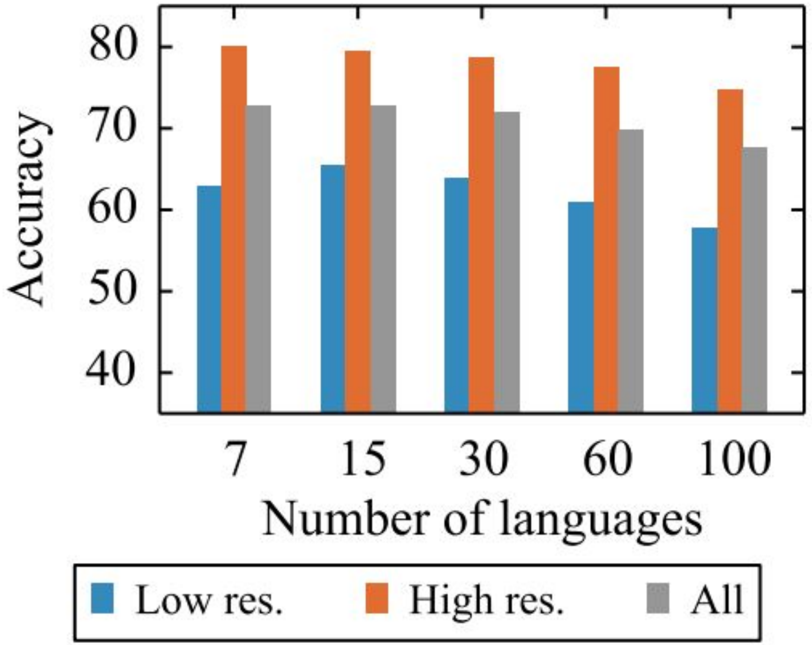
UDapter: Language Adaptation for Truly Universal Dependency Parsing

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, Gertjan van Noord

university of groningen

Curse of Multilinguality: Transfer-Interference Trade-Off

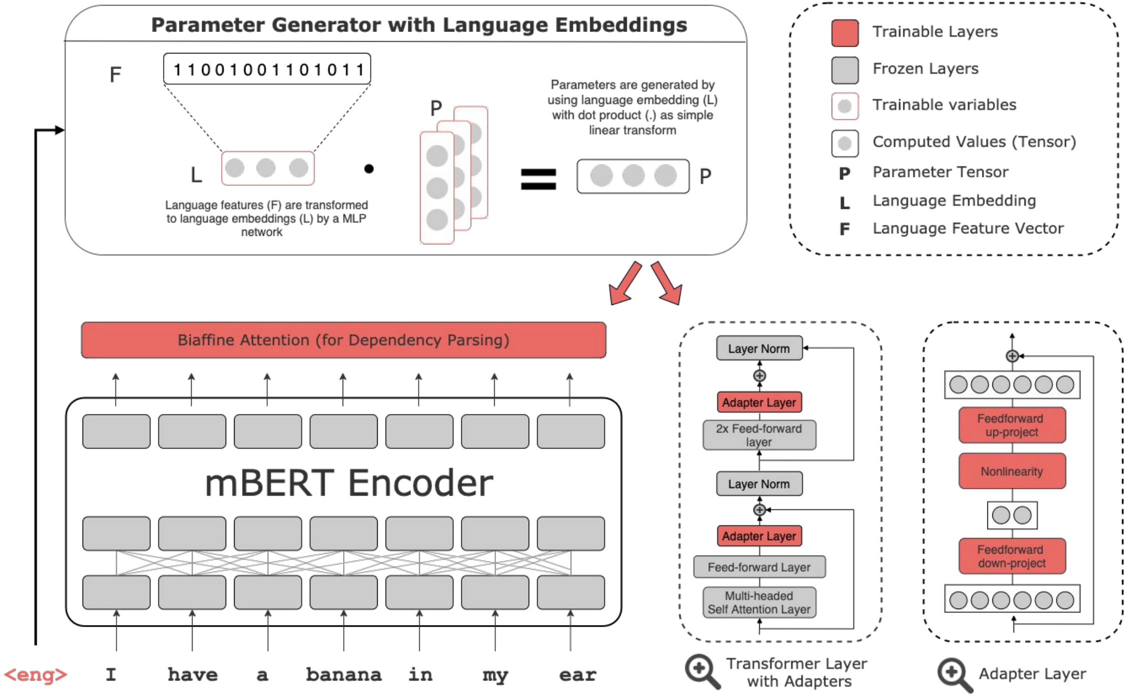


Our solution:

- Learn to adapt parameters of multilingual model for each language instead of training separate modules
- Increase per-language capacity by adapters
- Conditioning the adaptation to language typology features (zero-shot transfer)

Figure is taken from
Unsupervised Cross-lingual Representation Learning at Scale (Conneu et. at., 2020)

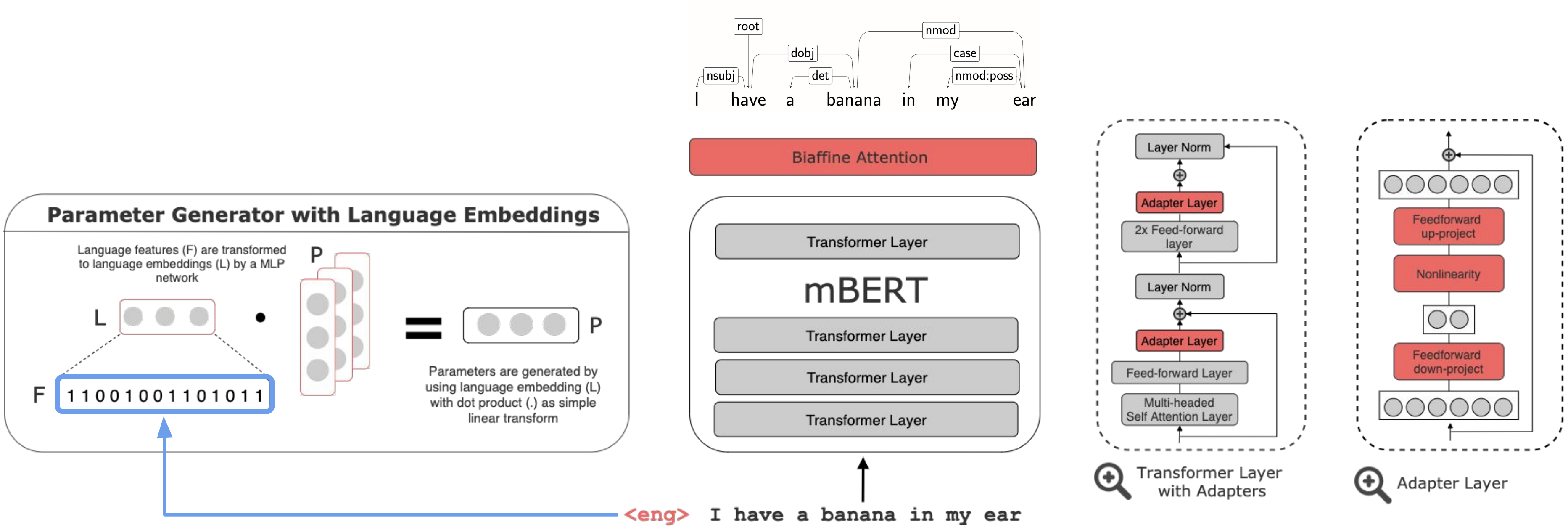
Multilingual Adaptation for All Languages



