePiece (Kudo et al., 2018): does not treat space as a separator, it takes the string as input in its original raw format, i.e. along with all spaces. It then uses e.g. BPE as its tokenizer to construct the vocabulary (size has grown to 250K)



```
In [20]: from transformers import AutoTokenizer

         WordPiece = AutoTokenizer.from_pretrained('bert-base-multilingual-cased')
         SentencePiece = AutoTokenizer.from_pretrained('xlm-roberta-base')

         zh = "你好！这是一个例句。"
         ja = "こんにちは！これは例文です。"
         ko = "안녕하세요! 나는 예시 문장이다."

         for lang, tokenizer in itertools.product([zh, ja, ko], [WordPiece, SentencePiece]):
             print(tokenizer.tokenize(lang))
```

```
Chinese   ['你', '好', '!', '这', '是', '一', '个', '例', '句', '。']  -> WP
          ['_', '你好', '!', '这是一个', '例', '句', '。']  -> SP
Japanese  ['こ', '##ん', '##に', '##ち', '##は', '!', 'これは', '例', '文', 'で', '##す', '。']  -> WP
          ['_', 'こんにちは', '!', 'これは', '例', '文', 'です', '。']  -> SP
Korean    ['안', '##녕', '##하', '##세', '##요', '!', '나는', '예', '##시', '문', '##장이', '##다', '.']  -> WP
          ['_안녕하세요', '!', '_나는', '_예', '시', '_문', '장', '이다', '.']  -> SP
```

There is no language detection, in the multilingual setting the tokenizer can mix up languages

SOTA: Subword Tokenization

Tokenization gets little attention but:

1. It prevents the vocab and model size from exploding
2. OOV words are rare
3. Better equipped to handle minor misspellings

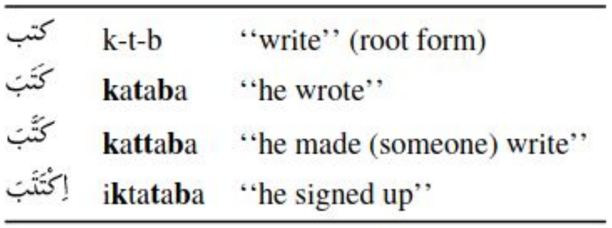
   -> reconstuctin = re - construct - in

4. It allows for easy adaption of models to the multilingual setting

SOTA: Subword Tokenization

Linguistic pitfalls:

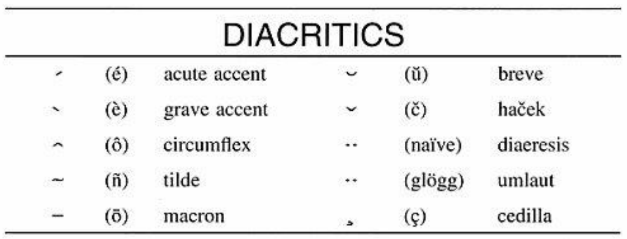
- Still not suitable for some languages that do not rely on word splitting e.g. Arabic:



كتب	k-t-b	‘‘write’’ (root form)
كَتَبَ	**kataba**	‘‘he wrote’’
كَتَّبَ	**kattaba**	‘‘he made (someone) write’’
إِكْتَتَبَ	**iktataba**	‘‘he signed up’’

Table 1: Non-concatenative morphology in Arabic.[3] When conjugating, letters are interleaved *within* the root. The root is therefore not separable from its inflection via any contiguous split.

- Difficult pre-processing trade-offs: Lowercase? Remove punctuation? Remove diacritics?



DIACRITICS

´	(é)	acute accent	˘	(ŭ)	breve
`	(è)	grave accent	ˇ	(č)	haček
ˆ	(ô)	circumflex	¨	(naïve)	diaeresis
˜	(ñ)	tilde	¨	(glögg)	umlaut
¯	(ō)	macron	¸	(ç)	cedilla

Left example from Clark et al., 2022, also good source for further reading

SOTA: Successful or not?

