Morphology in the Age of Pre-trained Language Models

Judit Ács

ELKH SZTAKI
acs.judit@sztaki.hu

February 14, 2024

My story

My story

- started this PhD program in 2014

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology
- deep learning took over in the next few years

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology
- deep learning took over in the next few years
- subword models started to get popular in machine translation then in language modeling

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology
- deep learning took over in the next few years
- subword models started to get popular in machine translation then in language modeling
- so I shifted towards modeling and evaluation

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology
- deep learning took over in the next few years
- subword models started to get popular in machine translation then in language modeling
- so I shifted towards modeling and evaluation

1. Part I. deals with deep learning for morphology (2018–2020)

My story

- started this PhD program in 2014
- original topic: Unsupervised Learning of Morphology
- deep learning took over in the next few years
- subword models started to get popular in machine translation then in language modeling
- so I shifted towards modeling and evaluation

1. Part I. deals with deep learning for morphology (2018–2020)
2. Part II. is about evaluating language models with special focus on morphosyntax (2019–2024)

Outline

Part I.
Deep Learning for Morphology

Encoder-decoder models for morphology

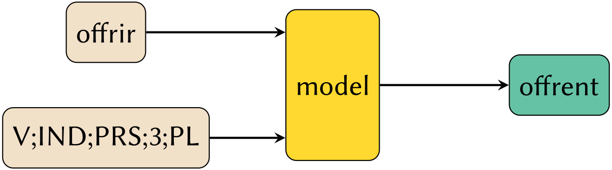
Encoder-decoder models for morphology

Thesis 1

Encoder-decoder (a.k.a. sequence-to-sequence or seq2seq) models are well-suited for morphological inflection and generation. This holds for type-level and sentence-level tasks in multiple languages.

These contributions were published in Ács (2018).

Morphological inflection



release	V;V.PTCP;PRS	releasing
deodourize	V;NFIN	deodourize
outdance	V;V.PTCP;PRS	outdancing
misrepute	V;NFIN	misrepute
vanquish	V;PST	vanquished
resterilize	V;3;SG;PRS	resterilizes

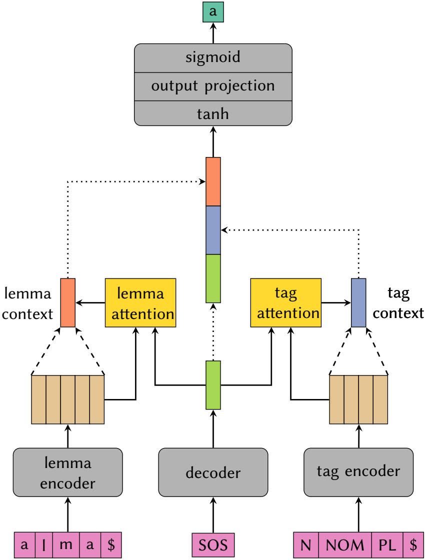
SIGMORPHON 2018 Shared tasks

Overview

- yearly competition computational morphology
- 2 tasks in 2018:
  1. Task 1: Type-level inflection
     - 110 languages
     - high (10,000), medium (1,000), low (100) data sizes
     - source: Wiktionary inflection tables
     - UniMorph schema (Kirov et al., 2018)
  2. Task 2: Inflection in context
     - 7 languages
- I participated as an individual team
- 3rd place in Task 1, 2nd place in Task 2

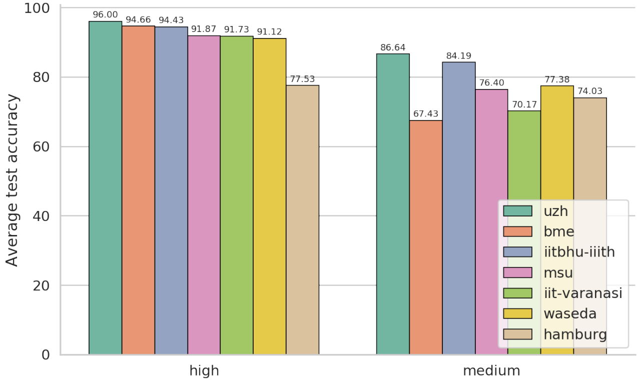
SIGMORPHON 2018 Shared tasks

My model for Task 1: Type-level inflection



SIGMORPHON 2018 Shared tasks

Task 1 results



Our team in orange (bme).

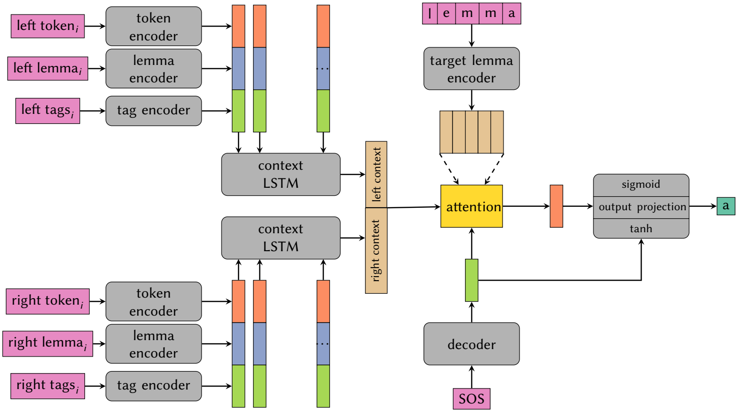
SIGMORPHON 2018 Shared tasks

Task 2: Inflection in context

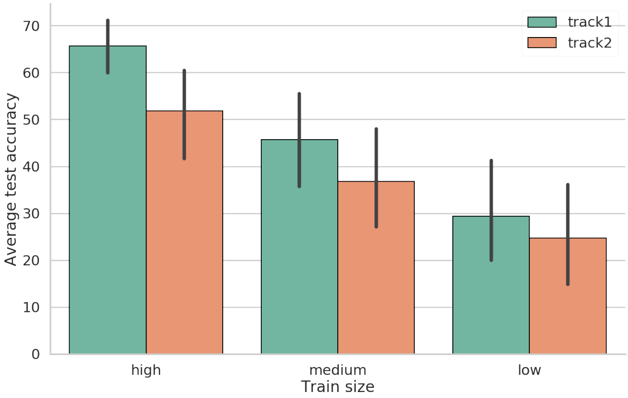
Les	le	DET;DEF;FEM;PL
compagnies	compagnie	N;FEM;PL
aériennes	aérien	ADJ;FEM;PL
à	à	ADP
bas	bas	ADJ;MASC;SG
coût	coût	N;MASC;SG
ne	ne	ADV;NEG
_	connaître	_
pas	pas	ADV;NEG
la	le	DET;DEF;FEM;SG
crise	crise	N;FEM;SG