Novel multilingual adaptation:

- Combining language-specific and multilingual adaptation with contextul parameter generator approach based on mBERT
- Learning adapters via language embeddings
- Learning language embeddings from typological features

Multilingual Adaptation for All Languages

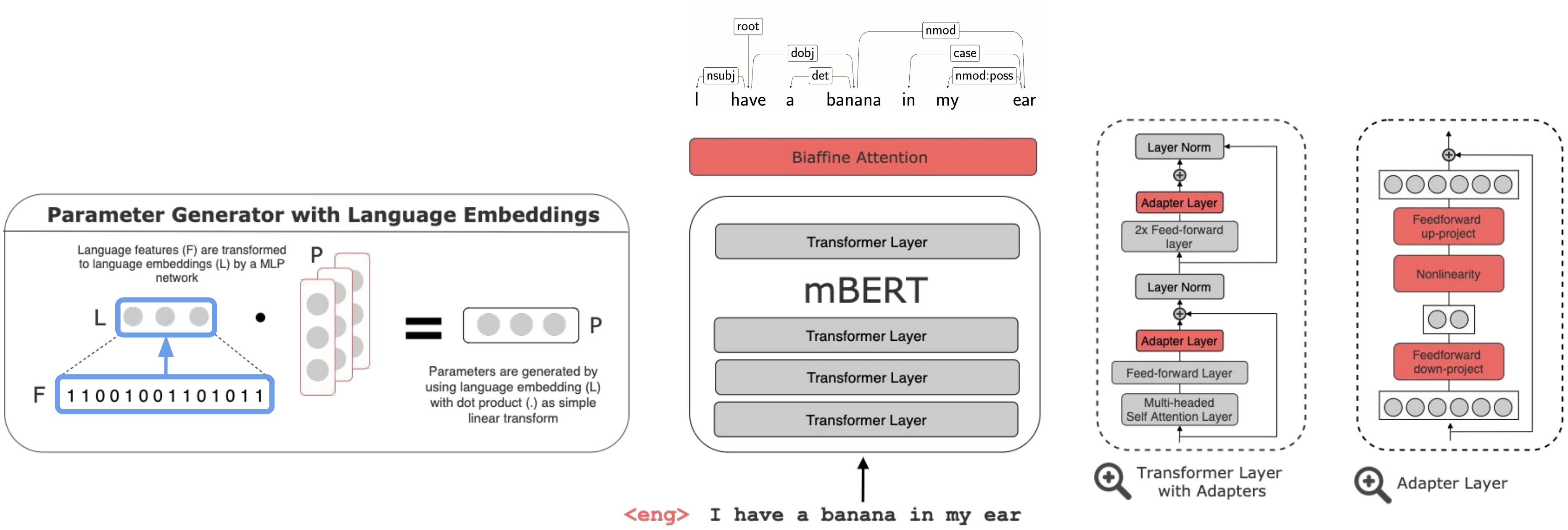


* ***Language-typology features*** *are* obtained from URIEL database for each language.

* ***All syntactic, phonological and inventory*** *features* are used (289 features in total).

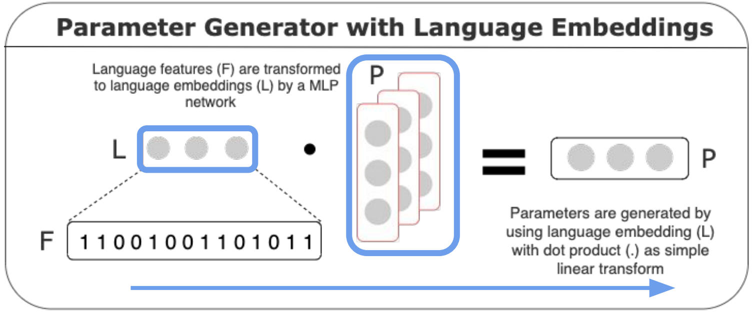
URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors (Littell et. at., 2017)

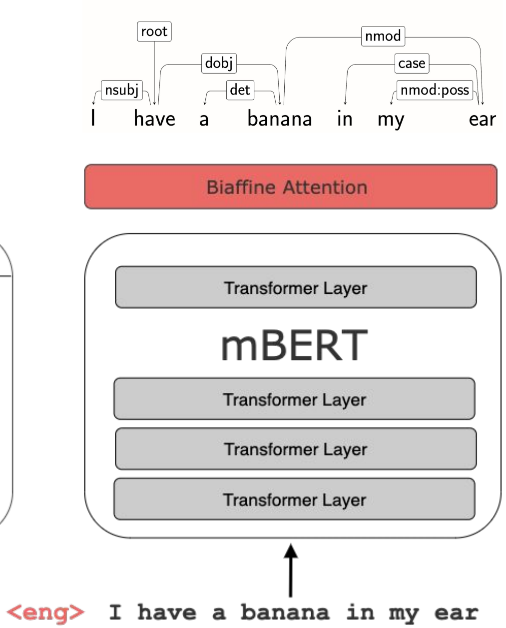
Multilingual Adaptation for All Languages

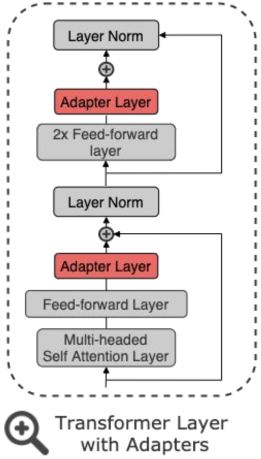


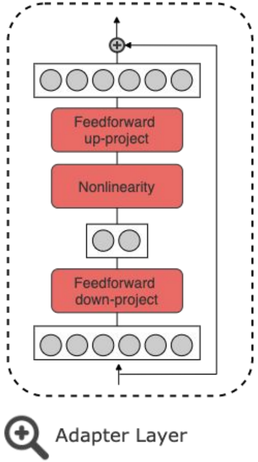
* ***Language embedding (L)*** is learned from ***typological feature vector (F)*** for each language
* ***A MLP*** is trained with the full model that is used to generate language embeddings for ***zero-shot languages in test time***

