Creating and Analysing Multilingually Comparable Text Corpora

Normunds Grūzītis
Artūrs Znotiņš
Viesturs Jūlijs Lasmanis

University of Latvia
AiLab.lv @ IMCS

6th Baltic Summer School of Digital Humanities
Large Language Models and Small Languages
25 July 2024, Riga, Latvia

Funded by the European Union NextGenerationEU

2027 National Development Plan

Language Technology Initiative
2.3.1.1.i.0/1/22/I/CFLA/002

Agenda

A showcase: the **ParlaMint** corpora

Universal Dependencies: cross-linguistic grammatical annotation and analysis

The task and **the toolkit**

Under the hood: small-scale yet efficient $_{L}$LMs for tagging & parsing – **BERT**

Hands-on work

- The grunt work: data acquisition via web scraping
- The actual work: parsing unstructured text into structured data
- Enjoying the fruits: running corpus queries

Comparable and interoperable parliamentary corpora

https://www.clarin.eu/parlamint

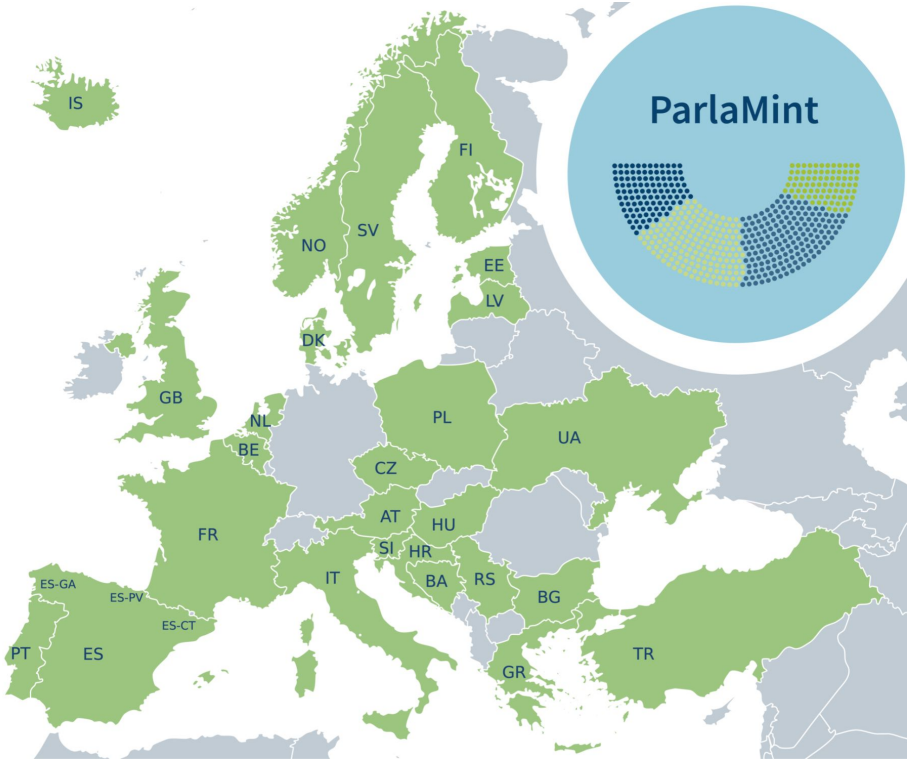
Version **4.1** contains corpora for **29** countries and autonomous regions

Linguistically annotated corpora: **morphology**, **syntax**, named entities

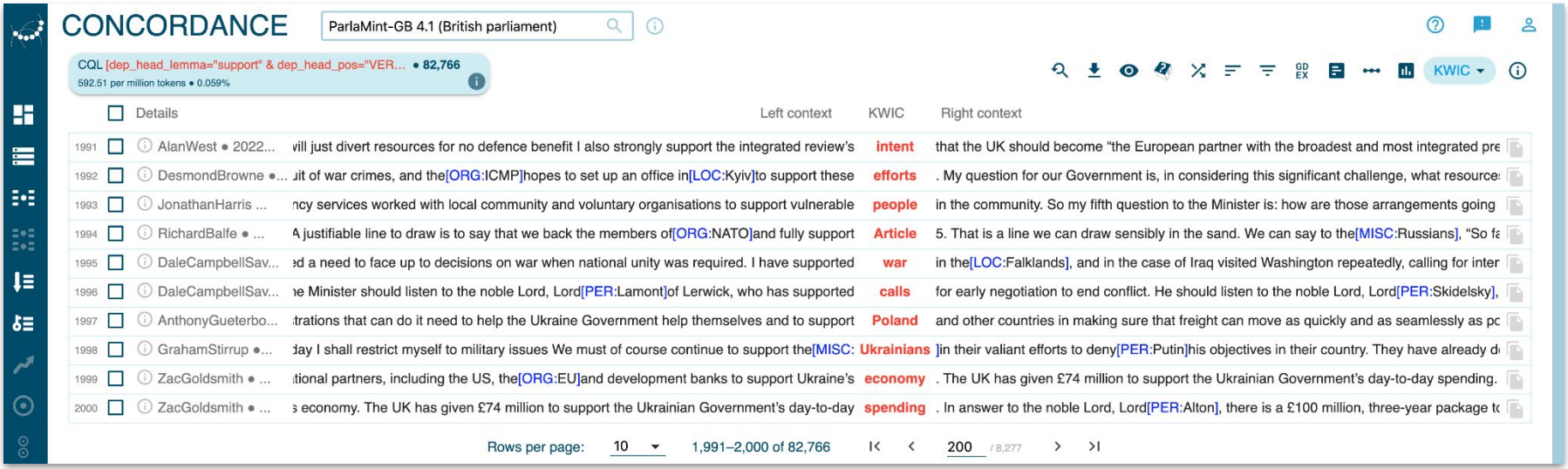
Interoperable* corpora: TEI, **UD**, **VERT**