Track 2: no lemmas or tags

Inflection in context model



Task 2 results



Neural pattern matching

Differentiable neural pattern matching for morphology

Thesis 2

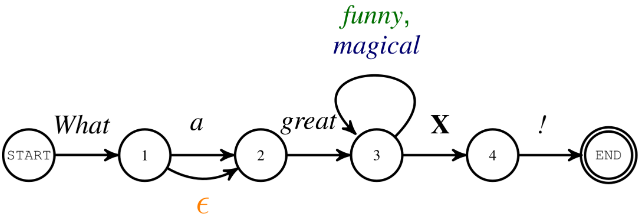
Differentiable neural pattern matching can extract morphosyntactic patterns in multiple languages when used as an encoder for morphological inflection and analysis.

Ács and Kornai (2020) was awarded the best paper award at the Hungarian Computational Linguistics Conference in 2020.

Neural pattern matching

Overview

- Schwartz et al. (2018) introduced SoPa or Soft Patterns, a differentiable pattern learner
- restricted to fixed length linear patterns with epsilon transitions and self-loops
- fully differentiable and end-to-end trainable
- they used it for sequence classification in English, token based



Neural pattern matching

My additions

- I reimplemented it as an encoder of an encoder-decoder model
- the decoder is an LSTM initialized with the final state of the SoPa encoder
- applied it at the character level
- each pattern matches a character span or subword

Neural pattern matching

Tasks

Language	Task	Source	Target
Hungarian	analysis	vásároljanak	V SBJV PRS INDF 3 PL
Hungarian	analysis	lepkékben	N IN+ESS PL
English	analysis	hugging	V V.PTCP PRS
French	analysis	désinstalleriez	V COND 2 PL
Hungarian	lemmatization	vásároljanak	vásárol
Hungarian	lemmatization	lepkékben	lepke
English	lemmatization	hugging	hug
French	lemmatization	désinstalleriez	désinstaller
Hungarian	copy	vásároljanak	vásároljanak
Hungarian	copy	lepkékben	lepkékben
English	copy	hugging	hugging
French	copy	désinstalleriez	désinstalleriez

The source is the same in all three tasks.

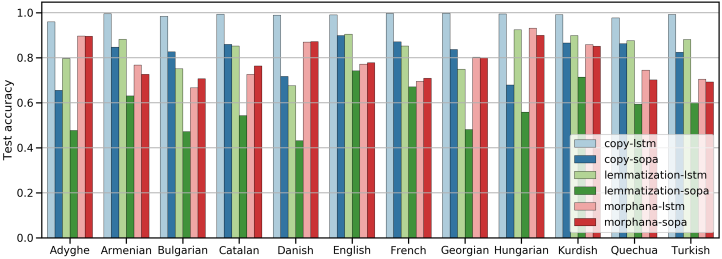
Neural pattern matching

Experimental setup

- 120 patterns: 40 3-long, 40 4-long, 40 5-long
- 12 typologically diverse languages
- 10,000 train, 2,000 dev, 2,000 test word types
- baseline: both the encoder and the decoder are LSTMs with attention
- SoPa seq2seq: SoPa encoder, LSTM decoder with attention on intermediate SoPa outputs

Neural pattern matching

Results



- the baseline is always better
- SoPa is not good at copying and lemmatization
- noticably better at morphological analysis

Neural pattern matching

Model similarity

- We define a similarity metric between two SoPa seq2seq models (M_1 and M_2) that work on the same input
- take the highest scoring T patterns for each input and compare the subwords
- for each pattern by M_1, find the most similar pattern in M_2
- average it over a dataset

$$\mathrm{Sim}(M_1, M_2, D) = \frac{1}{|D|} \sum_{d \in D} S(M_1(d), M_2(d))$$

$$S(M_1(d), M_2(d)) = \frac{1}{2T} \Big(\sum_{p_i \in P_1} \max_{p_j \in P_2} J(p_i, p_j) + \sum_{p_j \in P_2} \max_{p_i \in P_1} J(p_i, p_j) \Big)$$

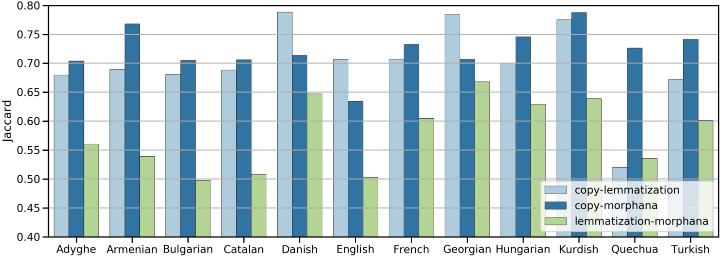
Neural pattern matching

Model similarity example

	^ablakban$	^ablakban$	^ablakban$	^ablakkban$	Max
^ablakban$	0	0.2	1	0.75	1
^ablakban$	0	0.5	0.5	0.75	0.75
^ablakban$	0	0.5	0	0.167	0.5
^ablakban$	0	0.75	0.167	0.33	0.75
Max	0	0.75	1	0.75	0.685

Neural pattern matching

Similarity results



Part II.
Evaluating Pre-trained Language Models

Morphosyntactic probing of PLMs

Morphosyntactic probing of PLMs

Thesis 3

Pre-trained language models (PLMs) trained on unannotated text learn morphology. PLMs' representations retain morphosyntactic information across a large set of typologically diverse languages and multiple tasks. This information can be recovered via probing or diagnostic classifiers.

