MUNI ARTS

Approaching Punctuation Errors in the New Proofreader of Czech

Raslan 2021

Vojtěch Mrkývka
mrkyvka@phil.muni.cz

Faculty of Arts, Masaryk University

December 2021

Foreword About Plinkorektor

What is Plinkorektor?

- The new proofreader of Czech,
- developed at Faculty of Arts, Masaryk University,
- intended as online tool,
- (in the end, hopefully) independent on any single user interface.

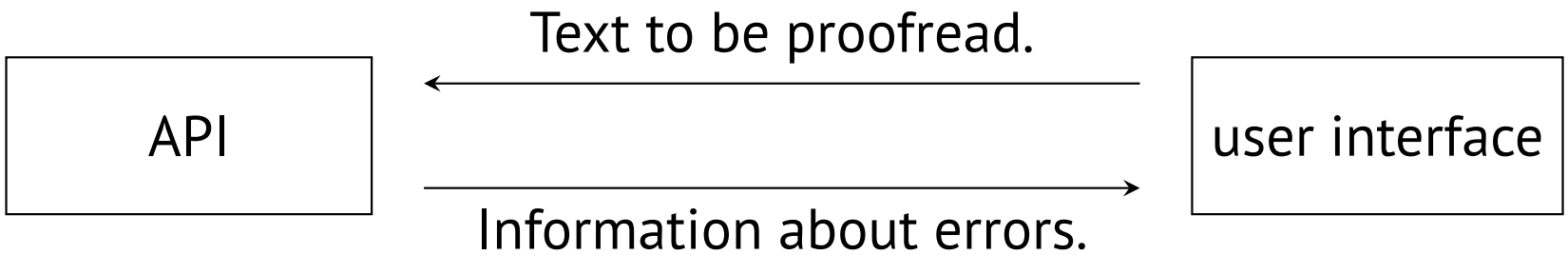
Foreword About Plinkorektor

Project Timeline

- 2018: First version of the proofreader,
- 2019–2022: TA CR project *Web-based corrector of spelling, grammar and typography for Czech*.

How It Works?

The Communication of the Proofreader



How It Works?

Example API Output (for „Hello whorld.")


```
{
  "tokens": ["Hello"," ","whorld","."],
  "mistakes": [{
    "highlights": [2],
    "description": "Word \"whorld\" was not found in the dictionary.",
    "corrections": [{
      "description": "Replace with \"world\".",
      "rules": {
        "2": "world"
      }
    }]
  }]
}
```

How It Works?

Example Correction Rule

$$2 \rightarrow \text{world}$$

0	1	2	3
Hello	␣	whorld	.
		↓	
Hello	␣	world	.

How It Works?