Open data: http://hdl.handle.net/11356/1912
Open access: https://**www.clarin.si/ske**/#**open**

* FAIR: findability, accessibility, interoperability, reusability



Uniform querying in the ParlaMint corpora

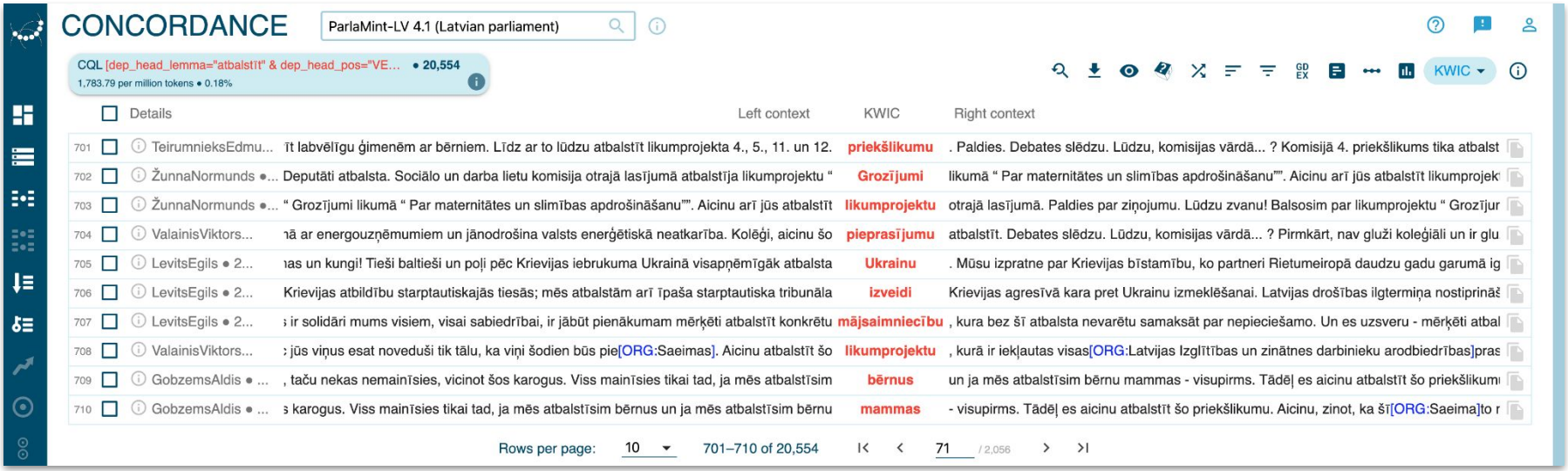



```
[dep_head_lemma="support"
 & dep_head_pos="VERB"
 & dep="obj" & pos="NOUN|PROPN"]
```

CLARIN.SI

NoSketch Engine

Uniform querying in the ParlaMint corpora

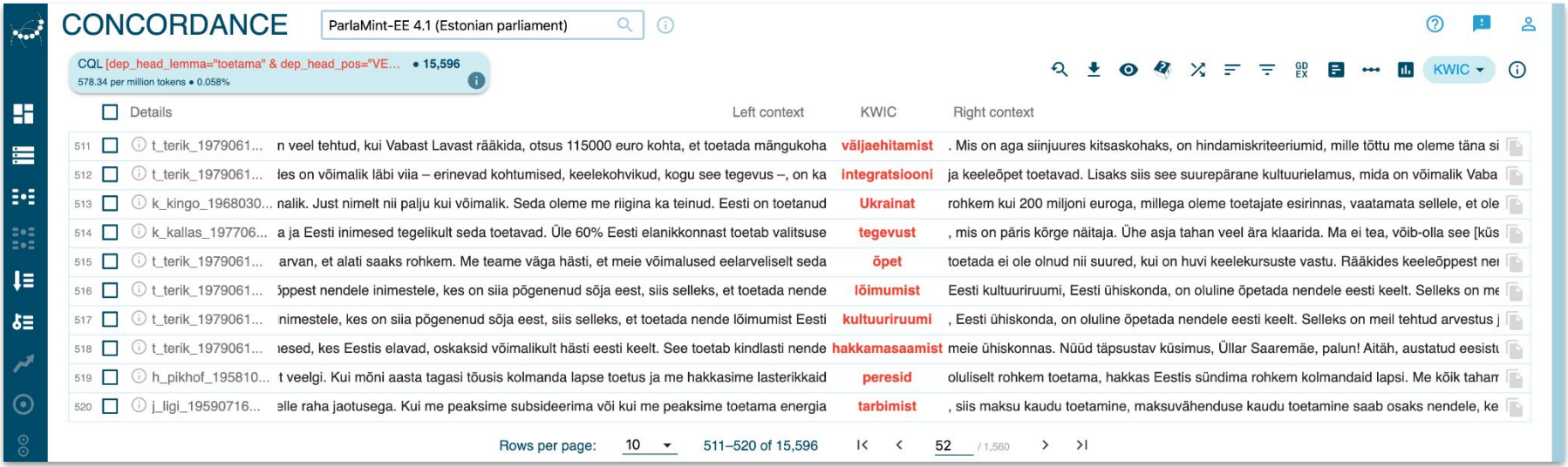



```
[dep_head_lemma="atbalstīt"
 & dep_head_pos="VERB"
 & dep="obj" & pos="NOUN|PROPN"]
```