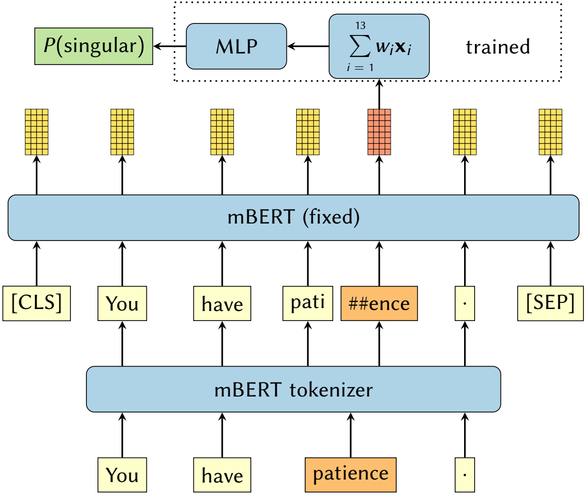
These contributions were published in (Ács, 2019; Ács et al., 2021; Acs et al., 2023).

Morphosyntactic probing of PLMs

Background

- Pre-trained Language Models or PLMs are probabilistic models of natural (written) language
- pre-trained on large unannotated text
- we mainly deal with masked language models
- contextual models
  - sentence representation (or longer)
  - word representation depends on the context
- BERT model family
- English and multilingual, later many language and domain specific
- evaluation by probing
  - take a set of annotated text
  - train a small classifier on top of the PLM's representation
  - if it performs well, the information is available in the model

Probing architecture



Morphosyntactic probing of PLMs

Universal Dependencies

Form	UPOS	Morphological features
The	DET	Definite=Def|PronType=Art
third	ADJ	Degree=Pos|NumType=Ord
was	AUX	Number=Sing|Person=3|Tense=Past|VerbForm=Fin
being	AUX	VerbForm=Ger
run	VERB	Tense=Past|VerbForm=Part|Voice=Pass
by	ADP	_
the	DET	Definite=Def|PronType=Art
head	NOUN	Number=Sing
of	ADP	_
an	DET	Definite=Ind|PronType=Art
investment	NOUN	Number=Sing
firm	NOUN	Number=Sing
.	PUNCT	_

Morphosyntactic probing dataset

Languages

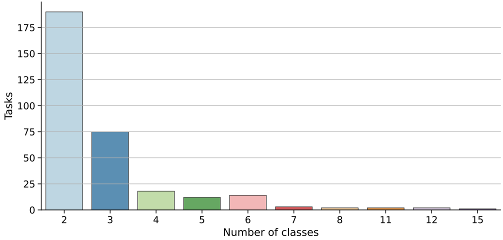
- UD: 122 languages
- mBERT: 104 languages
- XLM-RoBERTa: 100 languages
- intersection of these 3: 55 languages
- not enough morphosyntactic data: Chinese, Japanese, Vietnamese
- different tagging schema: Korean
- insufficient data in some languages
- external treebank for Albanian, silver data for Hungarian
- 42 languages

Morphosyntactic probing dataset

Tags and POS

- UD has over 130 different morphosyntactic tags
- most are only used for one or a few languages
- we pick 4 common tags: case, gender, number, tense
- 4 open POS classes: adj, noun, propn, verb
- 14 combinations are available
  - ⟨NOUN, Tense⟩ and ⟨PROPN, Tense⟩ are linguistically implausible
  - ⟨ADJ, Tense⟩ only in Estonian
- most common tasks are ⟨NOUN, Number⟩ (37 languages), ⟨NOUN, Gender⟩ (32) and ⟨VERB, Number⟩ (27)

Class number distribution

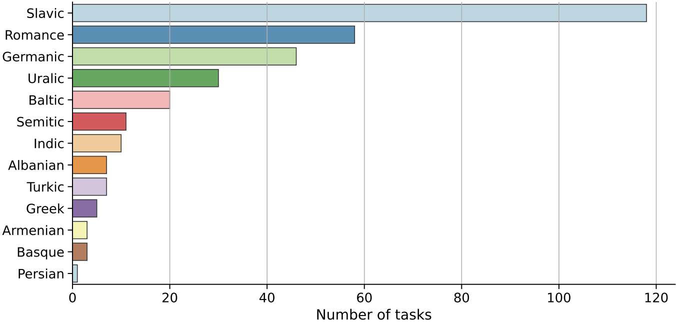


Most classes:

- ⟨Hungarian, NOUN, Case⟩: 18
- ⟨Estonian, NOUN, Case⟩: 15
- ⟨Finnish, NOUN, Case⟩: 12[1]
- ⟨Finnish, VERB, Case⟩: 12

[1] Infrequent classes were omitted.

Probing tasks by language family



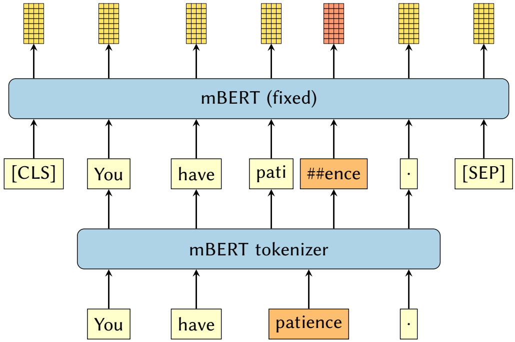
Number of morphological probing tasks by language family.

Probing dataset statistics

- 247 tasks
- 42 languages
- 10 language families (Indo-European subfamilies)
- 4 POS, 4 tags, 14 POS-tag combinations
- 2,000 train, 200 validaion and 200 test samples
- sentence length between 3 and 40 tokens, average 20.5

Subword pooling

Subword tokenization



- PLMs use subword tokenizers
- one token corresponds to multiple subwords, which one should we use?
- question for tagging problems too: POS and NER

Subword pooling

What do tokenizers do?

	mBERT		XLM-RoBERTa		
	count	2+	count	2+	_start
Arabic	1.95	48.9	1.49	35.0	3.4
Chinese	1.58	53.5	2.13	88.5	86.6
Czech	2.04	53.0	1.7	45.2	1.6
English	1.25	14.3	1.25	16.9	0.8
Finnish	2.32	67.3	1.86	53.0	2.3
French	1.34	22.4	1.41	28.7	2.1
German	1.64	30.6	1.57	29.7	1.3
Japanese	1.6	43.0	2.25	94.6	92.9
Korean	2.44	75.7	2.16	67.3	9.0

