

---

**parstdex**

**Amir Hossein Kargaran, Sajad Mirzababaei**

**Jul 25, 2022**



# CONTENTS

|          |                                   |           |
|----------|-----------------------------------|-----------|
| <b>1</b> | <b>Getting Started!</b>           | <b>3</b>  |
| 1.1      | How to Install parstdex . . . . . | 3         |
| 1.2      | How to use . . . . .              | 3         |
| <b>2</b> | <b>File Structure</b>             | <b>7</b>  |
| <b>3</b> | <b>How to contribute</b>          | <b>9</b>  |
| <b>4</b> | <b>Citation</b>                   | <b>11</b> |



Parstdex is a python package to extract time and date markers with high performance. Parstdex only uses regexes to extract markers.



## GETTING STARTED!

### 1.1 How to Install parstdex

```
pip install parstdex
```

### 1.2 How to use

```
from parstdex import Parstdex

model = Parstdex()

sentence = """ .. """
```

#### 1.2.1 Extract spans

```
model.extract_span(sentence)
```

output :

```
{"datetime": [[6, 47], [68, 78], [82, 111]], "date": [[6, 10], [68, 78], [82, 111]],
↪ "time": [[11, 47]]}
```

#### 1.2.2 Extract markers

```
model.extract_marker(sentence)
```

```
{
  "datetime": {
    "[6, 47]": " ",
    "[68, 78]": " ",
    "[82, 111]": " .."
  },
  "date": {
    "[6, 10]": "",
```

(continues on next page)

(continued from previous page)

```

        "[68, 78]": " ",
        "[82, 111]": " .."
    },
    "time": {
        "[11, 47]": " "
    }
}

```

### 1.2.3 Extract TimeML scheme

```
model.extract_time_ml(sentence)
```

output :

```

<TIMEX3 type='DATE'>
</TIMEX3>
<TIMEX3 type='TIME'>
</TIMEX3>
<TIMEX3 type='DURATION'>
</TIMEX3>
<TIMEX3 type='DATE'>
..
</TIMEX3>

```

### 1.2.4 Extract markers' NER tags (DATTIM)

```
model.extract_ner(sentence, mode="dattim")
```

output :

```

[("", "O"),
 ("", "B-DAT"),
 ("", "B-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),
 ("", "I-TIM"),

```

(continues on next page)



(continued from previous page)

```
(
    "", "O"),
    "", "O"),
    "", "O"),
    "", "O"),
    "", "B-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    "", "I-DAT"),
    ". ", "I-DAT"),
    "", "I-DAT"),
    ". ", "I-DAT"),
    "", "O"),
    "", "O"),
    "", "O"),
    "", "O")]
```

### 1.2.5 Extract markers' NER tags (TMP)

```
model.extract_ner(sentence, mode="tmp")
```

output :

```
[("", "O"),
  ("", "B-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "O"),
  ("", "O"),
  ("", "O"),
  ("", "O"),
  ("", "B-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP"),
  ("", "I-TMP")]
```

(continues on next page)

(continued from previous page)

```
("", "I-TMP"),  
("", "I-TMP"),  
("", "I-TMP"),  
("", "I-TMP"),  
("", "I-TMP"),  
("", "I-TMP"),  
("", "I-TMP"),  
(".", "I-TMP"),  
("", "I-TMP"),  
(".", "I-TMP"),  
("", "O"),  
("", "O"),  
("", "O"),  
("", "O")]
```

## FILE STRUCTURE

Parstdex architecture is very flexible and scalable and therefore suggests an easy solution to adapt to new patterns which haven't been considered yet.

```
├── parstdex
│   ├── utils
│   │   ├── annotation
│   │   │   └── ...
│   │   ├── pattern
│   │   │   └── ...
│   │   ├── special_words
│   │   │   └── words.txt
│   │   ├── const.py
│   │   ├── normalizer.py
│   │   ├── pattern_to_regex.py
│   │   ├── deprecation.py
│   │   ├── regex_tool.py
│   │   ├── spans.py
│   │   └── tokenizer.py
│   ├── marker_extractor.py
│   └── settings.py
├── Test
│   ├── data.json
│   └── test_parstdex.py
├── examples.py
├── performance_test.ipynb
├── requirement.txt
└── setup.py
```



## HOW TO CONTRIBUTE

Please feel free to provide us with any feedback or suggestions. You can find more information on how to contribute to Parstdex by reading the [contribution document](#).



**CITATION**

If you use any part of this library in your research, please cite it using the following BibTex entry.

```
@misc{parstdex,  
  author = {Kargaran, Amir Hossein and Mirzababaei, Sajad and Jahad, Hamid},  
  title = {Parstdex: Persian Time Date Extractor Python Library},  
  year = {2021},  
  publisher = {GitHub},  
  journal = {GitHub Repository},  
  howpublished = {\url{https://github.com/kargaranamir/parstdex}},  
}
```