The Only Operation

Let's eat grandma.
`4 → eat ,`

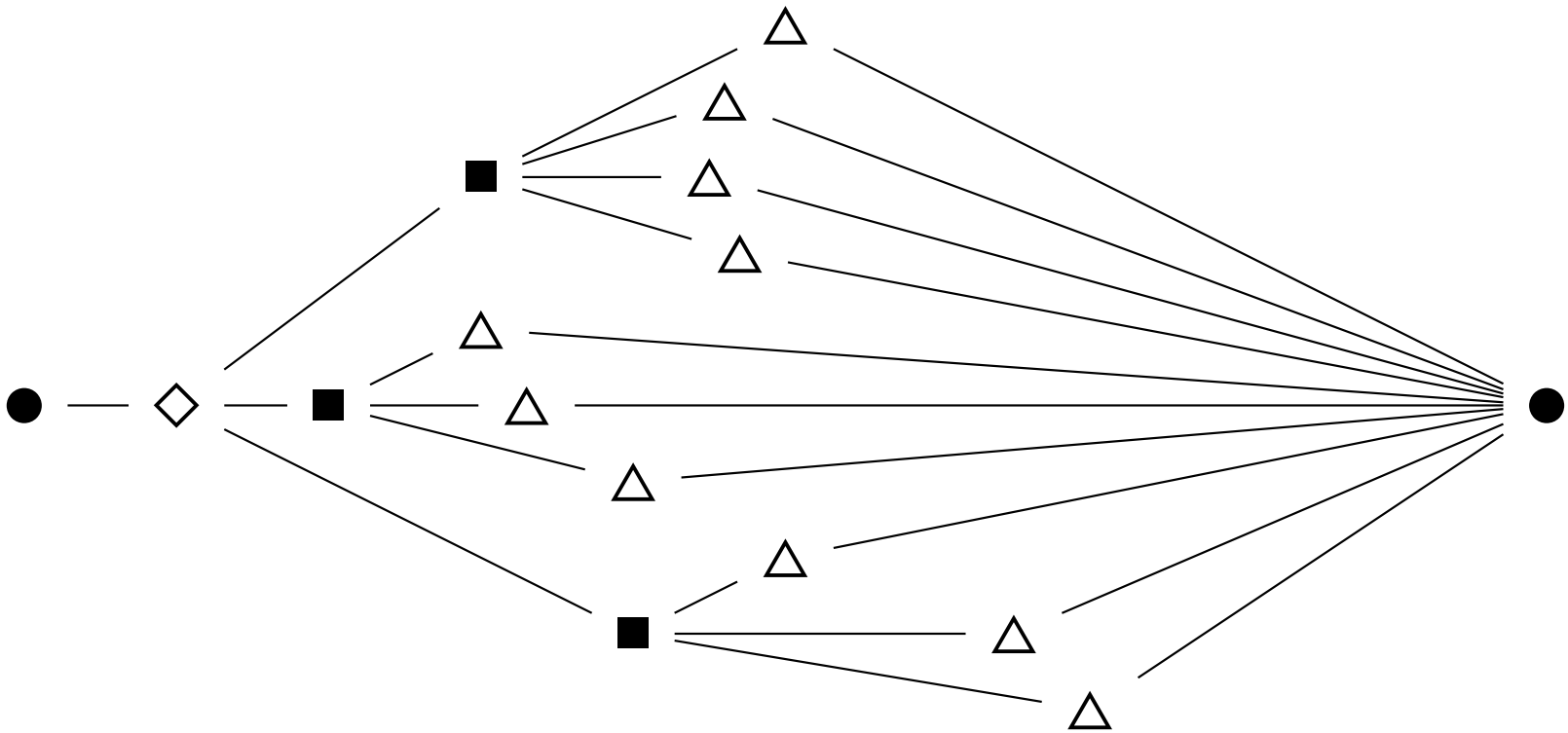
0	1	2	3	4	5	6	7
Let	'	s	␣	eat	␣	grandma	.
				↓			
Let	'	s	␣	eat,	␣	grandma	.

Let's eat , grandma.
`5 → ∅`

0	1	2	3	4	5	6	7	8
Let	'	s	␣	eat	␣	,	grandma	.
					↓			
Let	'	s	␣	eat		,	grandma	.

How It Works?