Multilingual Adaptation for All Languages







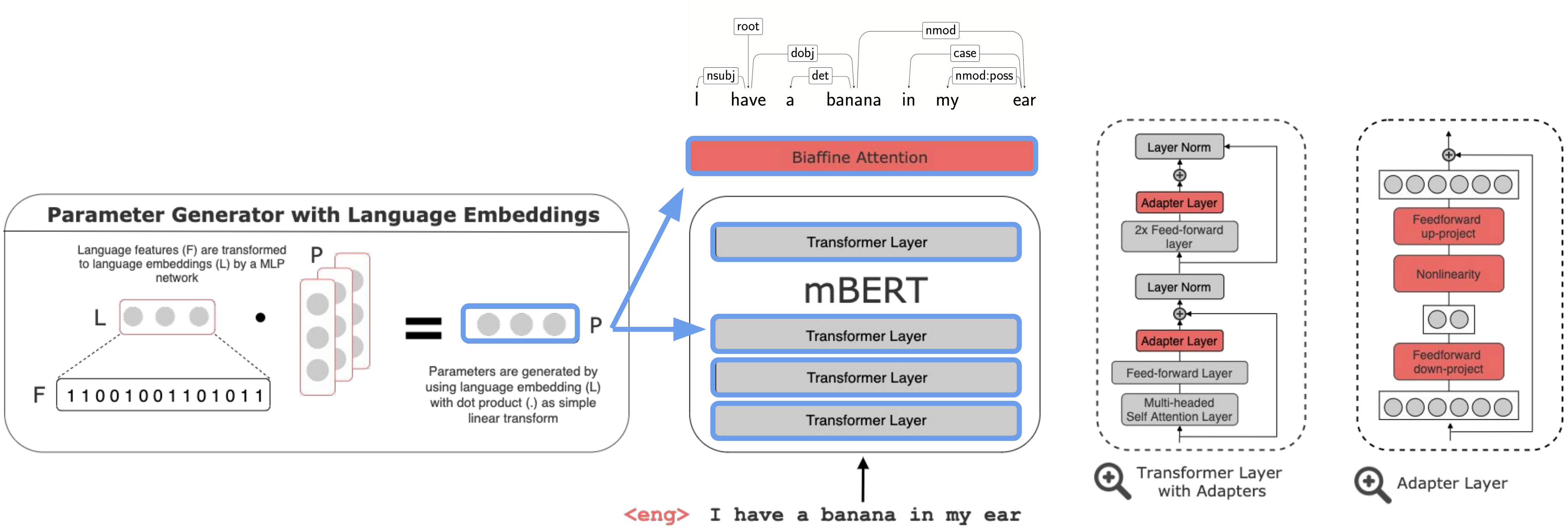


* ***Parameter Generator Network*** takes lang. embedding (L) and ***generates model parameters from shared parameter table.***

* Parameter generation is defined as ***simple linear transform*** (dot product)

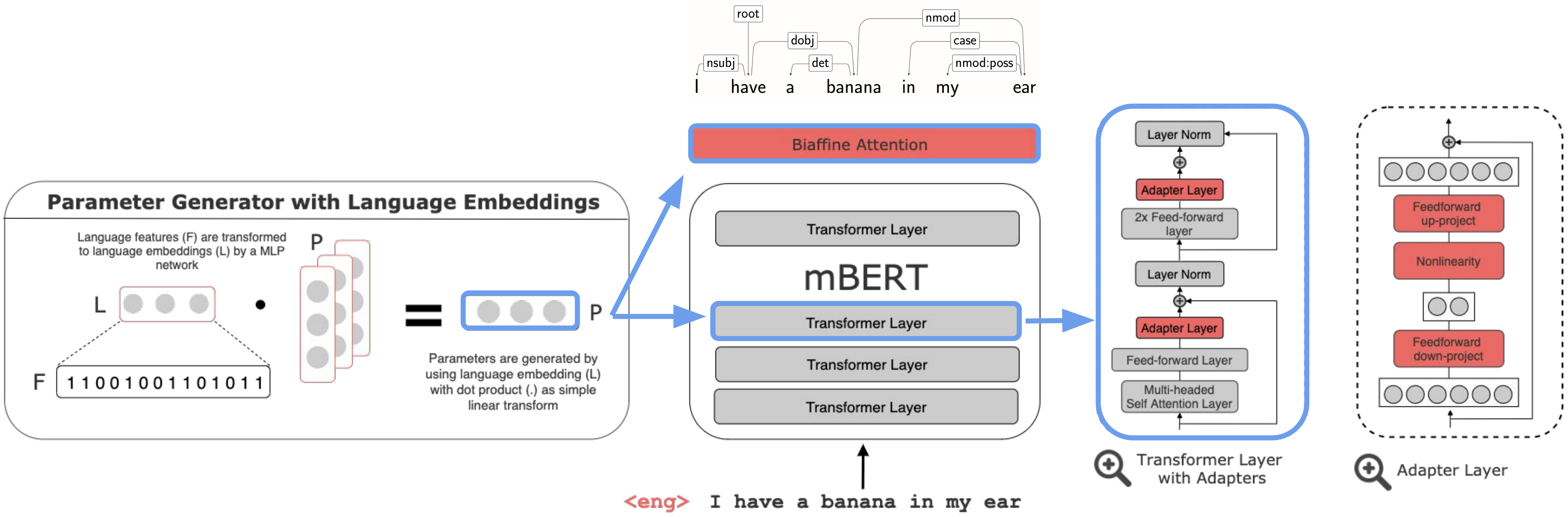
Contextual Parameter Generation for Universal Neural Machine Translation (Platanios et al.,2018)

Multilingual Adaptation for All Languages



* Only ***biaffine attention*** and ***adapters' parameters*** are generated/modified instead of full model by Parameter Generator