(RECAP) Approach:

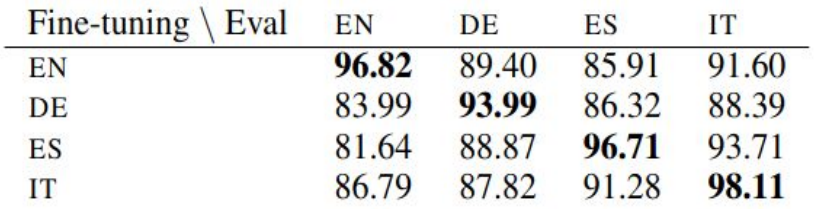
1. **Pretrain** multilingual BERT (M-BERT) -> yields multilingual general-purpose representations
2. **Fine-tune** the general purpose model on a high-resource language for e.g. the task of Part-of-speech tagging -> yields a task-specific model
3. Test the task-specific model on a different language -> **zero-shot transfer**

Test: does the model learn truly universal structures?

Results from Pires et al. 2020

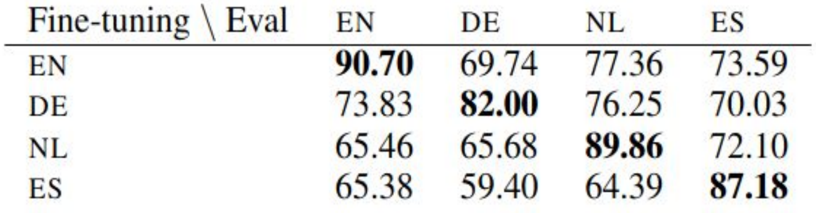
SOTA: Successful or not?

Surprisingly good zero-shot results!



Fine-tuning \ Eval	EN	DE	ES	IT
EN	**96.82**	89.40	85.91	91.60
DE	83.99	**93.99**	86.32	88.39
ES	81.64	88.87	**96.71**	93.71
IT	86.79	87.82	91.28	**98.11**

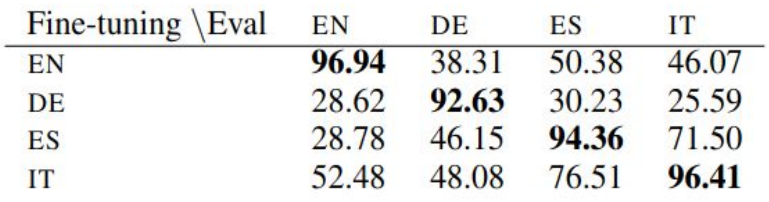
Table 2: POS accuracy on a subset of UD languages.



Fine-tuning \ Eval	EN	DE	NL	ES
EN	**90.70**	69.74	77.36	73.59
DE	73.83	**82.00**	76.25	70.03
NL	65.46	65.68	**89.86**	72.10
ES	65.38	59.40	64.39	**87.18**

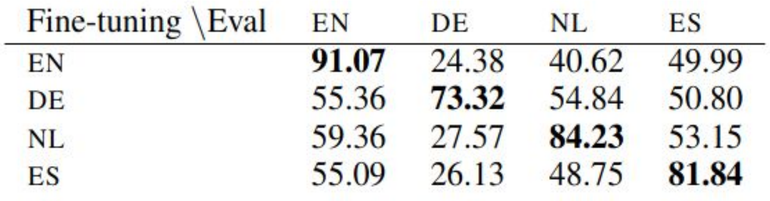
Table 1: NER F1 results on the CoNLL data.

vs



Fine-tuning \Eval	EN	DE	ES	IT
EN	**96.94**	38.31	50.38	46.07
DE	28.62	**92.63**	30.23	25.59
ES	28.78	46.15	**94.36**	71.50
IT	52.48	48.08	76.51	**96.41**

Table 8: POS accuracy on the UD test sets for a subset of European languages using EN-BERT.