Full API structure



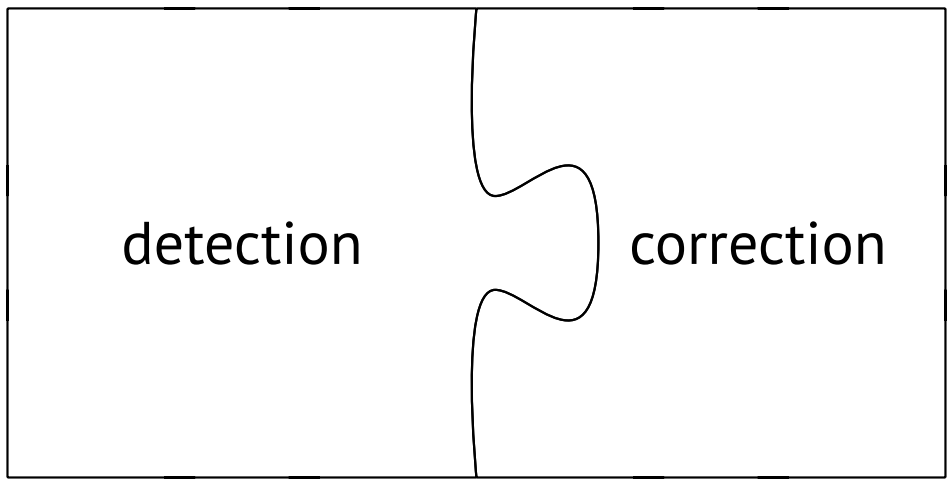
Proofreading Modules

Proofreading Areas

- Agreement,
- capital letters,
- commas,
- dependent clauses,
- nongramatical structures,
- preposition vocalisation,
- pronouns,
- punctuation,
- spelling,
- and other.

Proofreading Modules

The Structure of the Proofreading Module



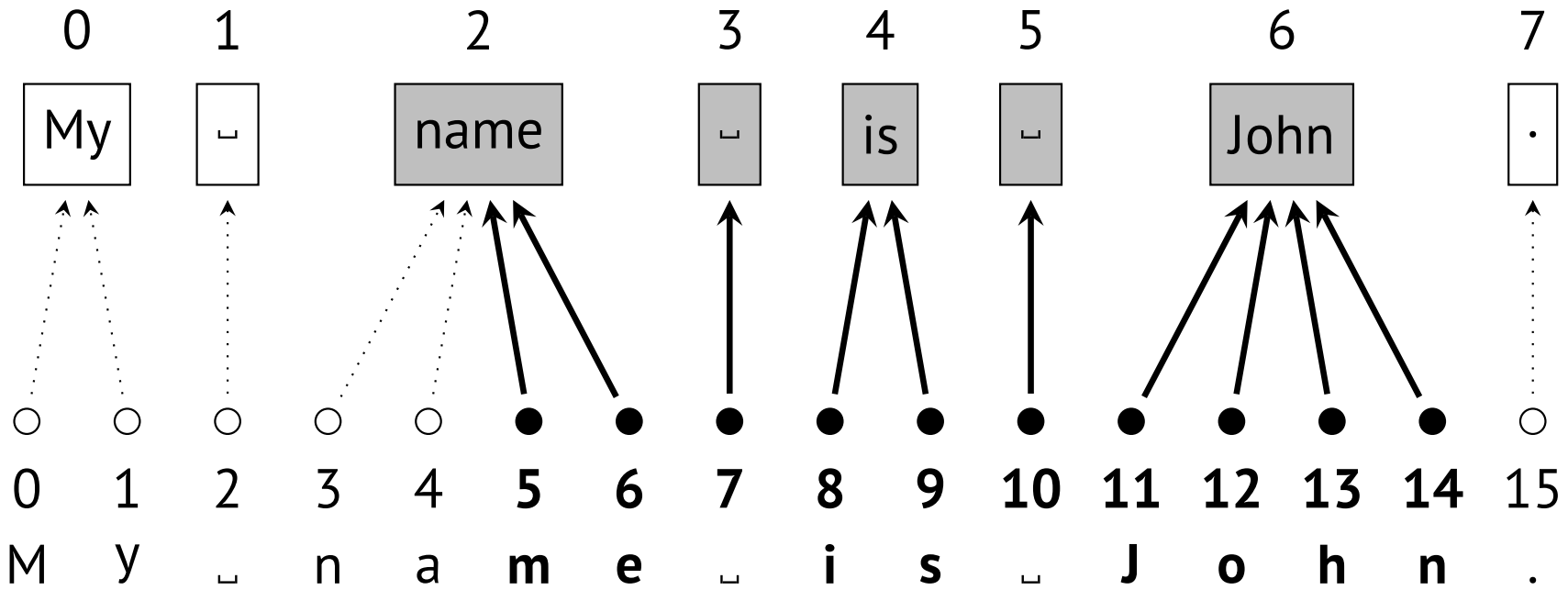
The Punctuation Module

The Correction Using Regular Expressions

- Based on the bachelor thesis of Zbyněk Michálek.
- 44 regular expressions usable for automatic correction.
- Not all expressions were suitable for the proofreader.
- IN PCRE, not in Python re format.
- There are sometimes minor errors in the expressions.