CLARIN.SI

NoSketch Engine

Uniform querying in the ParlaMint corpora




```
[dep_head_lemma="toetama"
 & dep_head_pos="VERB"
 & dep="obj" & pos="NOUN|PROPN"]
```

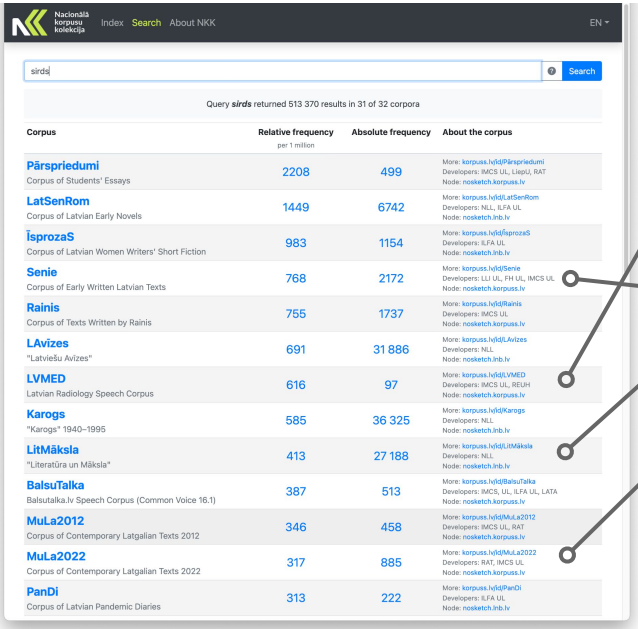
CLARIN.SI

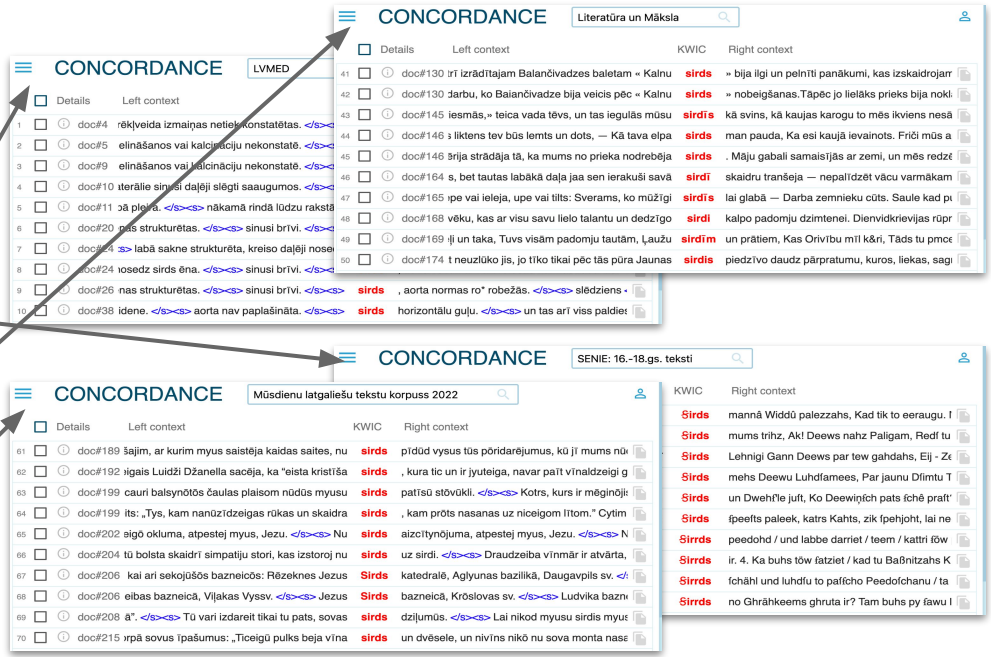
NoSketch Engine

Korpuss.lv: Latvian National Corpora Collection

- **39** corpora by **13** institutions, **2.8B** tokens
- **Unified** morpho-syntactic annotations
- **Federated** search, frequencies, timelines

text (30) speech (7) general (11) specialised (26) morphology (31) syntax (3) semantics (1)
error annotation (2) manually annotated (5) web (2) learner (2) literary (4) parallel (1)
parliamentary (1) diahronic (2) newspapers (5) representative (9) latgalian (3) blog (2)