Subword pooling

Experimental setup

- 9 typologically diverse languages
- 3 tasks: morphosyntactic probing, POS, NER
- 9 subword pooling methods
- mBERT and XLM-RoBERTa
- feature extraction, no fine-tuning

Method	Explanation
FIRST	first subword unit
LAST	last subword unit
LAST2	concatenation of the last two subword units
F+L	$wu_{\text{first}} + (1 - w)u_{\text{last}}$
SUM	elementwise sum
MAX	elementwise max
AVG	elementwise average
ATTN	Attention over the subwords, weights generated by an MLP
LSTM	biLSTM reads all vectors, final hidden state

Subword pooling

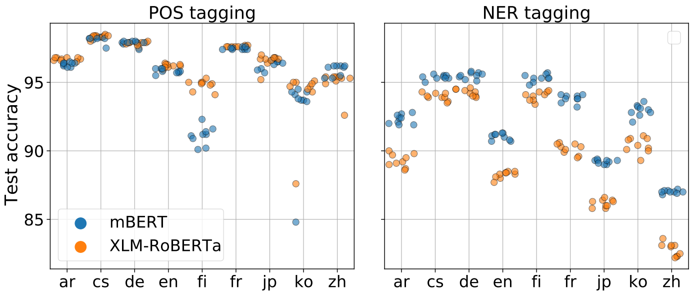
Main results

- Morphology
  - ATTN is the best pooling strategy but its advantage over LAST is small and often not significant
  - FIRST is the worst

Subword pooling

Main results

- Morphology
  - ATTN is the best pooling strategy but its advantage over LAST is small and often not significant
  - FIRST is the worst
- POS and NER
  - depends on the language
  - we recommend trying a simple strategy (LAST) and a parametric one (ATTN or LSTM)



Subword pooling

Conclusion

- most common: use first/last, max pooling
- the differences are very small
- last is usually better than first
- we pick either the first or the last based on the development data in all following experiments
  - last is better in over 90% of the tasks

Morphology in PLMs

Overview

- Do PLMs learn morphology?
- We use our 247 probing tasks
- We compare it against various baselines
- Extensive ablations

Morphology in PLMs

Models and baselines

main focus: mBERT and XLM-RoBERTa

other multilingual PLMs: XLM-Large, XLM-MLM-100, distilmBERT, mT5

main baseline: character LSTM (chLSTM) over full sentence, not pre-trained

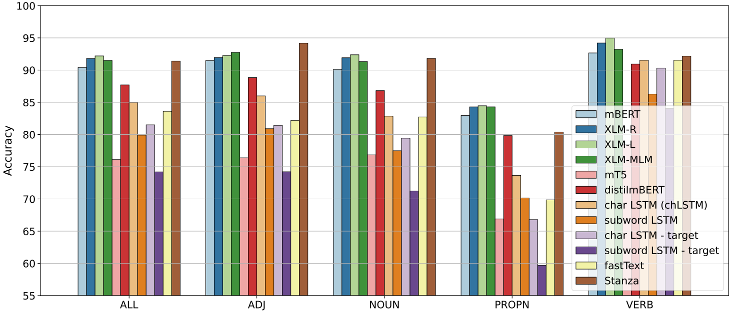
other baselines: subword LSTM on sentence, char LSTM and subword LSTM on target word only

fastText: language-specific bag-of-ngrams word vectors

Stanza: linguistic analysis toolkit for 70+ languages trained on UD

Morphology in PLMs

General results

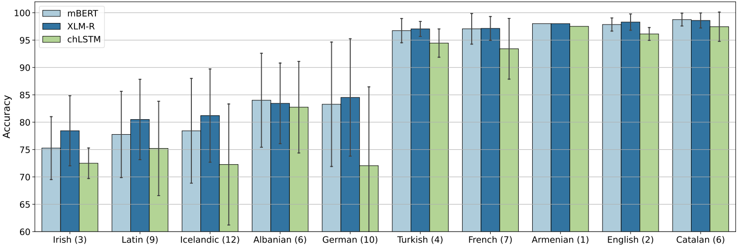


Accuracy of the pre-trained and the baseline models grouped by POS.

- XLM-RoBERTa is slightly better than mBERT, larger models are even better
- chLSTM is the best baseline, it's closest in verbal tasks

Morphology in PLMs

Easiest and hardest languages

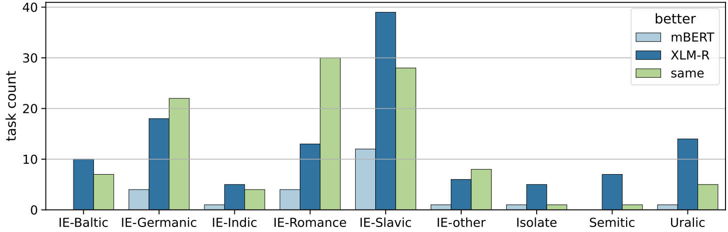


The best and the worst 5 languages by the average performance of mBERT and XLM-RoBERTa. The number of tasks in a particular language is listed in parentheses.

Morphology in PLMs

Model comparison

Which model is better[2] at how many tasks?



mBERT XLM-RoBERTa comparison by language family.

[2] Independent t-test over 10 runs.