Multilingual Adaptation for All Languages

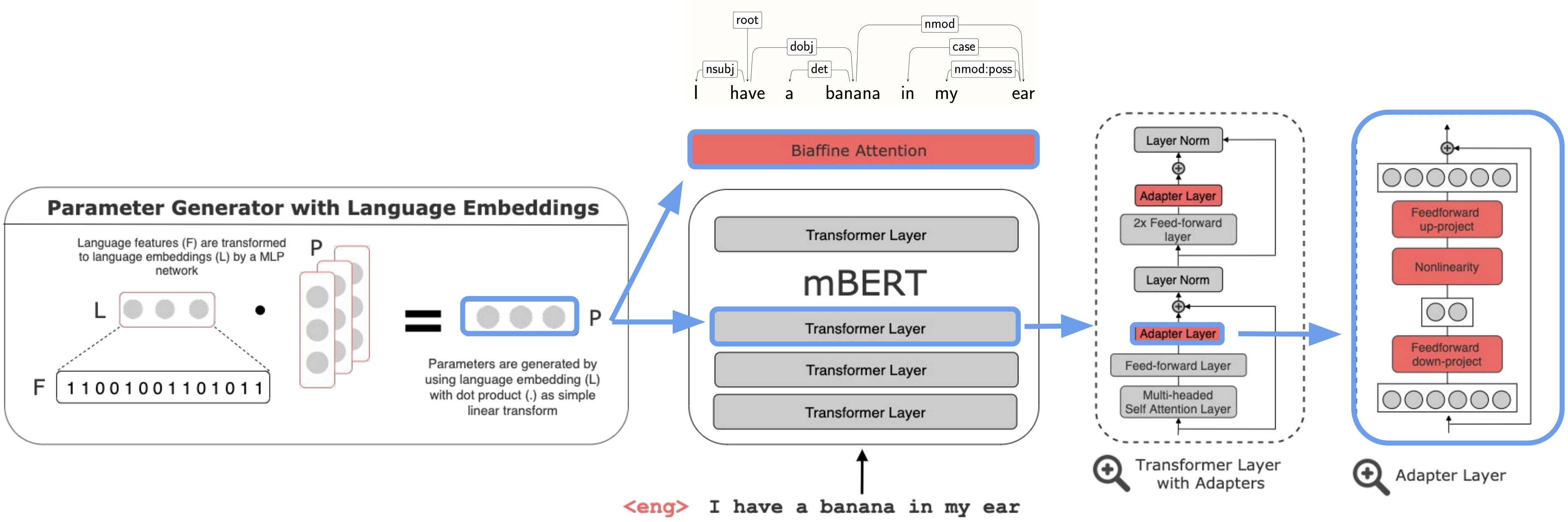


* ***Adapters*** *are* small bottleneck layers that are injected into BERT transformer layers.

* When ***fine-tuning with adapters***, original model's weights are not updated.

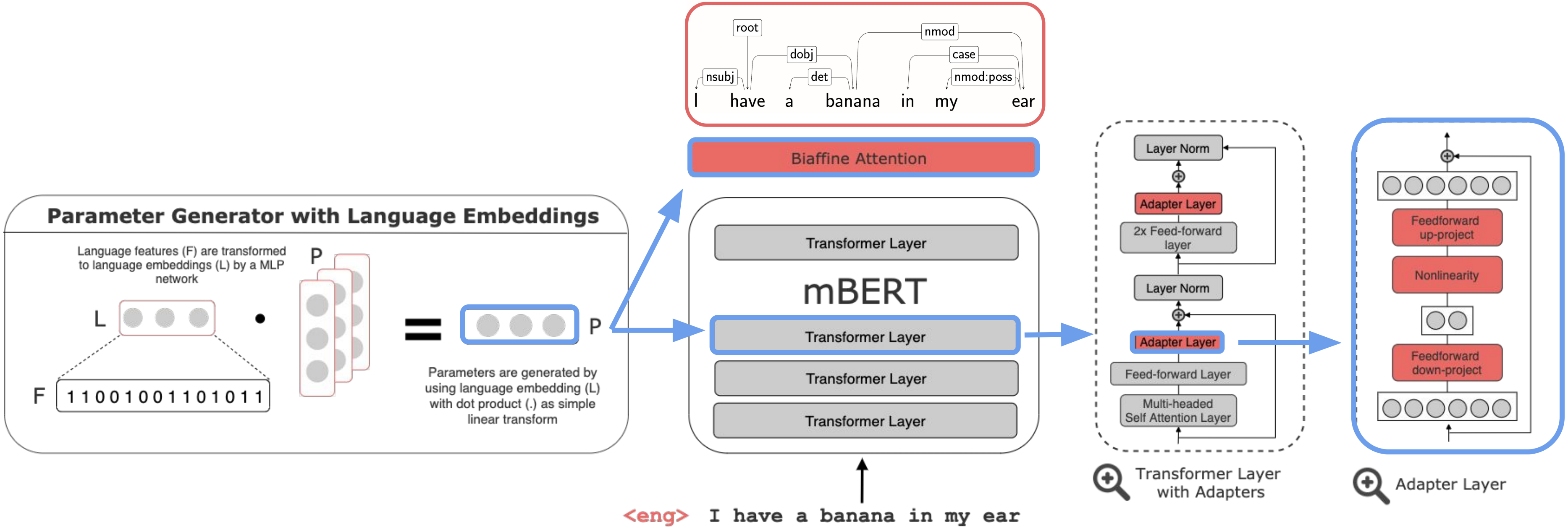
Parameter-Efficient Transfer Learning for NLP (Houlsby et al., 2019)

Multilingual Adaptation for All Languages



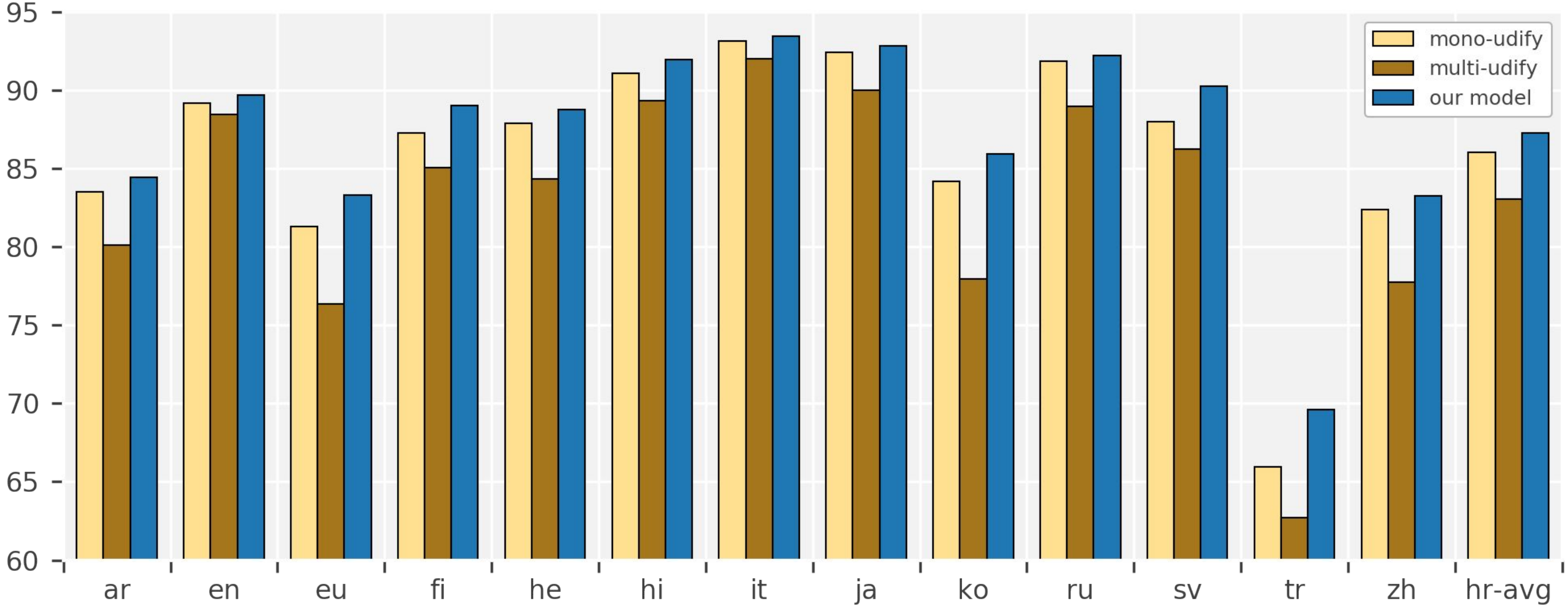
* ***Adapters*** only includes ***down- and up-projections with nonlinearity*** to fit in dimensionality of the pretrained model.

