

# clinlp



clinical

+



nl

+



NLP

=



clinlp

**clinlp**: a collaborative platform  
for advancing Dutch clinical NLP

Vincent Menger, Mark Snackey & Bram van Es  
3rd Dutch Clinical NLP Workshop  
June 24, 2024



UMC Utrecht

# About us...

---



**Vincent Menger,  
ML engineer**



**Mark Snackey,  
Product Owner**



**Bram van Es,  
Assistant Professor**

- Work on AI, ML, NLP @ UMCU
  - Team analytics
  - UPOD
- About myself
  - Technical background (CS/AI), ~4yrs PhD @ UU/UMCU, ~2yrs industry experience, now ~3yrs DIT Analytics

# Introduction

---

**If I work on....**

**I typically use...**

Processing tabular data



Classical machine learning



Neural Networks



Pretrained models, transformers



Natural Language Processing

spaCy

NLP with Dutch clinical text

clinlp

# What do these libraries have in common?

---



**Hugging Face**

spaCy

clinlp

1. Provide useful tools
2. Standardized framework
3. Production-ready quality
4. Open source collaboration

# Mot

- Caps



UMCU/Med

Token Classi

Setup

User Guide

Develop

For Algori

```
tokenizer = AutoTokenizer\
    .from_pretrained("UMCU/MedRoBERTa.nl_NegationDetection")
model = AutoModelForTokenClassification\
    .from_pretrained("UMCU/MedRoBERTa.nl_NegationDetection")

some_text = "De patient was niet aanspreekbaar en hij zag er grauw uit. \
Hij heeft de inspanningstest echter goed doorstaan."
inputs = tokenizer(some_text, return_tensors='pt')
output = model.forward(inputs)
probas = torch.nn.functional.softmax(output.logits[0]).detach().numpy()

# associate with tokens
input_tokens = tokenizer.convert_ids_to_tokens(inputs['input_ids'][0])
target_map = {0: 'B-Negated', 1: 'B-NotNegated', 2: 'I-Negated', 3: 'I-NotNegated'}
results = [{'token': input_tokens[idx],
            'proba_negated': proba_arr[0]+proba_arr[2],
            'proba_not_negated': proba_arr[1]+proba_arr[3]}
           for idx,proba_arr in enumerate(probas)]
```

NL.txt

NL1.txt

# Motivation

- Case: detecting negation for psychosis symptoms



Electronic supplementary material

Additional file 1: Regular expression Temporality module. (DOCX 41 KB)

ContextDutch Private  
biosemantics  
Java - 100% - 1 icon - 1 icon - 1 icon - Updated on Mar 22, 2021

ContextD-2 Private  
ContextD-2  
Java - 100% - 1 icon - 1 icon - 1 icon - Updated on Jan 11, 2018



**Negation detection in Dutch clinical texts: an evaluation of rule-based and machine learning methods**

Wouter de Groot, Ludo C. Bollen, Bartek C. Ten Hagen, Johannes Hendrikus Hendrikus Sebastianus, Arno de Vries, Jeroen A. J. van der Sluis & Sander Huisman

BMC Bioinformatics 14, Article number 11 (2013) | [View this article](#)

UMCU MedRoBERTa.nl\_NegationDetection

Token Classification Transformations PyTorch Subscribers Dutch roberta medical Inference Endpoints v1v1=2203.0470 v1v1=gg1.1.0

def contextdutch\_negation\_rules:

def contextdutch\_negation\_rules:

def contextdutch\_negation\_rules:

```
def contextdutch_negation_rules:
```



```
import spacy
import clinlp

nlp = spacy.blank("clinlp")

# Rule based
nlp.add_pipe("clinlp_context_algorithm")

# Transformer based
nlp.add_pipe("clinlp_negation_transformer")
```

```
doc = nlp(
    "De patient had geen psychose, "
    "maar was wel depressief"
)

for ent in doc.spans['ents']:
    print(ent, ent._.qualifiers_str)
```

- psychose {'Negation.Negated'}
- depressief {}

# Provide useful NLP tools

---

**clinlp**

- 1. Provide useful NLP tools**
2. Standardized framework
3. Production-ready quality
4. Open source collaboration

# Provide useful NLP tools

- Includes various components that perform generic NLP tasks
- Tailored to Dutch clinical text
- Work out of the box, but customizable
- Different methods: rule-based, classical ML, transformers, etc.