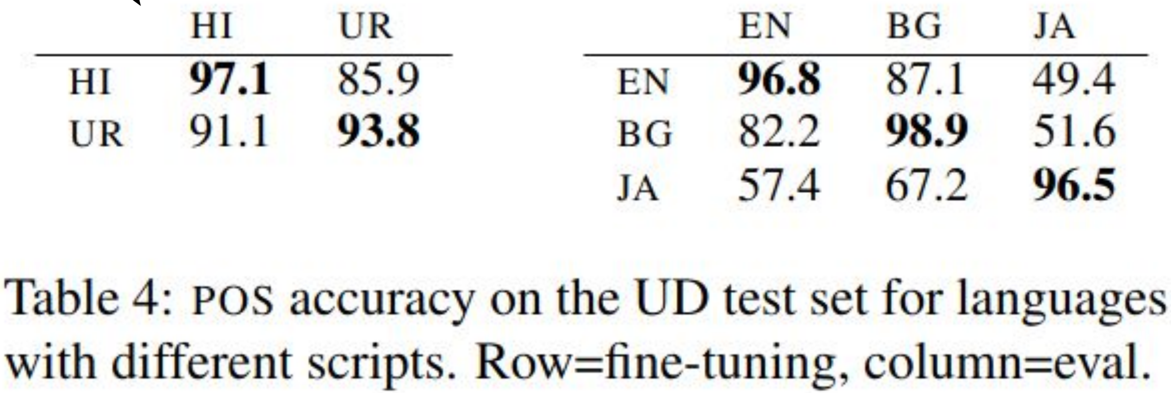


Fine-tuning \Eval	EN	DE	NL	ES
EN	**91.07**	24.38	40.62	49.99
DE	55.36	**73.32**	54.84	50.80
NL	59.36	27.57	**84.23**	53.15
ES	55.09	26.13	48.75	**81.84**

Table 7: NER results on the CoNLL test sets for EN-BERT.

SOTA: Successful or not?

WOW!



	HI	UR
HI	**97.1**	85.9
UR	91.1	**93.8**

	EN	BG	JA
EN	**96.8**	87.1	49.4
BG	82.2	**98.9**	51.6
JA	57.4	67.2	**96.5**

Table 4: POS accuracy on the UD test set for languages with different scripts. Row=fine-tuning, column=eval.

It gets more difficult when transferring between ‘less similar’ languages

But how can we define similarity?

Results from Pires et al. 2020

Defining language similarity

Different approaches e.g. lexical overlap: writing systems, vocabulary overlap (e.g. shared WordPieces)