The Punctuation Module

The Correction Using Regular Expressions




```
me.*n
```

The Punctuation Module

Regular expressions in Python

- Used `regex` package instead of `re` due to POSIX compatibility,
- with `start()`, `end()` or `span()` to determine position and
- simple pointer array to determine the context of error.

The Punctuation Module

Original Expression by Michálek

```
([:alpha:]),([:alpha:])                    \1, \2
```

The Punctuation Module

Altered Expression

```
[[:alpha:]](,)[[:alpha:]]
```

The Punctuation Module

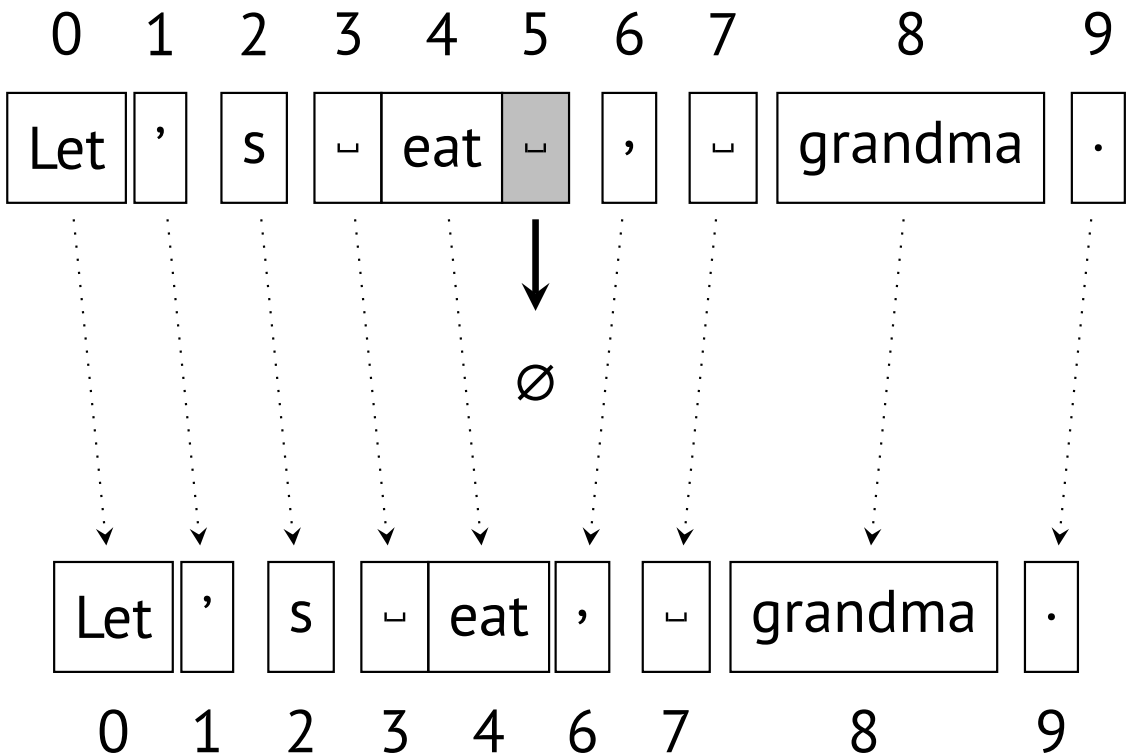
Incorrect rule example

```
([:alnum:][:punct:]?)
["’‘’”»´,›„]([:blank:]|\)|\])
```

```
[[:alnum:]][[:punct:]]?
(["’‘’”»´,›„])([:blank:]|\)|\]|$)
```