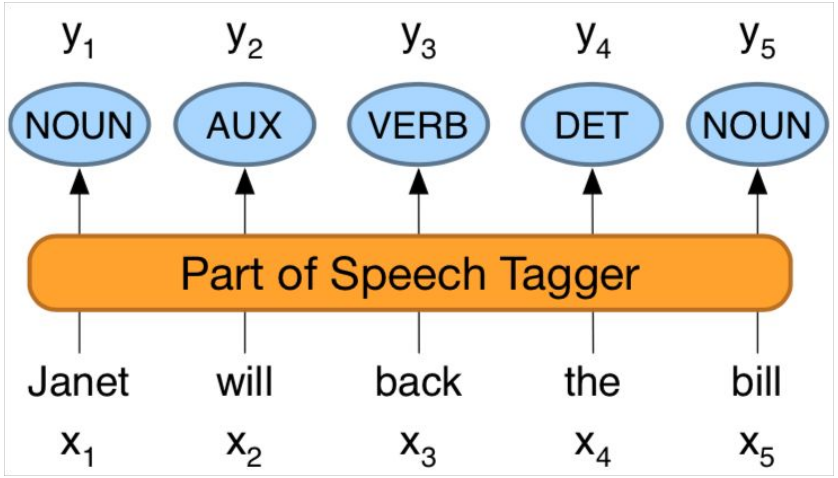




Part-of-Speech tagging

Sentence splitting ›› Tokenization ›› Tagging & Lemmatization ›› Parsing

State-of-the-art POS taggers: above **95%** accuracy



https://web.stanford.edu/~jurafsky/slp3/

https://corenlp.run

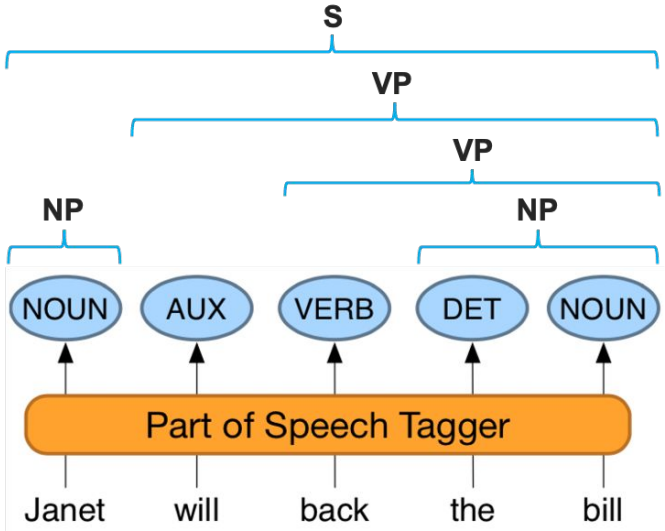
INDEX	FORM	LEMMA	UPOSTAG	XPOSTAG
#text=Dženeta atbalstīs likumprojektu .				
1	Dženeta	Dženeta	PROPN	npfsn4
2	atbalstīs	atbalstīt	VERB	vmnift330an
3	likumprojektu	likumprojekts	NOUN	ncmsa1
4	.	.	PUNCT	zs

https://nlp.ailab.lv

Syntactic parsing

Constituency (CFG/PSG) vs. **dependency** (UD) parsing

State-of-the-art parsers: above **90%** accuracy

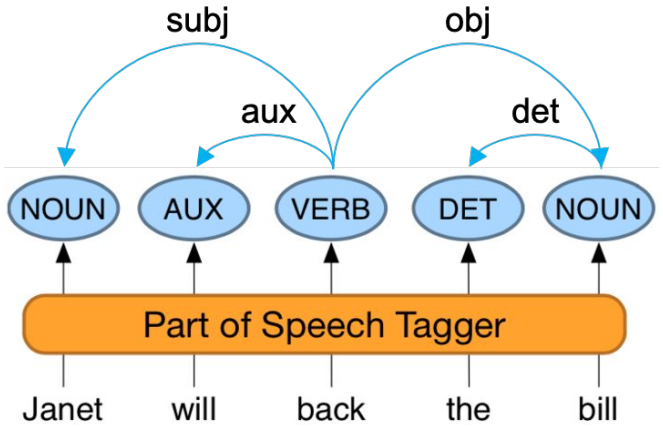


Chomsky (1956)

Tesnière (1959)

Production rules
e.g. `S → NP VP`

Syntactic relations
(functions)
(roles)



Universal Dependencies

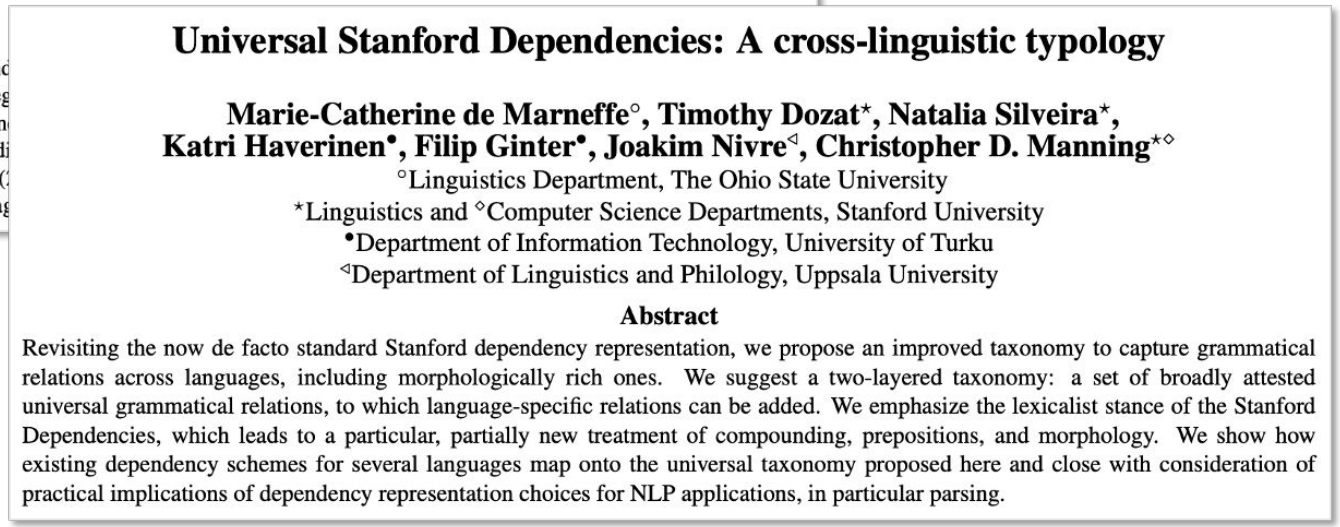
A Universal Part-of-Speech Tagset

Slav Petrov[1] **Dipanjan Das**[2] **Ryan McDonald**[1]