Multilingual Adaptation for All Languages



* ***Adapters*** only includes ***down- and up-projections with nonlinearity*** to fit in dimensionality of the pretrained model.

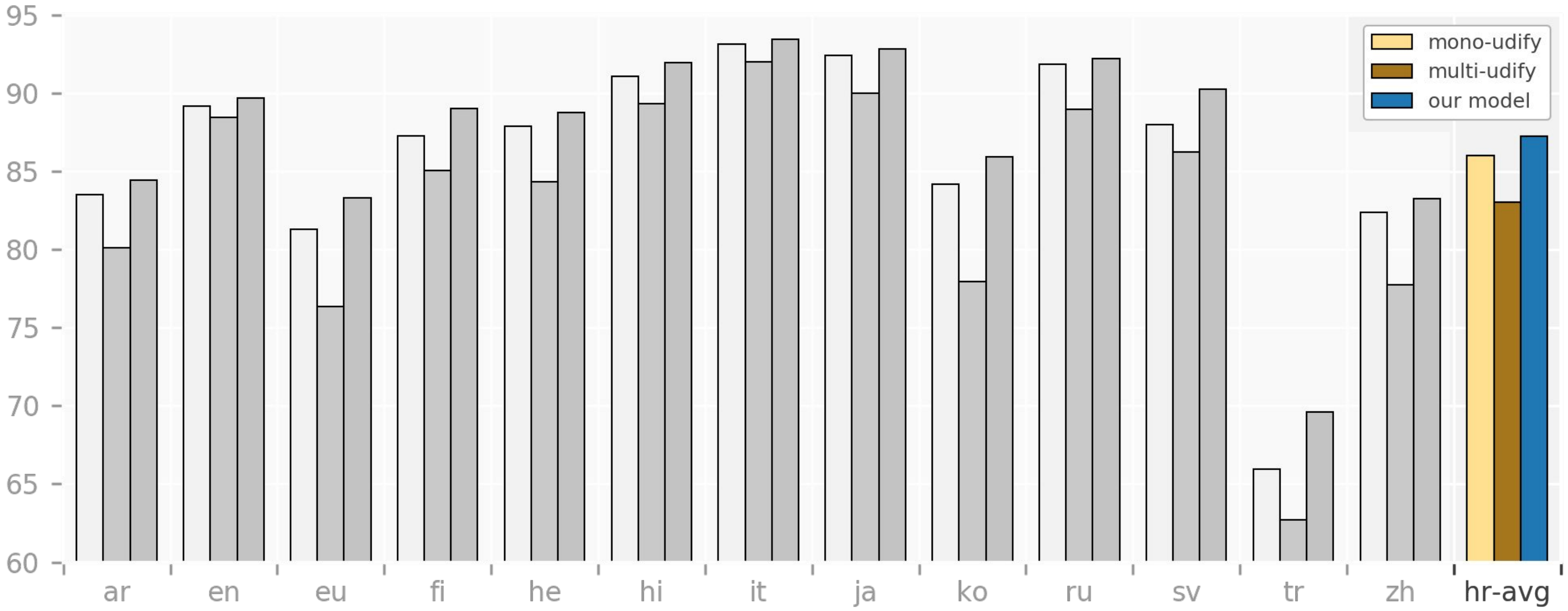
Results on High-Resource Languages



* **Training:** 13 *typologically diverse* (language-family, word-order, script) high-resource languages

UDify: 75 Languages, 1 Model: Parsing Universal Dependencies Universally (Kondratyuk et. at., 2019)

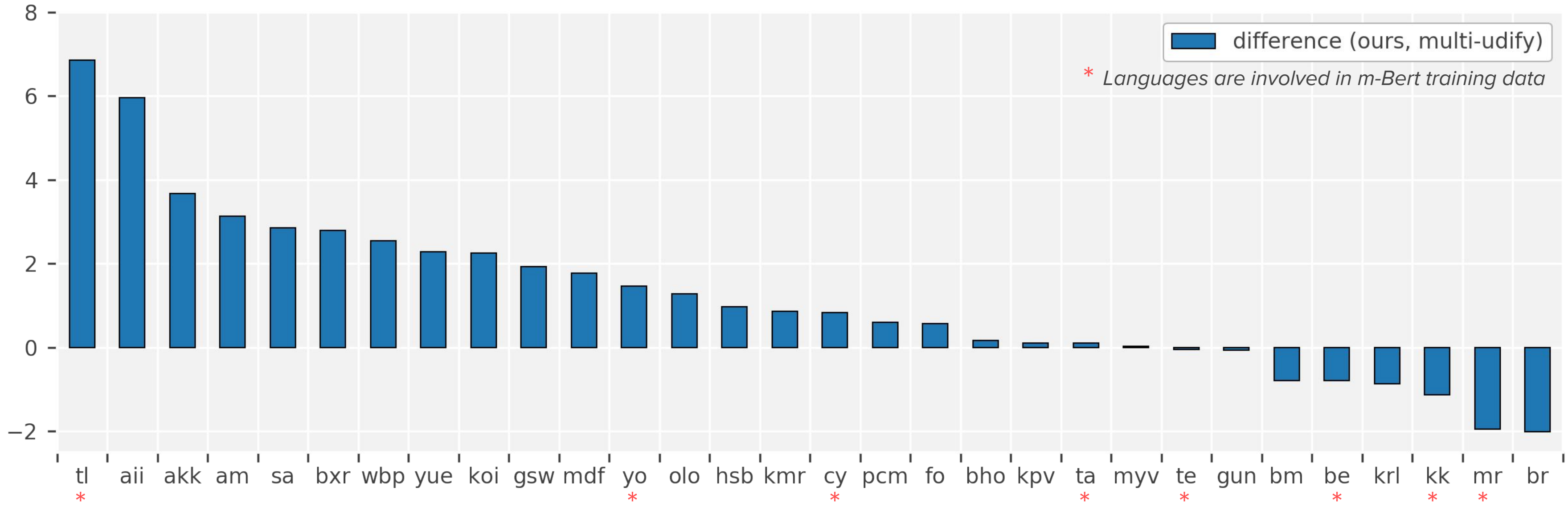
Results on High-Resource Languages



* **Training:** 13 *typologically diverse* (language-family, word-order, script) high-resource languages