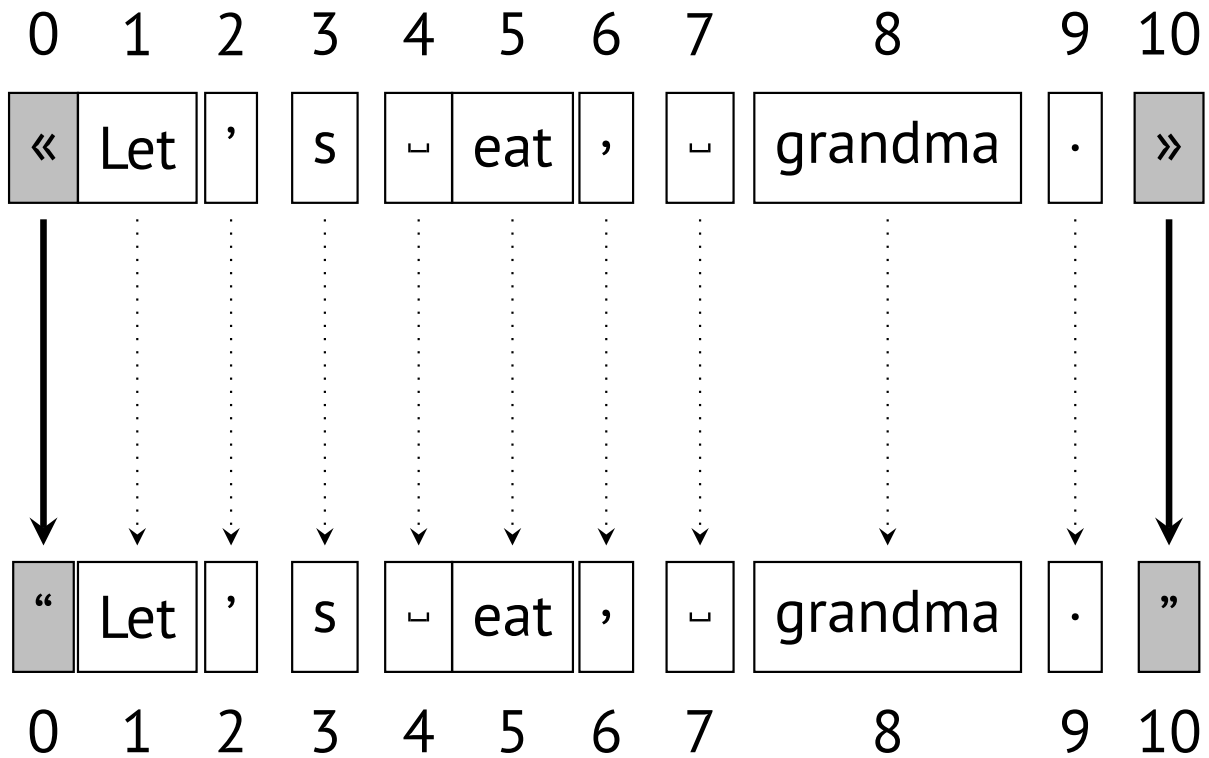
The Punctuation Module

Correction approaches



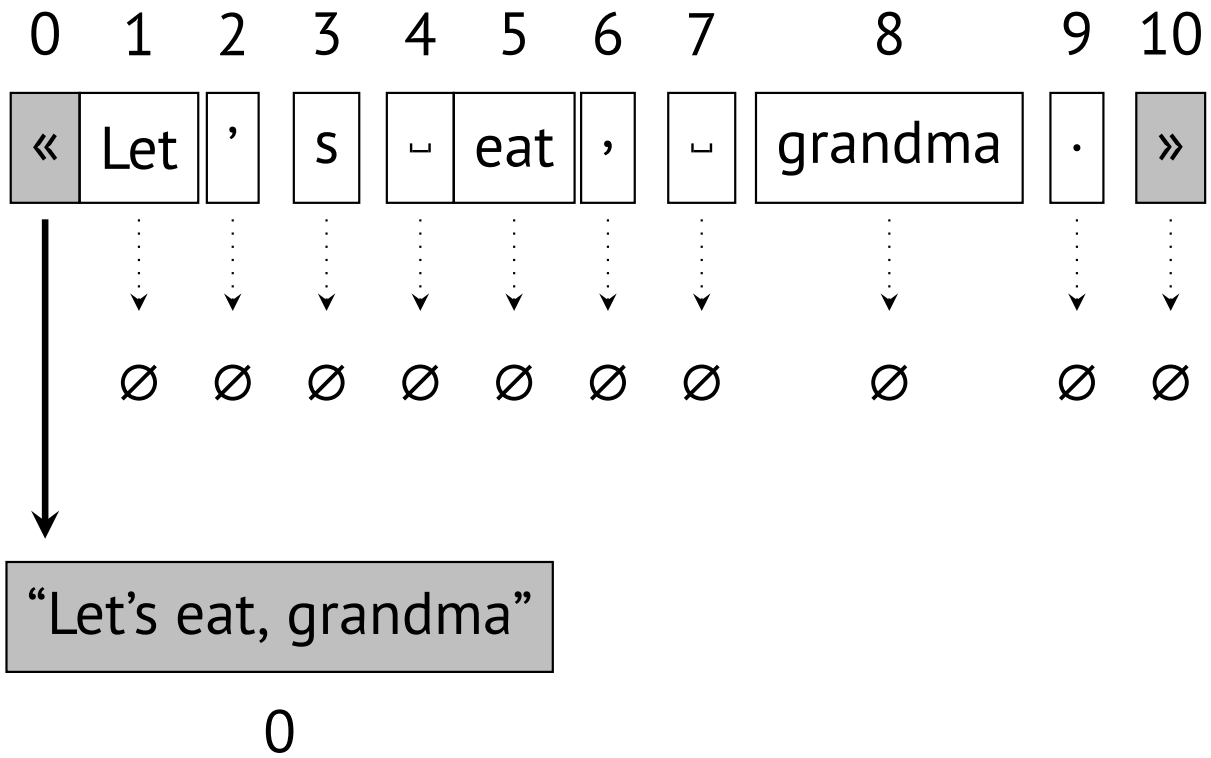
The Punctuation Module