[1]Google Research, New York, NY, USA, {slav,ryanmcd}@google.com
[2]Carnegie Mellon University, Pittsburgh, PA, USA, dipanjan@cs.cmu.edu

To facilitate future research in unsupervised inc
consists of twelve universal part-of-speech categ
to this universal set. As a result, when combin
consisting of common parts-of-speech for 22 di
compare tagging accuracies across languages, (
part-of-speech tags, and (3) use the universal tag

LREC 2012
Language and treebank specific POS tagsets ↠ 12 universal tags (currently: **17**)



Universal Stanford Dependencies: A cross-linguistic typology

Marie-Catherine de Marneffe[°], **Timothy Dozat**[⋆], **Natalia Silveira**[⋆], **Katri Haverinen**[•], **Filip Ginter**[•], **Joakim Nivre**[◁], **Christopher D. Manning**[⋆◇]

[°]Linguistics Department, The Ohio State University
[⋆]Linguistics and [◇]Computer Science Departments, Stanford University
[•]Department of Information Technology, University of Turku
[◁]Department of Linguistics and Philology, Uppsala University

Abstract

Revisiting the now de facto standard Stanford dependency representation, we propose an improved taxonomy to capture grammatical relations across languages, including morphologically rich ones. We suggest a two-layered taxonomy: a set of broadly attested universal grammatical relations, to which language-specific relations can be added. We emphasize the lexicalist stance of the Stanford Dependencies, which leads to a particular, partially new treatment of compounding, prepositions, and morphology. We show how existing dependency schemes for several languages map onto the universal taxonomy proposed here and close with consideration of practical implications of dependency representation choices for NLP applications, in particular parsing.

LREC 2014
A set of broadly attested universal syntactic relations (currently: **37**)

Universal Dependencies

Japanese	6	2,645K	
Kaapor	1	<1K	
Kangri	1	2K	
Karelian	1	3K	
Karo	1	2K	
Kazakh	1	10K	
Khunsari	1	<1K	
Kiche	1	10K	
Komi Permyak	1	1K	
Komi Zyrian	2	10K	
Korean	3	446K	
Kurmanji	1	10K	
Kyrgyz	1	7K	
Latin	5	983K	
Latvian	1	310K	
Ligurian	1	6K	
Lithuanian	2	75K	
Livvi	1	1K	
Low Saxon	1	4K	
Macedonian	1	1K	
Madi	1	<1K	
Makurap	1	<1K	
Malayalam	1	2K	
Maltese	1	44K	
Manx	1	20K	
Marathi	1	3K	
Mbya Guarani	2	13K	
Middle French	1	12K	
Moksha	1	4K	
Munduruku	1	1K	
Naija	1	140K	
Nayini	1	<1K	

This page pertains to UD version 2.

Universal Dependency Relations

The following table lists the 37 universal syntactic relations used in UD v2. It is a revised version of the relations originally described in *Universal Stanford Dependencies: A cross-linguistic typology* (de Marneffe *et al.* 2014).

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**Headless**	**Loose**	**Special**	**Other**
conj cc	fixed flat	list parataxis	compound orphan goeswith reparandum	punct root dep

https://**universaldependencies.org**

~150 languages
~250 treebanks
500+ contributors

Released twice a year (via CLARIN)

Used for:

- linguistic research
- training of language models

https://universaldependencies.org/u/**pos**/
https://universaldependencies.org/u/**dep**/

The CoNLL-U format (i.e., the data fields)

1	*Workers*	worker	NOUN	Number=Plur	2	**nsubj**
2	*dumped*	dump	VERB	Mood=Ind|Tense=Past|VerbForm=Fin	0	**root**
3	*sacks*	sack	NOUN	Number=Plur	2	**dobj**
4	*into*	into	ADP	_	6	**case**
5	*a*	a	DET	Definite=Ind|PronType=Art	6	**det**
6	*bin*	bin	NOUN	Number=Sing	2	**nmod**
7	.	.	PUNCT	_	2	**punct**

https://universaldependencies.org/format.html

UD parsing: CoNLL 2018 Shared Task, etc.

Big treebanks only

Macro-average **LAS**-F1 of the **61** big treebanks: af_afribooms, grc_perseus, grc_proiel, ar_padt, eu_bdt, bg_btb, ca_ancora, hr_set, cs_cac, cs_fictree, cs_pdt, da_ddt, nl_alpino, nl_lassysmall, en_ewt, en_gum, en_lines, et_edt, fi_ftb, fi_tdt, fr_gsd, fr_sequoia, fr_spoken, gl_ctg, de_gsd, got_proiel, el_gdt, he_htb, hi_hdtb, hu_szeged, zh_gsd, id_gsd, it_isdt, it_postwita, ja_gsd, ko_gsd, ko_kaist, la_ittb, la_proiel, **lv_lvtb**, no_bokmaal, no_nynorsk, fro_srcmf, cu_proiel, fa_seraji, pl_lfg, pl_sz, pt_bosque, ...