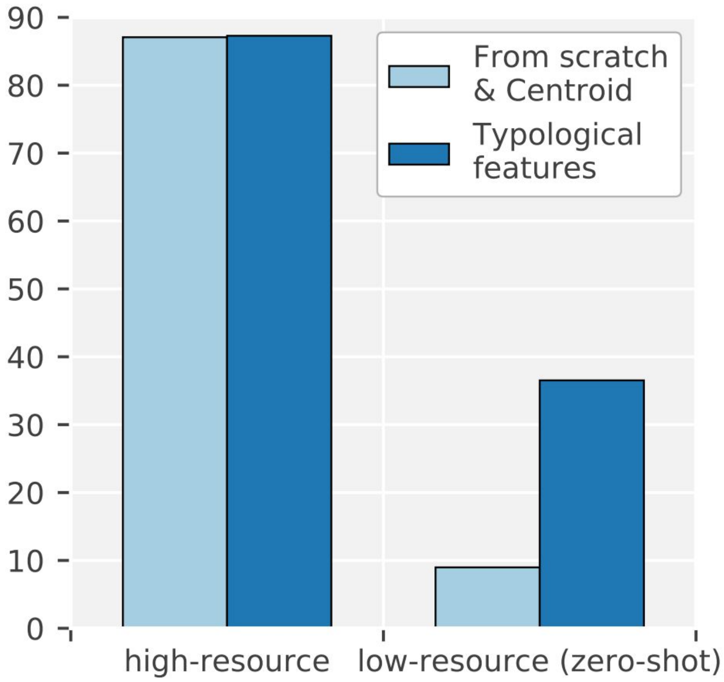
UDify: 75 Languages, 1 Model: Parsing Universal Dependencies Universally (Kondratyuk et. at., 2019)

Results on Low-Resource Languages



* **Zero-shot:** 30 genuinely low-resource languages from different language families

Impact of Language Typology Features



Lang. Embeddings with or w/o Typology Features:

- Training languages can have good adaptation w/o typology features
- For zero-shot transfer, typological features are crucial for UDapter.

Check the paper for more analysis

https://ahmetustun.github.io/udapter

Paper, Code and Slides

Experiments and Data

Language	Code	Treebank	Family	Word Order	Train	Test
Arabic	ar	PADT	Afro-Asiatic, Semitic	VSO	6.1k	680
Basque	eu	BDT	Basque	SOV	5.4k	1799
Chinese	zh	GSD	Sino-Tibetan	SVO	4.0k	500
English	en	EWT	IE, Germanic	SVO	12.5k	2077
Finnish	fi	TDT	Uralic, Finnic	SVO	12.2k	1555
Hebrew	he	HTB	Afro-Asiatic, Semitic	SVO	5.2k	491
Hindi	hi	HDTB	IE, Indic	SOV	13.3k	1684
Italian	it	ISDT	IE, Romance	SVO	13.1k	482
Japanese	ja	GSD	Japanese	SOV	7.1k	551
Korean	ko	GSD	Korean	SOV	4.4k	989
Russian	ru	SynTagRus	IE, Slavic	SVO	15k*	6491
Swedish	sv	Talbanken	IE, Germanic	SVO	4.3k	1219
Turkish	tr	IMST	Turkic, Southwestern	SOV	3.7k	975

- **Training:** 13 *typologically diverse* (language-family, word-order, script) high-resource languages

- **Zero-shot:** 30 genuinely low-resource languages from different language families

Language	Code	Treebank(s)	Family	Test
Akkadian	akk	PISANDUB	Afro-Asiatic, Semitic	1074
Amharic	am	ATT	Afro-Asiatic, Semitic	101
Assyrian	aii	AS	Afro-Asiatic, Semitic	57
Bambara	bm	CRB	Mande	1026
Belarusian	be	HSE	IE, Slavic	253
Bhojpuri	bho	BHTB	IE, Indic	254
Breton	br	KEB	IE, Celtic	888
Buryat	bxr	BDT	Mongolic	908
Cantonese	yue	HK	Sino-Tibetan	1004
Erzya	myv	JR	Uralic, Mordvin	1550
Faroese	fo	OFT	IE, Germanic	1207
Karelian	krl	KKPP	Uralic, Finnic	228
Kazakh	kk	KTB	Turkic, Northwestern	1047
Komi Permyak	koi	UH	Uralic, Permic	49
Komi Zyrian	kpv	LATTICE, IKDP	Uralic, Permic	210
Kurmanji	kmr	MG	IE, Iranian	734
Livvi	olo	KKPP	Uralic, Finnic	106
Marathi	mr	UFAL	IE, Indic	47
Mbya Guarani	gun	THOMAS, DOOLEY	Tupian	98
Moksha	mdf	JR	Uralic, Mordvin	21
Naija	pcm	NSC	Creole	948
Sanskrit	sa	UFAL	IE, Indic	230
Swiss G.	gsw	UZH	IE, Germanic	100
Tagalog	tl	TRG	Austronesian, Central Philippine	55
Tamil	ta	TTB	Dravidian, Southern	120
Telugu	te	MTG	Dravidian, South Central	146
Upper Sorbian	hsb	UFAL	IE, Slavic	623
Warlpiri	wbp	UFAL	Pama-Nyungan	54
Welsh	cy	CCG	IE, Celtic	956
Yoruba	yo	YTB	Niger-Congo, Defoid	100

Training languages set is taken from
Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing – A Tale of Two Parsers Revisited∗ (Kulmizev et. at., 2019)

How well the model represent languages?