## Tokenizing

Prednisolon gebruik 20 mg/dag

Prednisolon gebruik 20 mg / dag

Mn 's nachts wel last

Mn 's nachts wel last

NIBP[mmHg]: 124/85(98)

NIBP [ mmHg ] : 124 / 85 ( 98 )

Zie status [PERSOON-1] [DATUM-3]

Zie status [PERSOON-1] [DATUM-3]

disharmonisch profiel VIQ>PIQ

disharmonisch profiel VIQ > PIQ

Contracties;-

Contracties ; -

VG/ o.a. alcoholische levercirrose

VG / o.a. alcoholische levercirrose



# Provide useful NLP tools: components (1)

## Sentence boundary detection

```
||Betreft: dhr. [NAAM]. ||  
  
||Geachte collega,||  
  
||Bovengenoemde patient bezoekt onze polikliniek jl. februari. || Er werd de  
volgende medicatie z.n. voorgeschreven:||  
  
||- Prednisolon 20 mg/dag||  
  
||- Pantoprazol 1 dd||  
  
||Hoogachtend, [NAAM]||
```

## Entity recognition

- Exact matching Hypotensie bij **prematuur** geboren baby.
- Normalized matching **Hypotensie** bij **prematuur** geboren baby.
- Fuzzy matching **Hypotensie** bij **prematuur** geboren baby.
- Proximity matching **Bloeddruk sterk verlaagd** bij **prematuur** geboren baby.
- Pseudo matching **Hypotensie** bij baby met prematuur ademhalingspatroon.

# Provide useful NLP tools: components (2)

---

## Entity qualification (rule-based, transformer-based)

Presence: Patient is **somber** . Patient is niet **somber** .

Temporality: Patient is **somber** . Patient was bij eerdere opname **somber** .

Experiencer: Patient is **somber** . Partner van patient is **somber** over toekomst.

- Currently: information extraction components
- Plans for: NER+L, prodigy compatibility, active learning, summarization, speech to text, ... anything!

# Standardized framework

---

**clinlp**

1. Provide useful NLP tools
- 2. Standardized framework**
3. Production-ready quality
4. Open source collaboration

# Standardized framework



- Organized in spaCy framework
  - Easy implementation
  - Consistency
  - Good integration
  - Mix and match components in pipeline
  - Wrap pre-trained models

```
@dataclass(frozen=True)
class Qualifier:
    """
    A qualifier for an entity.

    A qualifier is a piece of information that provides additional context to an entity.
    For example, a ``Presence`` qualifier with a value of ``Present`` or ``Absent``. A
    qualifier has a fixed value.
    """
    name: str = field(compare=True)
    """The name of the qualifier."""
    value: str = field(compare=True)
    """The value of the qualifier."""
    is_default: bool = field(compare=True)
    """Whether the value is the default value."""
    priority: int = field(default=0, compare=False)
    """The priority of the qualifier."""
    prob: Optional[float] = field(default=None, compare=False)
    """The probability of the qualifier."""
```

# Standardized framework: full IE pipeline

```
import spacy
from clinlp.ie import Term

nlp = spacy.blank("clinlp")

# Normalization
nlp.add_pipe("clinlp_normalizer")

# Sentences
nlp.add_pipe("clinlp_sentencizer")

# Entities
concepts = {
    "prematuuriteit": [
        "preterm", "<p3", "prematuuriteit", "partus praematurus"
    ],
    "hypotensie": [
        "hypotensie", Term("bd verlaagd", proximity=1)
    ],
    "veneus_infarct": [
        "veneus infarct", Term("VI", attr="TEXT")
    ]
}

entity_matcher = nlp.add_pipe("clinlp_rule_based_entity_matcher", config={"attr": "NORM", "fuz": 0.5})
entity_matcher.load_concepts(concepts)

# Qualifiers
nlp.add_pipe("clinlp_context_algorithm", config={"phrase_matcher_attr": "NORM"})
```