1.	HIT-SCIR (Harbin)	software1-P	84.37
2.	Stanford (Stanford)	software2	83.03
3.	TurkuNLP (Turku)	software1-P	81.85
4.	UDPipe Future (Praha)	software1-P	81.83
5.	ICS PAS (Warszawa)	software1-P	81.72
6.	CEA LIST (Paris)	software1-P	81.66
7.	LATTICE (Paris)	software1-P	80.97
8.	NLP-Cube (București)	software1-P	80.48
9.	ParisNLP (Paris)	software1-P	80.29
10.	Uppsala (Uppsala)	software1-P	80.25
11.	SLT-Interactions (Bengaluru)	software2-P	79.67
12.	AntNLP (Shanghai)	software1-P	79.61
13.	LeisureX (Shanghai)	software1-P	77.98
14.	UniMelb (Melbourne)	software1-P	77.69
15.	IBM NY (Yorktown Heights)	software1-P	77.55
16.	Fudan (Shanghai)	software5-P	75.42
17.	KParse (İstanbul)	software1-P	74.84
18.	BASELINE UDPipe 1.2 (Praha)	software1-P	74.14
19.	Phoenix (Shanghai)	software1-P	73.93
20.	BOUN (İstanbul)	software2-P	72.85
21.	CUNI x-ling (Praha)	software1-P	71.54
22.	ONLP lab (Ra'anana)	software3-P	67.08
23.	iParse (Pittsburgh)	software1-P	66.55
24.	HUJI (Yerushalayim)	software1-P	62.07
25.	ArmParser (Yerevan)	software1-P	58.14

lv_lvtb

1.	HIT-SCIR (Harbin)	software1-P	83.97
2.	Stanford (Stanford)	software2	81.85
3.	TurkuNLP (Turku)	software1-P	80.81
4.	ICS PAS (Warszawa)	software1-P	80.71
5.	CEA LIST (Paris)	software1-P	80.29
6.	UDPipe Future (Praha)	software1-P	79.32
7.	NLP-Cube (București)	software1-P	78.18
8.	ParisNLP (Paris)	software1-P	78.16
	SLT-Interactions (Bengaluru)	software2-P	78.16
10.	Uppsala (Uppsala)	software1-P	76.97
11.	LATTICE (Paris)	software1-P	76.91
12.	AntNLP (Shanghai)	software1-P	75.56
13.	UniMelb (Melbourne)	software1-P	75.28
14.	IBM NY (Yorktown Heights)	software1-P	73.17
15.	LeisureX (Shanghai)	software1-P	73.13
16.	KParse (İstanbul)	software1-P	72.33
17.	Fudan (Shanghai)	software5-P	70.04
18.	BASELINE UDPipe 1.2 (Praha)	software1-P	69.43
19.	Phoenix (Shanghai)	software1-P	69.06
20.	BOUN (İstanbul)	software2-P	68.47
21.	CUNI x-ling (Praha)	software1-P	67.23
22.	ONLP lab (Ra'anana)	software3-P	59.67
23.	ArmParser (Yerevan)	software1-P	57.88
24.	HUJI (Yerushalayim)	software1-P	55.19
25.	SParse (İstanbul)	software1-P	0.00
26.	iParse (Pittsburgh)	software1-P	0.00

UD parsing: Stanza's pretrained UD models

Search: Lat

Language	Treebank	UPOS	XPOS	UFeats	AllTags	Lemmas	UAS	LAS
Latin	ITTB	98.76	95.72	96.69	94.63	99.09	88.82	86.80
Latin	LLCT	99.57	96.65	96.79	96.47	98.09	96.13	94.88
Latin	Perseus	91.40	78.09	81.83	76.37	83.57	72.07	63.16
Latin	PROIEL	96.95	97.15	91.58	90.59	96.86	77.98	73.99
Latin	UDante	90.08	74.55	81.64	73.07	86.95	68.00	58.84
Latvian	LVTB	96.70	89.72	94.73	89.29	96.12	88.91	85.77

Showing 1 to 6 of 6 entries (filtered from 138 total entries)

https://stanfordnlp.github.io/stanza/performance.html

UD parsers: Stanza, UDpipe, spaCy, LV-PIPE, etc.

Stanza: a Python package, supports **70**+ languages (out of the box)

- https://stanfordnlp.github.io/stanza/models.html
- **easy to use**

UDpipe: a REST web-service, supports **70**+ languages

- https://ufal.mff.cuni.cz/udpipe/2/models

spaCy: a Python package, supports **25**+ languages (out of the box)

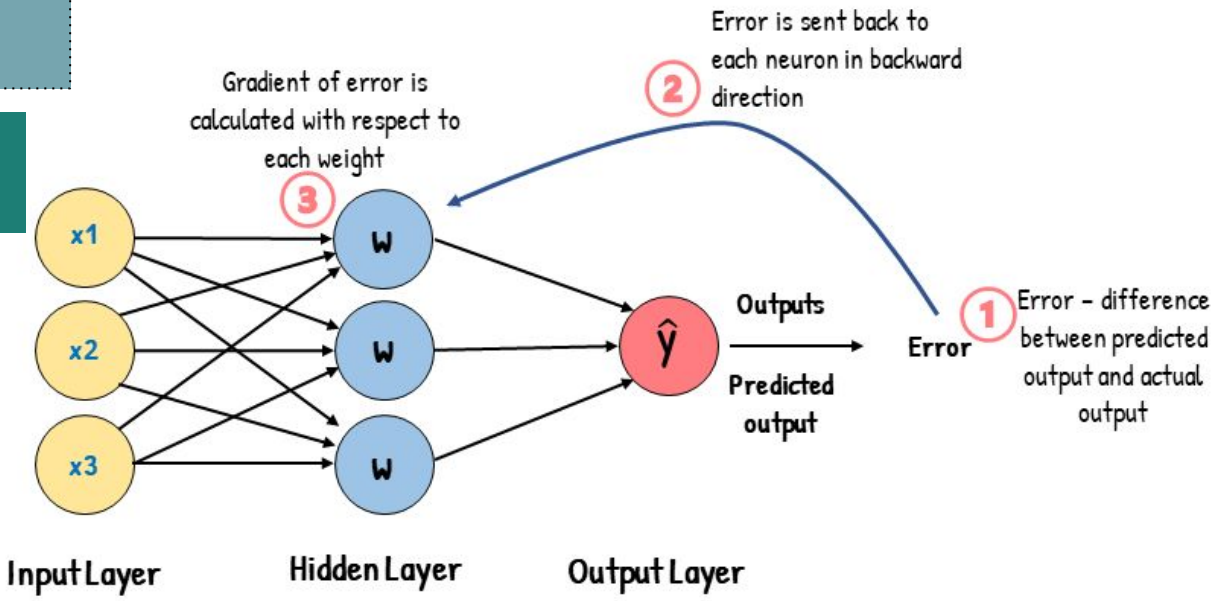
- https://spacy.io/models
- industry-strength