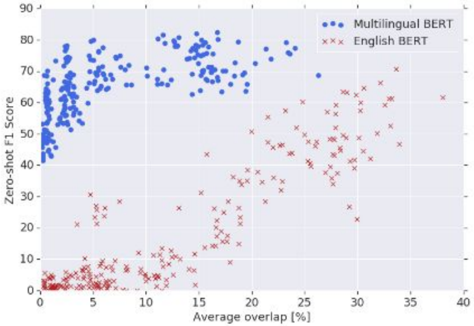


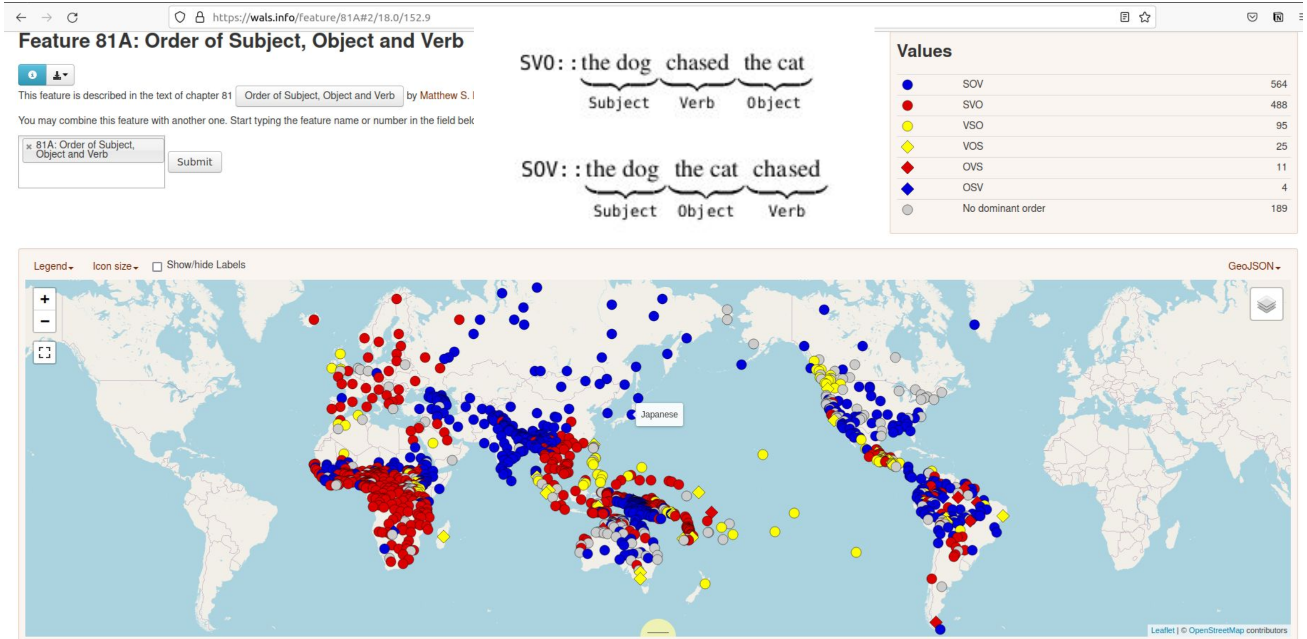
Figure 1: Zero-shot NER F1 score versus entity word piece overlap among 16 languages. While performance using EN-BERT depends directly on word piece overlap, M-BERT's performance is largely independent of overlap, indicating that it learns multilingual representations deeper than simple vocabulary memorization.

Transferability not dependent on lexical overlap.
Other possible explanations?

Results from Pires et al. 2020

Defining language similarity

Linguistic Typology studies, categorizes and documents the variation in the world's languages through systematic cross-linguistic comparisons (Croft, 2002)



https://wals.info/feature/81A#2/18.0/152.9

Feature 81A: Order of Subject, Object and Verb

This feature is described in the text of chapter 81 Order of Subject, Object and Verb by Matthew S. I

You may combine this feature with another one. Start typing the feature name or number in the field bel

× 81A: Order of Subject, Object and Verb

Submit

$$\text{SVO: : } \underbrace{\text{the dog}}_{\text{Subject}}\ \underbrace{\text{chased}}_{\text{Verb}}\ \underbrace{\text{the cat}}_{\text{Object}}$$

$$\text{SOV: : } \underbrace{\text{the dog}}_{\text{Subject}}\ \underbrace{\text{the cat}}_{\text{Object}}\ \underbrace{\text{chased}}_{\text{Verb}}$$

Values

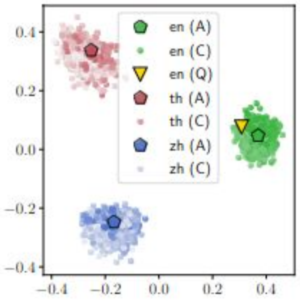
●	SOV	564
●	SVO	488
●	VSO	95
◆	VOS	25
◆	OVS	11
◆	OSV	4
●	No dominant order	189

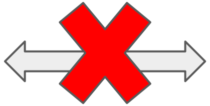
Japanese is a SOV language, Bulgarian and English are SVO -> maybe that's why transfer is easier between the latter?