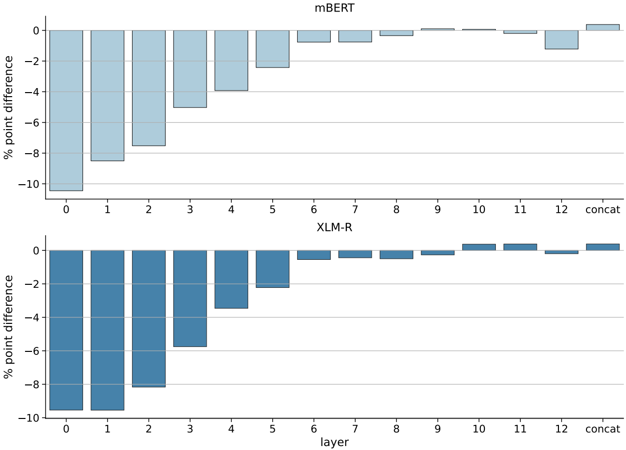
Ablations

Overview

- Probing has its fair share of criticism (Belinkov, 2021; Ravichander et al., 2021)
- We run various ablations to address them
- We find that:
  1. the choice of probe (linear, MLP, 2 layers) doesn't matter
  2. probing individual layers is no better or worse than probing the weighted sum of all layers
  3. fine-tuning is actually harmful (and wasteful)
  4. randomly initialized models (Voita and Titov, 2020) are much worse

Ablations

Pooling individual layers



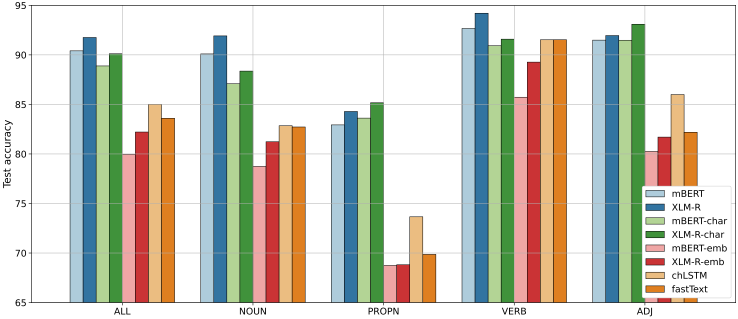
The difference between probing a single layer and probing the weighted sum of layers. *concat* is the concatenation of all layers. *0* is the embedding layer.

Ablations

Model ablations

mBERT-char: use character tokenization instead of subword tokenization

mBERT-emb: probe the embedding layer instead of the weighted sum of all layers



Language-specific models

Language-specific models

Thesis 4

Monolingual PLMs are better in their respective languages than multilingual PLMs but the difference is small and often not statistically significant. Moreover both monolingual and multilingual PLMs can be successfully transfered to new languages as long as the new language uses the same writing system.

These contributions were published in (Ács et al., 2021b) and (Ács et al., 2021a).

Models for Hungarian

Morphosyntactic evaluation tasks

Morph tag	POS	#classes	Values
Case	noun	18	Abl, Acc, . . . , Ter, Tra
Degree	adj	3	Cmp, Pos, Sup
Mood	verb	4	Cnd, Imp, Ind, Pot
Number[psor]	noun	2	Sing, Plur
Number	adj	2	Sing, Plur
Number	noun	2	Sing, Plur
Number	verb	2	Sing, Plur
Person[psor]	noun	3	1, 2, 3
Person	verb	3	1, 2, 3
Tense	verb	2	Pres, Past
VerbForm	verb	2	Inf, Fin

Models for Hungarian

Sequence tagging tasks

Part-of-speech tagging

1. Szeged UD Treebank (Farkas et al., 2012)
   - gold standard automatically converted to UD
   - 910/441/449 sentences
2. Webcorpus 2 subsample
   - tagged with emtsv (Indig et al., 2019)
   - 10,000/2,000/2,000 sentences

Named entity recognition

1. Szeged NER corpus
   - 8172/503/900 sentences

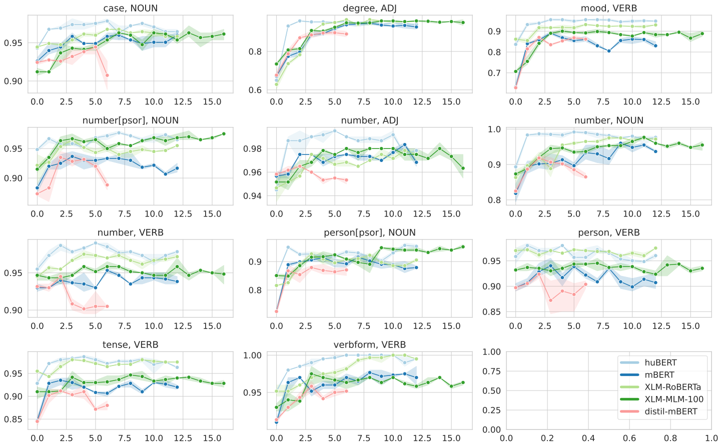
Models for Hungarian

Experimental setup

- same probing architecture for morphology
- similar setup for POS and NER
- no fine-tuning due to resource limitations
- huBERT: the only Hungarian model at the time
- multilingual models: mBERT, XLM-RoBERTa, XLM-MLM-100, distilmBERT

Models for Hungarian

Morphology results by Tranformer layer

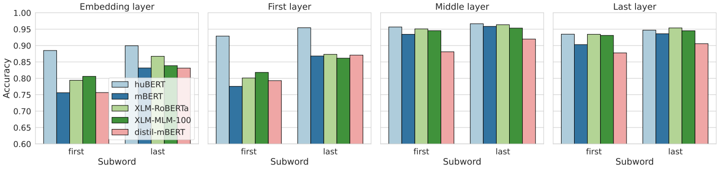


The layerwise accuracy of morphological probes using the last subword. Shaded areas represent confidence intervals over 3 runs.

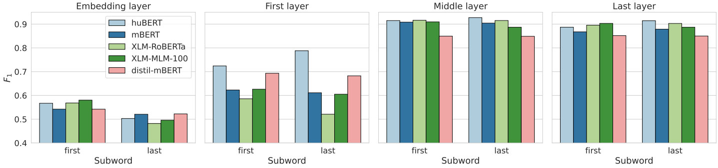
Models for Hungarian

POS and NER results

Szeged POS



Szeged NER



Models for Uralic languages

Overview

Experimental setup:

- same as the Hungarian evaluation
- fine-tuning
- include every Uralic language with data regardless of model support