Correction approaches



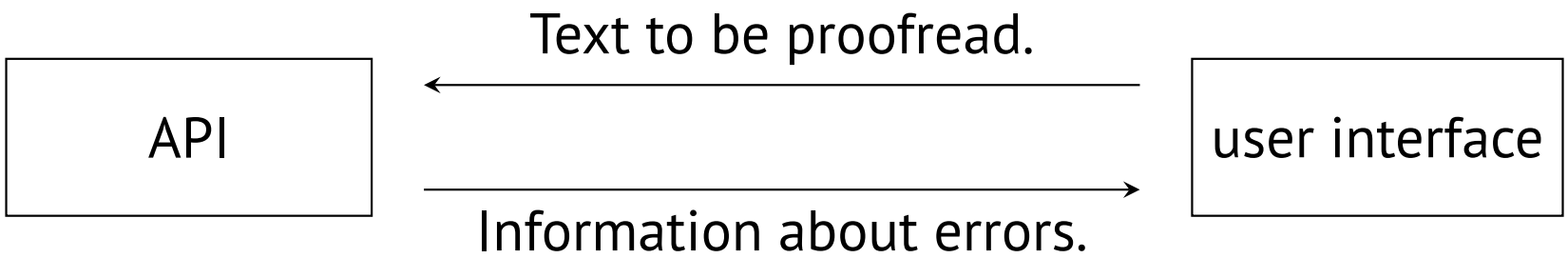
The Punctuation Module

Correction approaches



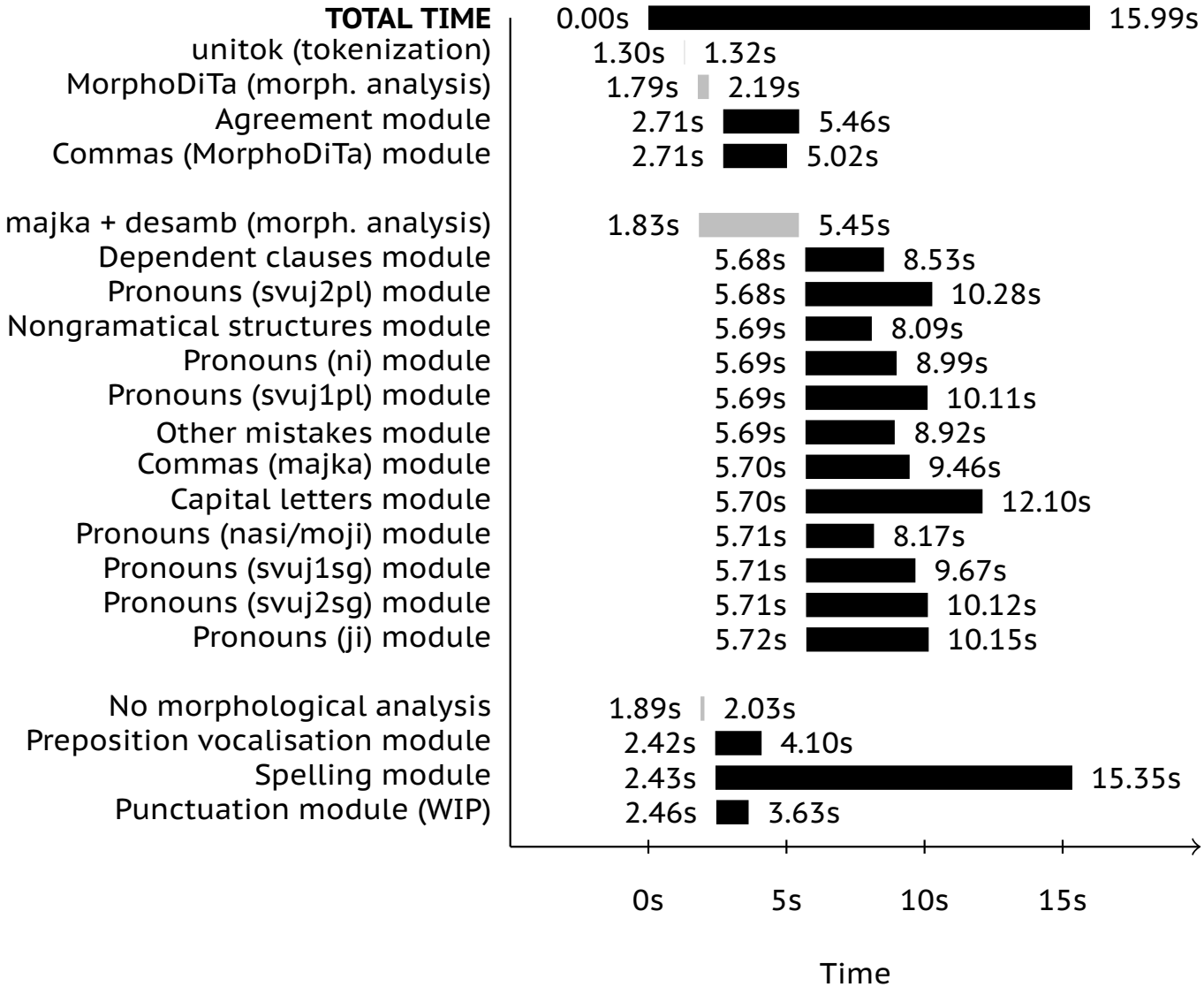
Other API issues

Current UI Dependency



Other issues

Current Run Time



Thank You for Your Attention!

MASARYK
UNIVERSITY