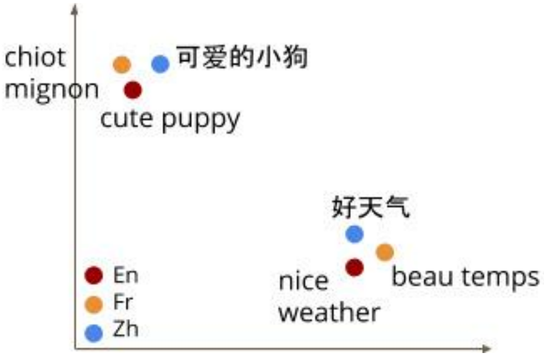
SOTA: Successful or not?

Surprisingly good results!

… but does it really create universal representations?

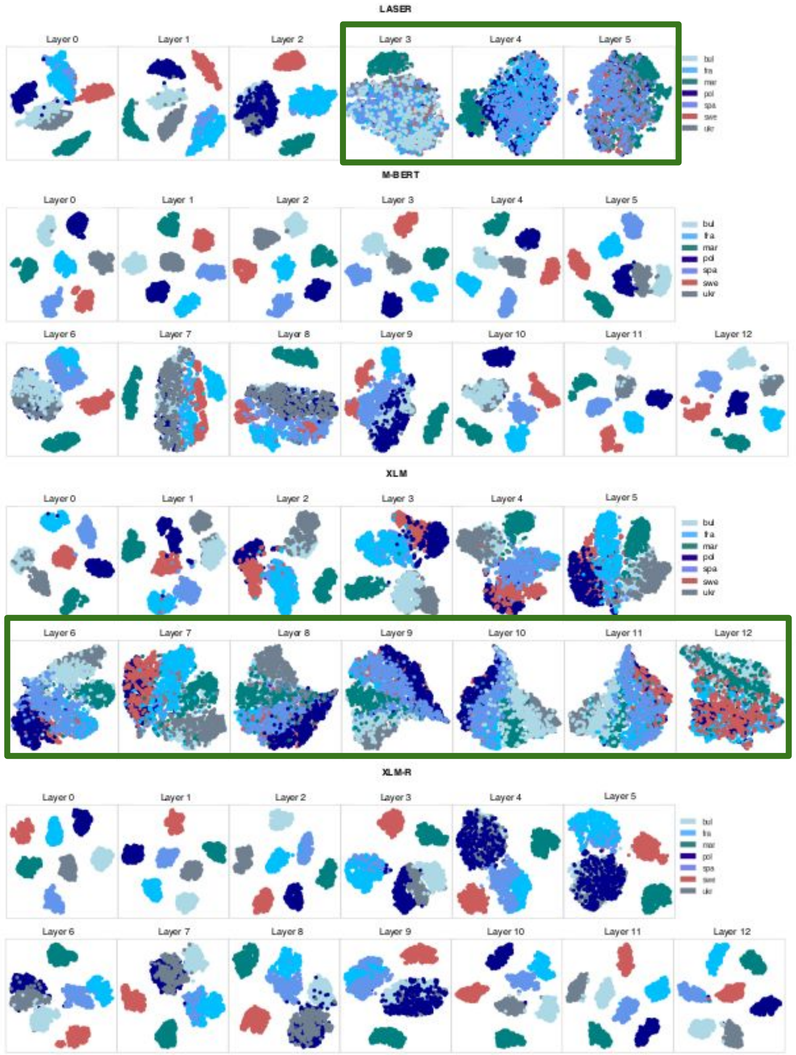


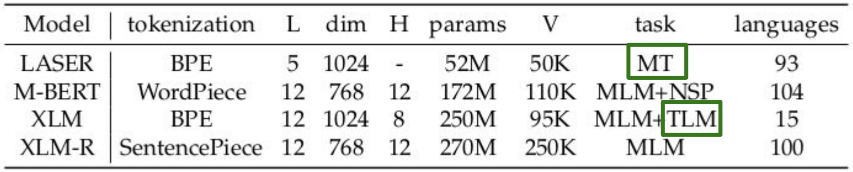




Result: sentence representations from M-BERT are clustered by language

Original goal: cluster sentences with similar meaning together irrespective of language