Models:

language-specific: HuBERT, FinBERT, EstBERT, Russian BERT

multilingual: mBERT, XLM-RoBERTa

random mBERT: random weights, mBERT tokenizer

Models for Uralic languages

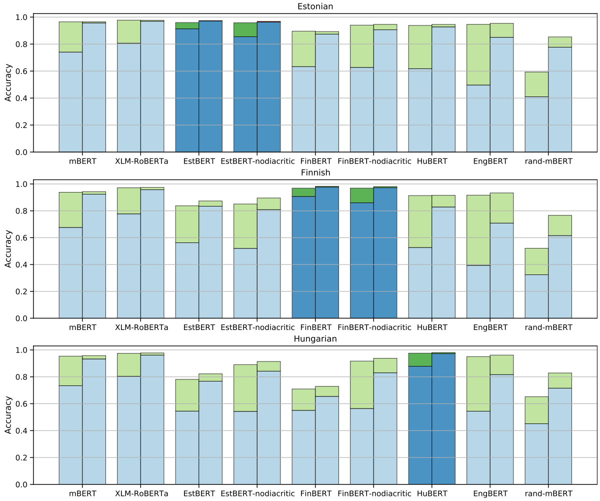
Languages and data

Language	Code	Morph	POS	NER
Hungarian	[hu]	26k	2000	2000
Finnish	[fi]	38k	2000	2000
Estonian	[et]	26k	2000	2000
Erzya	[myv]	0	1680	1800
Moksha	[mdf]	0	164	400
Karelian	[krl]	0	224	0
Livvi	[olo]	0	122	0
Komi Permyak	[koi]	0	78	2000
Komi Zyrian	[kpv]	0	562	1700
Northern Sami	[sme]	0	2000	1200
Skolt Sami	[sms]	0	101	0

Size of training data for each language.

Models for Uralic languages

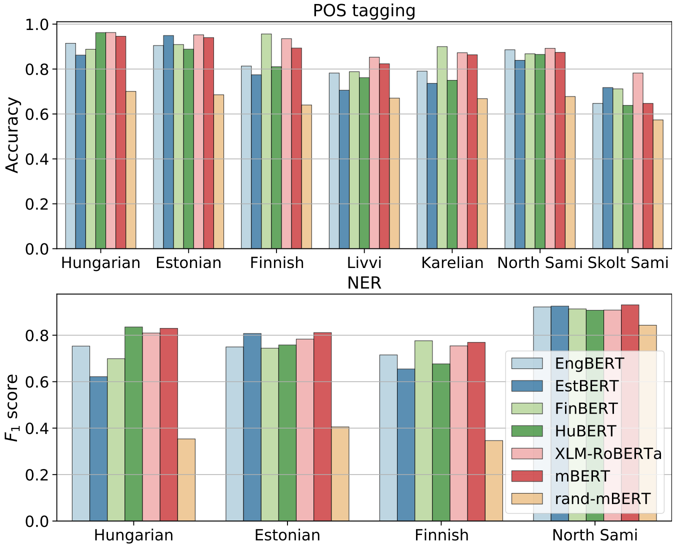
Morphology results



Pairs of bars: probing the first and last subword. Monolingual models are highlighted.

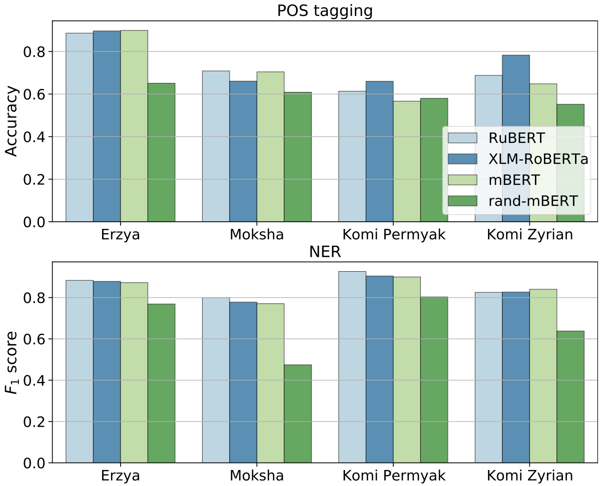
Models for Uralic languages

POS and NER results - Latin script



Models for Uralic languages

POS and NER results - Cyrillic script



Models for Uralic languages

Conclusions

- monolingual models are the best when available
- multilingal models are close
- model transfer is surprisingly good even for unsupported languages
- state-of-the-art POS and NER models for minority languages with no language-specific effort

Perturbations and Shapley values

Perturbations and Shapley values

Thesis 5

The source of morphosyntactic information is often localized in a sentence. The systemic removal of certain information (perturbations) reveals where the information is stored. The role of context in morphosyntax can be quantified via Shapley values and the results often comply with linguistic intuitions.

These contributions were published in (Acs et al., 2023).

Perturbations

- Perturbations are a systematic removal of information from the sentence.
- We retrain the probe on the perturbed sentence and quantify the change as:

$$\text{Effect}(m, t, p) = 1 - \frac{\text{Acc}(m, t, p)}{\text{Acc}(m, t)},$$

where m is the model, t is a probing task and p is a perturbation.

Perturbations

- Perturbations are a systematic removal of information from the sentence.
- We retrain the probe on the perturbed sentence and quantify the change as:

$$\text{Effect}(m, t, p) = 1 - \frac{\text{Acc}(m, t, p)}{\text{Acc}(m, t)},$$

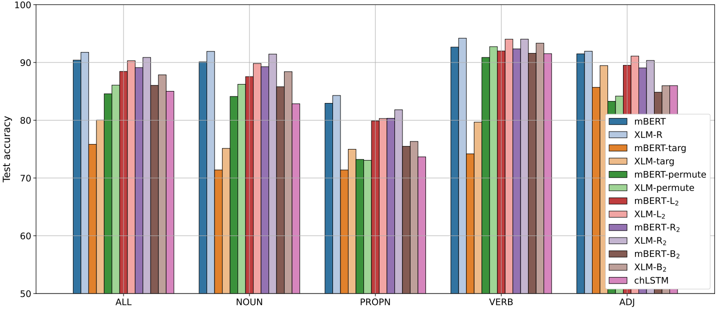
where m is the model, t is a probing task and p is a perturbation.