# Production-ready quality

---

**clinlp**

1. Provide useful NLP tools
2. Standardized framework
- 3. Production-ready quality**
4. Open source collaboration

# Production-ready quality

Follows industry best practices, such as:

```
class QualifierDetector(Pipe):
    """Abstract pipeline component for detecting qualifiers in clinical text."""

    def __init__(-

    @property
    @abstractmethod
    def qualifier_classes(self) -> dict[str, QualifierClass]: ...

    @staticmethod
    def add_qualifier_to_ent(entity: Span, new_qualifier: Qualifier) -> None: ...

    def _initialize_ent_qualifiers(self, entity: Span) -> None: ...

    @abstractmethod
    def _detect_qualifiers(self, doc: Doc) -> None: ...

    def __call__(self, doc: Doc) -> Doc: ...
```

Proper system design with relevant abstractions, etc.

```
> ruff format && ruff check
41 files left unchanged
All checks passed!
```

Clean and consistent codebase

```
634/634 73.6s
✓ clinlp
  ✓ tests
    > ✓ integration
    > ✓ regression
  ✓ unit
    > ✓ ie
    > ✓ metrics
    > ✓ test_language.py
    > ✓ test_normalizer.py
    > ✓ test_sentencizer.py
    > ✓ test_util.py
```

Tests with good coverage

<b>v0.8.0</b>
2 weeks ago -> 852f345 zip tar.gz Notes
<b>v0.7.0</b>
on May 16 -> ac4a54c zip tar.gz Notes
<b>v0.6.6</b>
on Apr 24 -> 262d56c zip tar.gz Notes
<b>v0.6.5</b>
on Apr 24 -> 009bf34 zip tar.gz Notes
<b>v0.6.4</b>
on Feb 13 -> 0b19a52 zip tar.gz Notes

Frequent releases

266 workflow runs	Event	Status	Branch	Actor
✓ Merge pull request #101 from umcu/add-documentation-pages	format-lint #121: Commit 05c332 pushed by vmenger	33s		
✓ Merge pull request #101 from umcu/add-documentation-pages	tests #121: Commit 05c332 pushed by vmenger	5m 3s		
✓ Add documentation pages	tests #120: Pull request #101 synchronize by vmenger	4m 40s		
✓ Add documentation pages	format-lint #120: Pull request #101 synchronize by vmenger	31s		

CI/CD, automated workflows

clinlp 0.8.0 documentation

- CLINLP
- Introduction
- Installation
- Getting started
- Roadmap
- Citing
- USAGE
- Components
- Metrics

# clinlp

Welcome to the documentation pages for `clinlp`, a Python library for performing NLP on clinical text written in Dutch. In the menu to the left, you should be able to find the information you are looking for. If you have any questions, need help getting started, found a bug, or have a feature request, please don't hesitate to [contact us](#)!

## Links

- [GitHub repository](#)

[docs: clinlp.readthedocs.io](https://docs.clinlp.readthedocs.io)

# Open source collaboration

---

**clinlp**

1. Provide useful NLP tools
2. Standardized framework
3. Production-ready quality
4. **Open source collaboration**



# Open source collaboration

---

- Committed to maintaining clinlp as a free and open source package
- The more people join, the more it works for everyone
- Benefits: reproducibility, comparing results, fast prototyping, robust pipelines, availability, sharing work, easier collaboration
- If you have some code/project lying around, please contact us!

# We need your help!

---



[github.com/umcu/clinlp](https://github.com/umcu/clinlp)



[clinlp.readthedocs.io](https://clinlp.readthedocs.io)

## How to join?

If this is relevant to you → use it

If you want features or found bugs → give feedback

If you have a contribution → **contact us**, or fork, change and PR

## Contact

GitHub issue / [v.j.menger-2@umcutrecht.nl](mailto:v.j.menger-2@umcutrecht.nl) / [bes3@umcutrecht.nl](mailto:bes3@umcutrecht.nl) / afterwards 😊

Thanks to: Bram, Daniel, Karin, Malin, Mark, Martijn, Richard, Rosanne, Ruben, Sander, Saskia, Saskia, Teus, Tuur, Vincent, Zimbo