LV-PIPE: a REST web-service, supports only Latvian

- https://nlp.ailab.lv ↠ https://korpuss.lv, https://proza.lnb.lv, etc.
- ~**90%** LAS score (based on LV-BERT, 2020-2022)

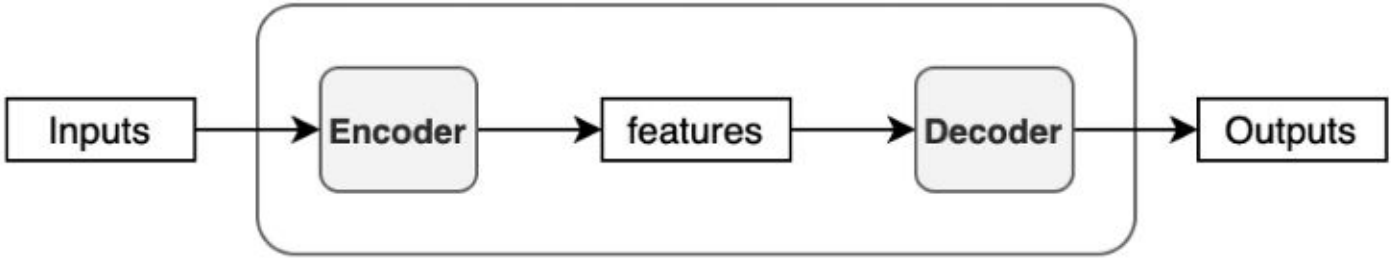
Under the hood: Training NLP models

- Training data
- Feature engineering
- Model training
- Model evaluation
- Model inference



Under the hood: Transformer architectures

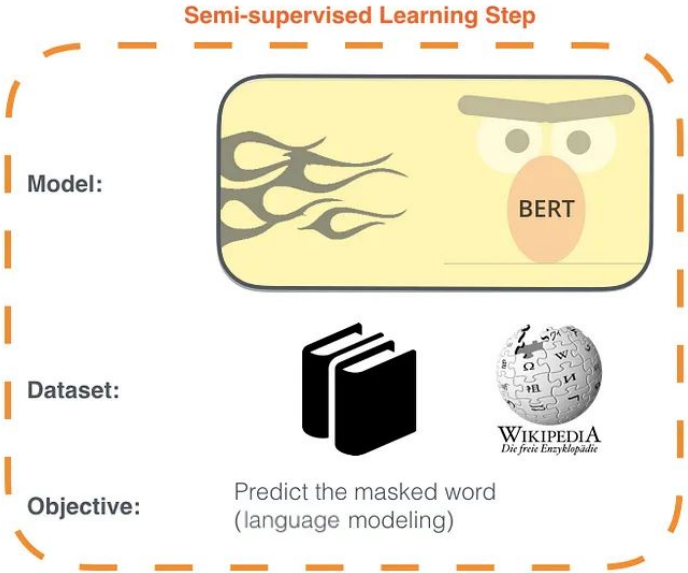
Encoder	Encoder-Decoder	Decoder
Classification, sequence tagging	Machine translation, image captioning	Fluent text generation
Fine-tuning for specific tasks	Control tokens to control decoding	Instruction tuning, prompting
BERT, RoBERTa	T5, BART	GPT, LLAMA



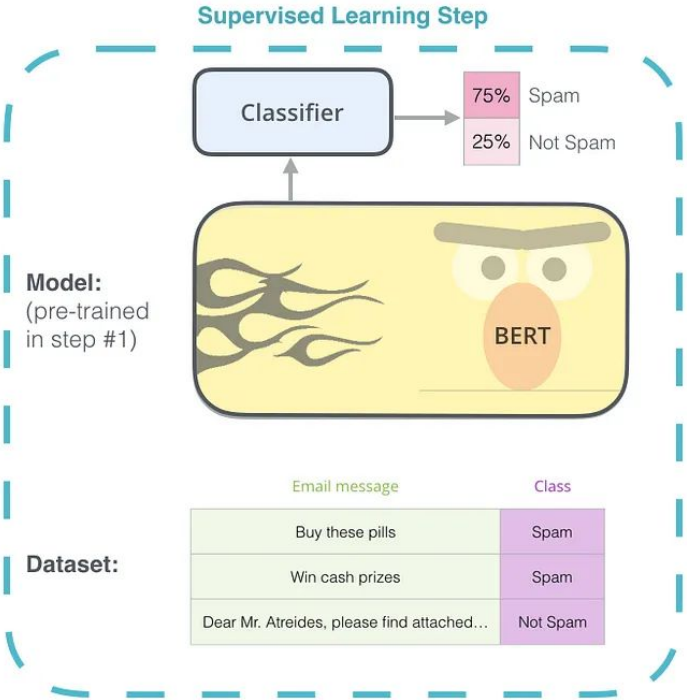
Under the hood: BERT et al.