Perturbation	Explanation	Example
Original		Then he ripped open Hermione 's letter and **read** it out loud .
TARG	mask target word	Then he ripped open Hermione 's letter and **[M]** it out loud .
L2	mask previous 2 words	Then he ripped open Hermione 's [M] [M] **read** it out loud .
R2	mask next 2 words	Then he ripped open Hermione 's letter and **read** [M] [M] loud .
B2	mask 2 on each side	Then he ripped open Hermione 's [M] [M] **read** [M] [M] loud .
PERMUTE	shuffle word order	and open **read** Then letter . it out he ripped 's Hermione loud

List of perturbation methods with examples. The target word is in **bold**. The mask symbol is abbreviated as [M].

Perturbations

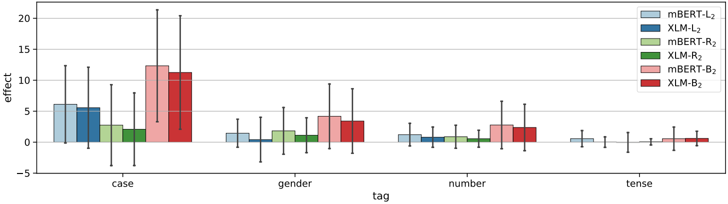
Perturbed accuracy



Test accuracy of the perturbed probes grouped by POS. The first group is the average of all 247 tasks. The first two bars in each group are the unperturbed probes' accuracy.

Perturbations

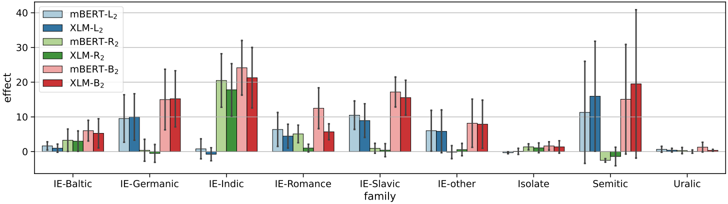
Context masking results



- case is the only tag affected strongly by context masking
- L2 is bigger than R2, the left context is more important

Perturbations

Context masking results

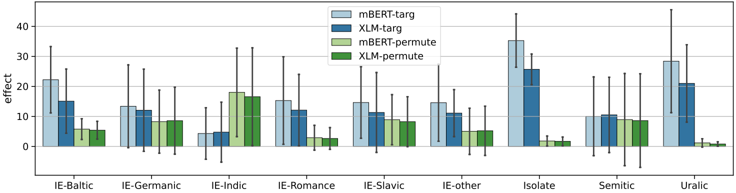


The effect of context masking on case tasks groupby by language family.

- std is larger than the effects
- L_2 is smaller than R_2 in Baltic and Indic languages
- context masking has neglible effect on Uralic languages

Perturbations

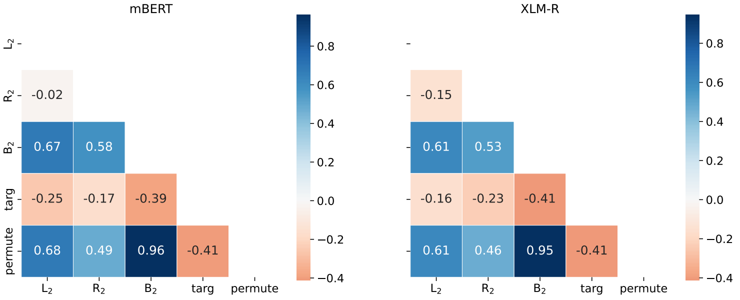
Target masking and permutation



- TARG has by far the largest effect
- TARG and PERMUTE have opposite effects?

Perturbations

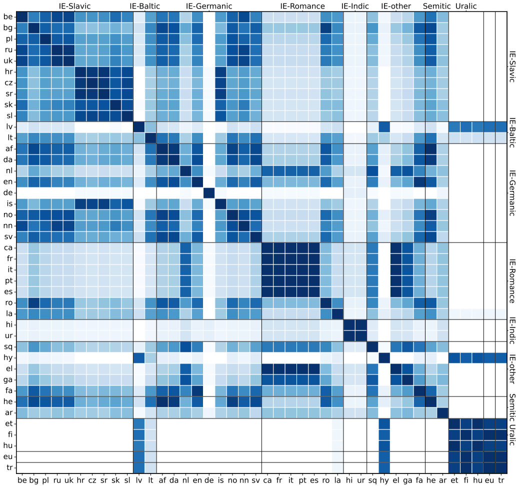
Relationship between perturbations



- TARG and PERMUTE indeed have a negative correlation
- PERMUTE and B_2 are almost identical in effect

Perturbations

Typology



Co-occurrence counts for each languages pair over 100 k-means clustering runs.

Shapley values

Formulation

Let's split the sentence into 9 parts or 9 players:

- T is the target word
- L_1 is the previous word, R_1 is the next word
- and so on:

$$S = L_{4+}, L_3, L_2, L_1, T, R_1, R_2, R_3, R_{4+}$$

Each player's contribution can be quantified as:

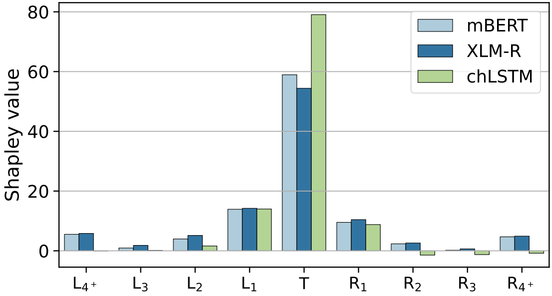
$$\varphi(i) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{n-1}{|S|}},$$

where the value function of a subset of players S is:

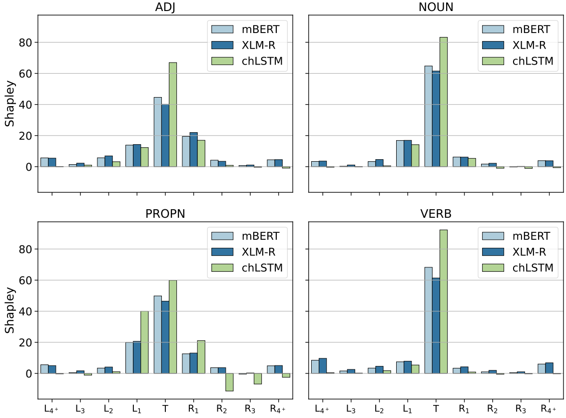
$$v(S) = 100 - 100 \cdot \frac{\mathrm{Acc}_S - \mathrm{Acc}_{\text{all masked}}}{\mathrm{Acc}_{\text{mBERT}} - \mathrm{Acc}_{\text{all masked}}}$$