Model	tokenization	L	dim	H	params	V	task	languages
LASER	BPE	5	1024	-	52M	50K	MT	93
M-BERT	WordPiece	12	768	12	172M	110K	MLM+NSP	104
XLM	BPE	12	1024	8	250M	95K	MLM+TLM	15
XLM-R	SentencePiece	12	768	12	270M	250K	MLM	100

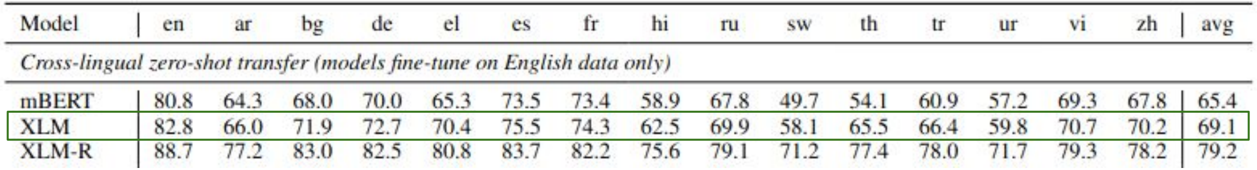
Table 1: Summary statistics of the model architectures: tokenization method, number of layers L, dimensionality of sentence representations dim, number of attention heads H, number of model parameters, vocabulary size V and pretraining tasks used.

Cross-lingual pre-training seems to result in more universal representations?

For further reading see Choenni & Shutova 2022

Cross-lingual pre-training seems to result in more universal representations?

Zero-shot performance on XNLI



Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	avg
Cross-lingual zero-shot transfer (models fine-tune on English data only)																
mBERT	80.8	64.3	68.0	70.0	65.3	73.5	73.4	58.9	67.8	49.7	54.1	60.9	57.2	69.3	67.8	65.4
XLM	82.8	66.0	71.9	72.7	70.4	75.5	74.3	62.5	69.9	58.1	65.5	66.4	59.8	70.7	70.2	69.1
XLM-R	88.7	77.2	83.0	82.5	80.8	83.7	82.2	75.6	79.1	71.2	77.4	78.0	71.7	79.3	78.2	79.2

… but in practice not worth the extra cost?

Results taken from Xue et al. 2021

SOTA: Successful or not?

Problems with evaluation:

1. We actually only know performance on a handful of languages
2. The languages for which we can measure performance tend to be typologically similar

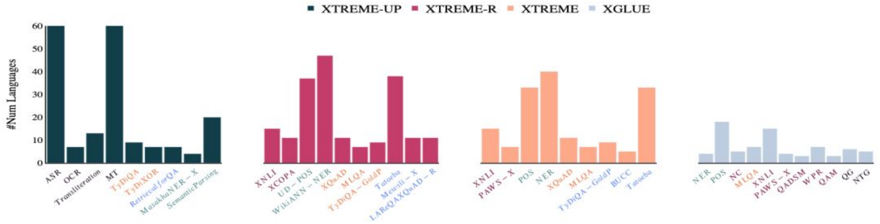


Figure 1: We report the datasets included in each benchmark along with the number of languages that they cover. The datasets are color-coded by type of task: classification, retrieval, question answering, or structured prediction.

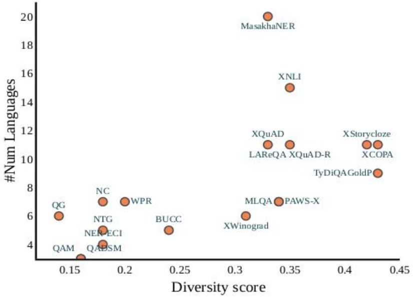


Figure 2: Number of test languages for each task and the average typological diversity score between them computed as the average cosine similarity between URIEL features of each language pair.

Problems: Balance

Multilingual models need to:

1. To generalize over many different languages by finding ‘universal’ representations *(language-agnostic information)*
2. Yet at the same time still capture enough subtle nuances of each individual language *(language-specific information)*

Finding a perfect balance is hard!