	ar	en	eu	fi	he	hi	it	ja	ko	ru	sv	tr	zh	hr-avg	lr-avg
Previous work:															
uuparser-bert [1]	81.8	87.6	79.8	83.9	85.9	90.8	91.7	92.1	84.2	91.0	86.9	64.9	83.4	84.9	-
udpipe [2]	82.9	87.0	82.9	87.5	86.9	91.8	91.5	**93.7**	84.2	92.3	86.6	67.6	80.5	85.8	-
udify [3]	82.9	88.5	81.0	82.1	88.1	91.5	**93.7**	92.1	74.3	**93.1**	89.1	67.4	**83.8**	85.2	34.1
Monolingually trained (one model per language):															
mono-udify	83.5	89.4	81.3	87.3	87.9	91.1	93.1	92.5	84.2	91.9	88.0	66.0	82.4	86.0	-
Multilingually trained (one model for all languages):															
multi-udify	80.1	88.5	76.4	85.1	84.4	89.3	92.0	90.0	78.0	89.0	86.2	62.9	77.8	83.0	35.3
adapter-only	82.8	88.3	80.2	86.9	86.2	90.6	93.1	91.6	81.3	90.8	88.4	66.0	79.4	85.0	32.9
udapter	**84.4**	**89.7**	**83.3**	**89.0**	**88.8**	**92.0**	93.5	92.8	**85.9**	92.2	**90.3**	**69.6**	83.2	**87.3**	**36.5**

Table 1: Labelled attachment scores (LAS) on high-resource languages for baselines and UDapter. Last two columns show average LAS of 13 high-resource ('hr-avg') and 30 low-resource ('lr-avg') languages respectively. Previous work results are reported from (Kulmizev et al., 2019) [1] and (Kondratyuk and Straka, 2019a) [2,3].

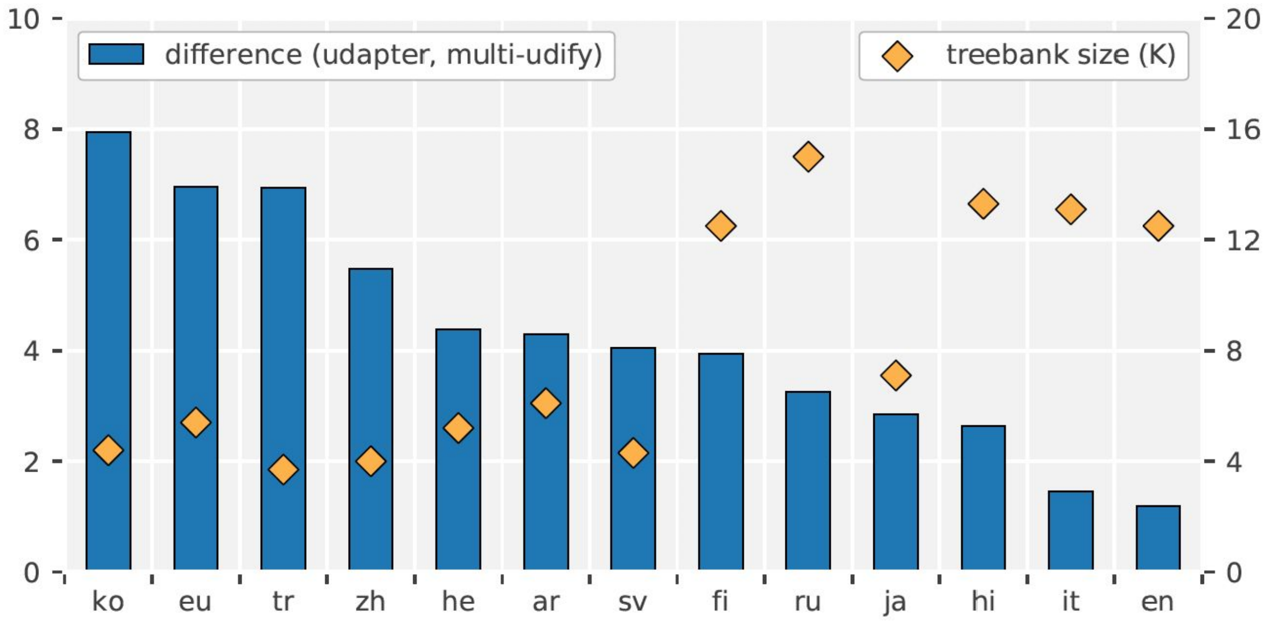
Results on Low-Resource Languages

	be	br*	bxr*	cy	fo*	gsw*	hsb*	kk	koi*	krl*	mdf*	mr	olo*	pcm*	sa*	tl	yo*	yue*	avg
multi-udify	**80.1**	**60.5**	26.1	53.6	68.6	43.6	53.2	**61.9**	20.8	**49.2**	24.8	**46.4**	42.1	36.1	19.4	62.7	41.2	30.5	45.2
udapter-proxy	69.9	-	-	-	64.1	23.7	44.4	45.1	-	45.6	-	29.6	41.1	-	15.1	-	-	24.5	-
udapter	79.3	58.5	**28.9**	**54.4**	**69.2**	**45.5**	**54.2**	60.7	**23.1**	48.4	**26.6**	44.4	**41.7**	**36.7**	**22.2**	**69.5**	**42.7**	**32.8**	**46.2**

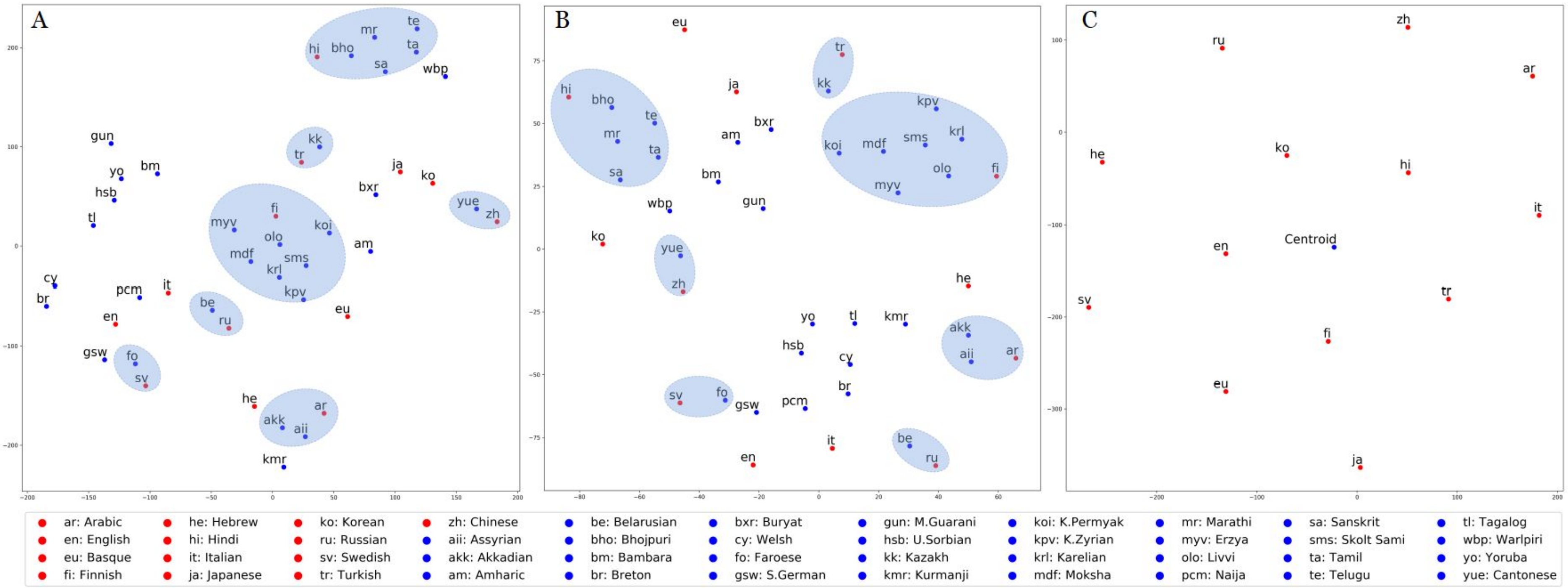
Table 2: Labelled attachment scores (LAS) on a subset of 30 low-resource languages. Languages with '*' are not included in mBERT training corpus. (Results for all low-resource languages, together with the chosen proxy, are given in Appendix B.)