Shapley values

Average Shapley values

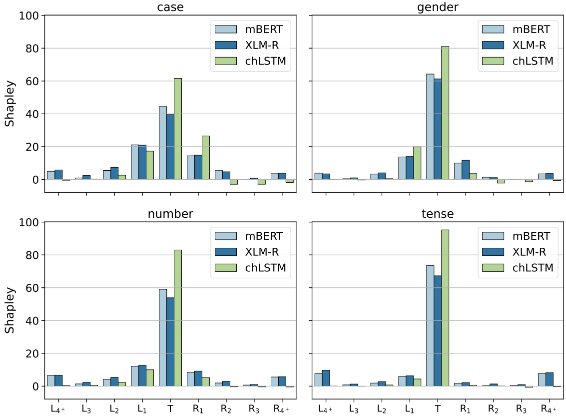


Shapley values
Values by POS

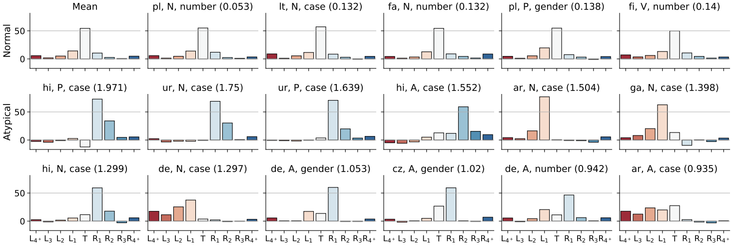


Shapley values

Values by tag



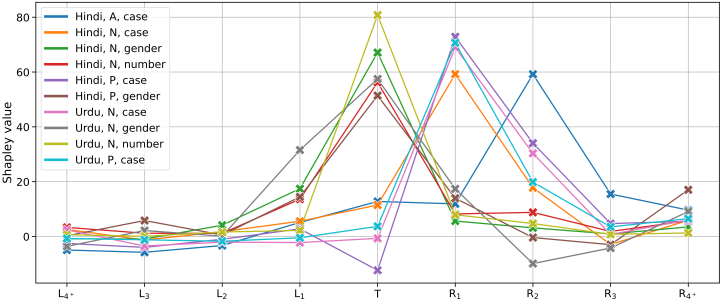
Shapley values
Outliers



Least and most anomalous Shapley distributions. The first row are the mean Shapley values of the 247 tasks and the 5 tasks *closest* to the mean distribution, i.e. the least anomalous as measured by the dfm distance from the average Shapley values. The rest of the rows are the most anomalous Shapley values in descending order. For each particular task, its distance from the mean (dfm) is listed in parentheses above the graphs.

Shapley values

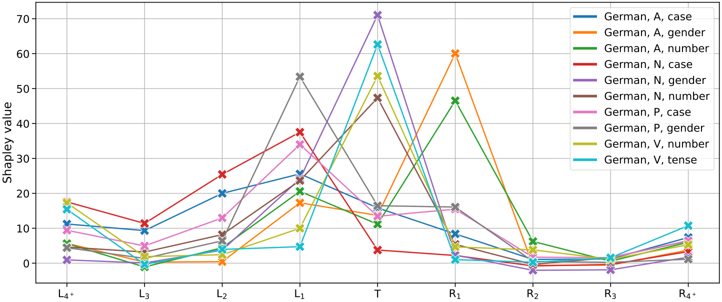
Hindi and Urdu tasks



Shapley values in Indic tasks.

Shapley values

German tasks



Shapley values in German tasks.

Statistics

- Number of experiments: appr. 500,000
  - unperturbed and perturbed experiments run 10 times
  - Shapley computation is exponential, 460,000 experiments
  - 40 days of runtime
- Maximum 200 epochs. Early stopping in 98% of the time
- Average 22 epochs
- 43 tables, 63 figures, 117 references for now

References I

Judit Ács. 2018. BME-HAS system for CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 121–126.

Judit Ács. 2019. Exploring BERT's vocabulary. http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html. Accessed: 2021-05-14.

Judit Acs, Endre Hamerlik, Roy Schwartz, Noah A. Smith, and Andras Kornai. 2023. Morphosyntactic probing of multilingual BERT models. *Natural Language Engineering*, page 1–40.

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

References II

Judit Ács and András Kornai. 2020. The role of interpretable patterns in deep learning for morphology.

Judit Ács, Dániel Lévai, and András Kornai. 2021a. Evaluating transferability of BERT models on Uralic languages. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*. Association for Computational Linguistics.

Judit Ács, Dániel Lévai, Dávid Márk Nemeskey, and András Kornai. 2021b. Evaluating contextualized language models for Hungarian. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2020)*, Szeged.

Yonatan Belinkov. 2021. Probing Classifiers: Promises, Shortcomings, and Advances. *arXiv:2102.12452 [cs]*. ArXiv: 2102.12452.

References III

Richárd Farkas, Veronika Vincze, and Helmut Schmid. 2012. Dependency parsing of Hungarian: Baseline results and challenges. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 55–65, Stroudsburg, PA, USA. Association for Computational Linguistics.

Balázs Indig, Bálint Sass, Eszter Simon, Iván Mittelholcz, Péter Kundráth, and Noémi Vadász. 2019. emtsv – Egy formátum mind felett [emtsv – One format to rule them all]. In *XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019)*, pages 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport.

References IV

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian Mielke, Arya D. McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. Unimorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Dávid Márk Nemeskey. 2021. Introducing `huBERT`. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021)*, pages 3–14, Szeged.

Joakim Nivre, Mitchell Abrams, Željko Agić, et al. 2018. Universal Dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

References V

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Roy Schwartz, Sam Thomson, and Noah A. Smith. 2018. SoPa: Bridging CNNs, RNNs, and Weighted Finite-State Machines. In *Proc. 56th ACL Annual Meeting*, pages 295–305, Melbourne, Australia.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.