Problems: Conflict of interests

The curse of multilinguality: Languages will start fighting for model capacity

-> When performance improves for some languages others start to suffer

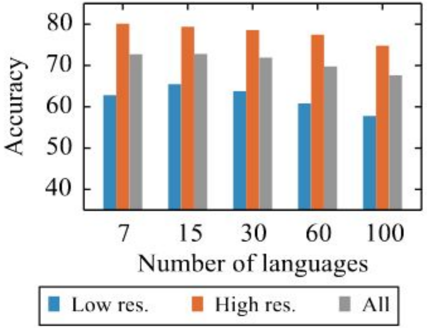


Figure 2: The transfer-interference trade-off: Low-resource languages benefit from scaling to more languages, until dilution (interference) kicks in and degrades overall performance.

Figure from Conneau et al. 2020

Problems: Conflict of interests

Negative interference: Performance on high resource languages for which we normally obtain good results deteriorate

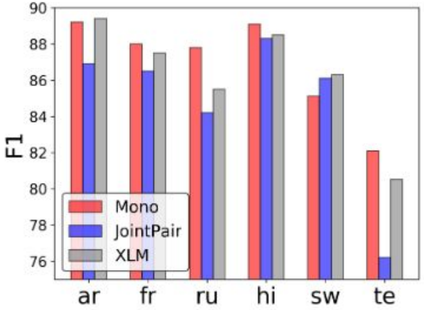


Figure 1: Comparing monolingual vs multilingual models on NER. Lower performance of multilingual models is likely an indicator of negative interference.

Figure from Wang et al. 2020

New directions: Modular deep learning

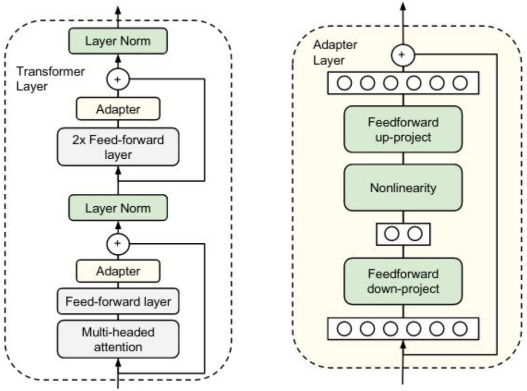
- Modularity definition:

  The correspondence between strongly interconnected components of a system (i.e., modules) and the functions they perform (Baldwin & Clark, 2000; Ulrich, 1995).

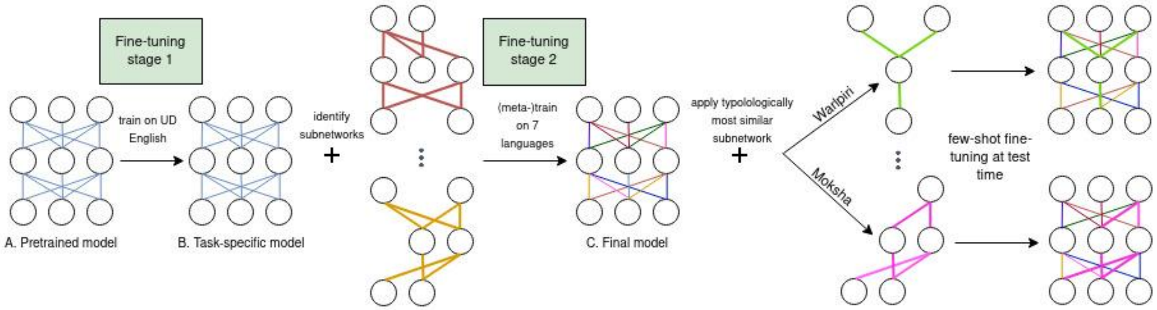
- Each module is specialised for a unique purpose, for which it is reused consistently
- Solution to the curse of multilinguality: disentangle fully shared models using specialised modules for individual languages
- Common approaches: adapter modules and sparse fine-tuning with subnetworks