* ***UDapter-Proxy*** is trained without typological features (w/ language one-hot encodings), a language from the same family in the training set is used as proxy for LR languages.

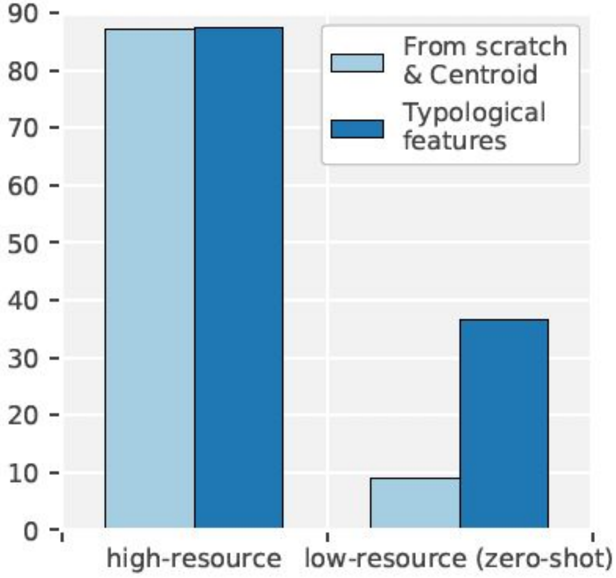
Difference on High-Resource Languages

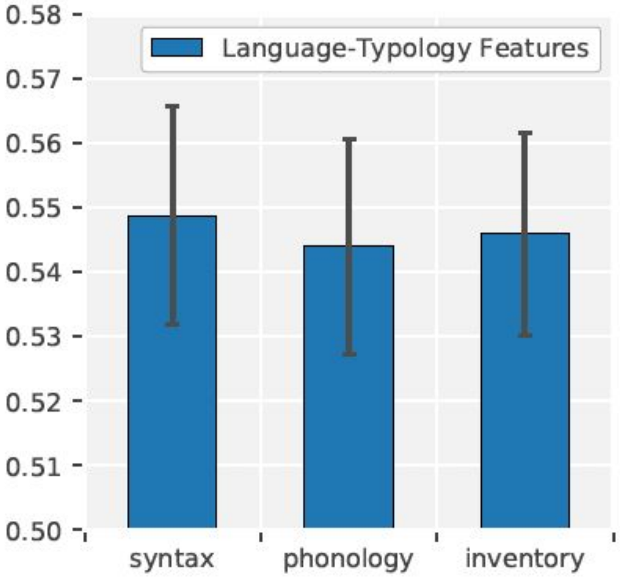


How well the model represent languages?

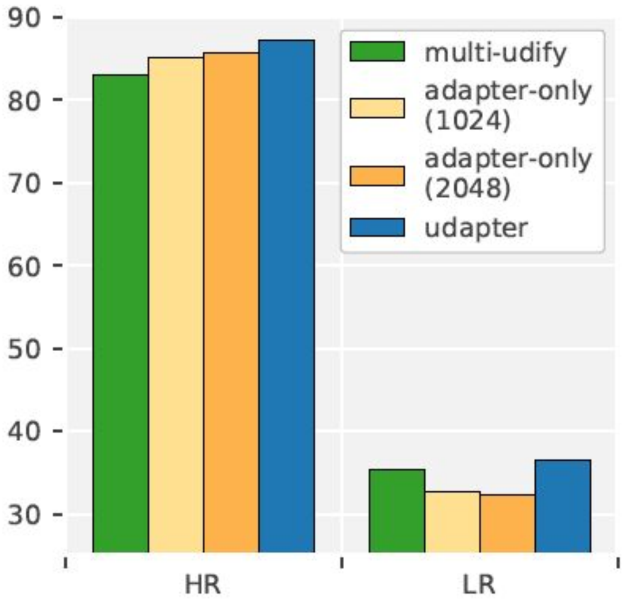


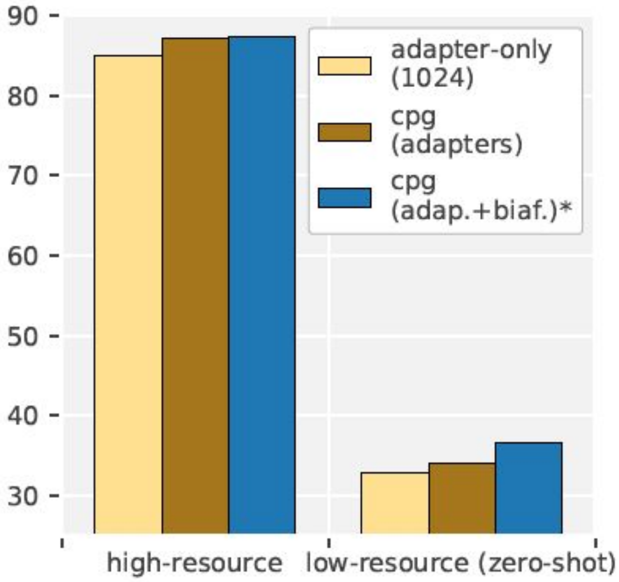
Impact of Language Typology Features





Impact of Different UDapter Components





Parameter Cost for Models

Model/Component	Parameter Cost
Standard Fine-Tuning	191M
Adapters (256)	9.4M
+Biaffine	7.8M
+CPG (L_e: 32)	x32 (Regardless of the #Langs)
Monolingual (#Langs: 13)	x13

* **Platanios et al. (2018)** also suggest to apply parameter grouping, which allows to learn separate low-rank projections of language embeddings (with lower dimensions) for the adapters, the biaffine parameters groups.