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



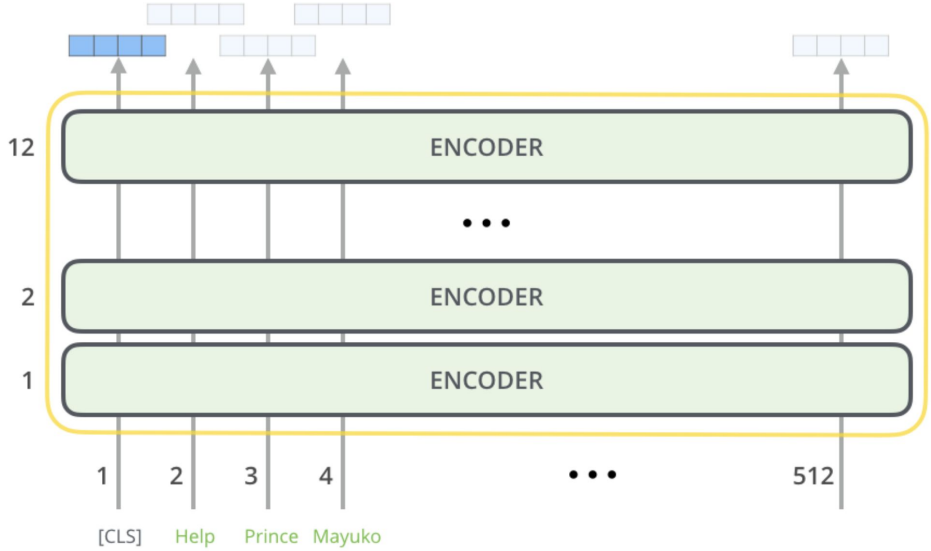
2 - Supervised training on a specific task with a labeled dataset.



https://jalammar.github.io/illustrated-bert/

Under the hood: What does BERT learn?

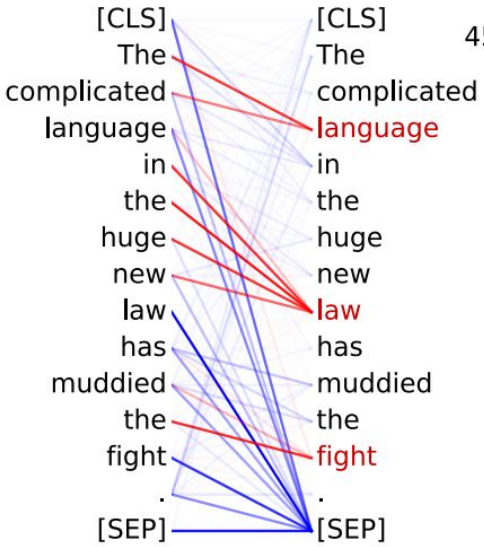
Bidirectional contextual embeddings

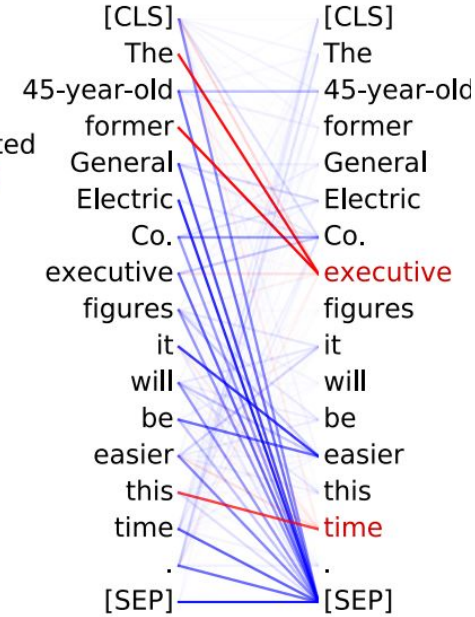


What does BERT look at?
An Analysis of BERT's Attention

Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the **det** relation





The Task

Create a teeny-tiny UD annotated text corpus of ~**10** currently trending news articles

- In a language of your choice
- Together, you will create a multilingual corpus

Get a list of such articles (links)

- Use **Europe Media Monitor** as a source
- Extract the links form an **RSS** feed, using `feedparser`

Extract text from the webpages, using `bs4`

Parse the texts, using `stanza`, and create a **VERT** file

We will concatenate your VERT files into a single file

- It will be uploaded to a **NoSketch Engine** instance for running **CQL** queries over this dataset

https://github.com/LUMII-AILab/NLP_Course/blob/main/notebooks/**BSSDH2024.ipynb**

Assignment & Grading

Submission: your final version of the `BSSDH2024.ipynb` template

- Improvements, extensions, comments, runtime output, etc.

Criteria:

- Grade **7**–**8**: you have managed to do basic plain-text extraction (which may still be noisy) and have produced a valid VERT file with basic UD annotations (`lemma`, `pos`, `dep`)
- Grade **8**–**9**: the extracted plain-text is rather clean and rather well segmented, the VERT file contains extra UD annotations (`dep_head_lemma`, `dep_head_pos`, `dep_head_dep`)
- Grade **10**: surprise us 😉
  - e.g., scale from 10 to 100 articles, or try few-shot learning, or add named entity recognition, or use another framework, etc.
- +/–**1** point for a well/poorly commented and documented notebook with some/no conclusions