New directions: Adapters

- Introduced by Houlsby et al. 2019 for more efficient transfer
- Instead of updating all weights during fine-tuning a *few* trainable parameters are added per task
- Traditional fine-tuning: add a new layer to fit the targets specified in the downstream task, and train the new layer together with the pretrained weights
- Adapter tuning strategy: inject new layers (randomly initialized) into the original network. Parameter sharing between tasks is supported by keeping the pretrained model parameters frozen



New directions: Sparse fine-tuning with subnetworks



- This framework relies on the notion that the knowledge for different languages is somehow localizable in specific sets of model parameters
- , and that those parameters can individually be fine-tuned in an autonomous and parameter-efficient manner

Questions?

References

- Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597-610.
- Beinborn, L., & Choenni, R. (2020). Semantic drift in multilingual representations. *Computational Linguistics*, *46*(3), 571-603.
- Choenni, R., & Shutova, E. (2022). Investigating language relationships in multilingual sentence encoders through the lens of linguistic typology. *Computational Linguistics*, 1-37.
- Clark, J. H., Garrette, D., Turc, I., & Wieting, J. (2022). Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, *10*, 73-91.
- Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. *Advances in neural information processing systems*, *32*.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020, July). Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440-8451).
- Coulmance, J., Marty, J. M., Wenzek, G., & Benhalloum, A. (2015). Trans-gram, Fast Cross-lingual Word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1109-1113).
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Duong, L., Cohn, T., Bird, S., & Cook, P. (2015, July). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)* (pp. 845-850).
- Gouws, S., & Søgaard, A. (2015). Simple task-specific bilingual word embeddings. NAACL, 1302–1306.
- Guo, J., Che, W., Wang, H., & Liu, T. (2016, December). A universal framework for inductive transfer parsing across multi-typed treebanks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 12-22).
- Hermann, K. M., & Blunsom, P. (2013). Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*.
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR

- Kudo, T., & Richardson, J. (2018, November). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 66-71).
- Lin, Y. H., Chen, C. Y., Lee, J., Li, Z., Zhang, Y., Xia, M., ... & Neubig, G. (2019, July). Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Vol. 57).
- Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- Pappas, N., & Popescu-Belis, A. (2017). Multilingual Hierarchical Attention Networks for Document Classification. In *8th International Joint Conference on Natural Language Processing (IJCNLP)*
- Pires, T., Schlinger, E., & Garrette, D. (2019, July). How Multilingual is Multilingual BERT?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4996-5001).
- Platanios, E. A., Sachan, M., Neubig, G., & Mitchell, T. (2018). Contextual Parameter Generation for Universal Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 425-435).
- Ponti, E. M., O’horan, H., Berzak, Y., Vulić, I., Reichart, R., Poibeau, T., ... & Korhonen, A. (2019). Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, *45*(3), 559-601.
- Hwa, R., Resnik, P., Weinberg, A., & Kolak, O. (2002, July). Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 392-399).
- Ruder, S., Vulić, I., & Søgaard, A. (2019a). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, *65*, 569-631.
- Ruder, S. (2019b). *Neural transfer learning for natural language processing* (Doctoral dissertation, NUI Galway).
- Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1715-1725).
- Schuster, M., & Nakajima, K. (2012, March). Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5149-5152). IEEE.
- Tiedemann, J., Agić, Ž., & Nivre, J. (2014, June). Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning* (pp. 130-140).
- Üstün, A., Bisazza, A., Bouma, G., & van Noord, G. (2020, November). UDapter: Language Adaptation for Truly Universal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 2302-2315).
- Wang, Z., Lipton, Z. C., & Tsvetkov, Y. (2020, November). On Negative Interference in Multilingual Models: Findings and A Meta-Learning Treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4438-4450).
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... & Raffel, C. (2021, January). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *NAACL-HLT*.
- Zeman, D., & Resnik, P. (2008). Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.