MedExtractor

Release 1.0.0

Fapra Gruppe 5

CONTENTS:

1	Medextractor - Konsolenapplikation	3
2	Konfigurationsdatei config.json	5
3	Vokabular-Dateien	7
4	Aufruf des Programms4.1Voraussetzungen4.2Aufruf	9 9
5	Entity-Linker	11
6	medextractor 6.1 medextractor package	13
7	Indices and tables	19
Рy	thon Module Index	21
In	dex	23

Fachpraktikum WS 22/23 Natural Language Processing (NLP) mit spaCy

CONTENTS: 1

2 CONTENTS:

\sim	ш	٨	D	ГΕ	R
L	п	А	Р.	ᇉ	ĸ

ONE

MEDEXTRACTOR - KONSOLENAPPLIKATION

Die Medextractor-Konsolenapplikation analysiert Texte und sucht darin nach Krankheiten und deren Symptomen und erstellt eine Wissensrepräsentation, die die gefundenen Krankheiten und Symptome miteinander in Beziehung setzt. Die Wissensrepräsentation wird im RDF (Resource Discription Framework)-Format gespeichert. Zusätzlich erstellt der Medextractor eine xml-Datei mit Daten für den Entity Linker von spaCy sowie eine Datei name.kb, in dem die erstellte Wissensbasis in Binärcode abgespeichert wird.

KONFIGURATIONSDATEI CONFIG.JSON

Das Python-Modul, mit dem die Konsolenapplikation gestartet wird, ist die Datei: medextractor.py. Im selben Order von medextractor.py muss sich die Konfigurationsdatei config.json befinden.

Sollte es noch keine config.json Datei geben, wird beim Aufruf von medextractor.py eine Beispiel-Datei erzeugt, die anschließend vom Nutzer angepasst werden kann.

Die Konfigurationsdatei enthält folgende Informationen:

- 1. Pfad und Name der xml-Datei für den Export im RDF-Format
- 2. Pfad und Name der xml-Datei für den Export für den Entity Linker
- 3. Pfad und Name der KnowledgeBase Datei
- 4. Pfad zu dem Order, der die zu analysierenden Texte enthält
- 5. Spezifikation, ob die Knowledgebase Datei überschrieben werden soll (True oder False)
- 6. Pfad und name der .txt-Datei, die das Krankheiten-Vokabular enthält
- 7. Pfad und name der .txt-Datei, die das Symptome-Vokabular enthält

Die Pfade müssen relativ zu dem Order angegeben werden, in dem sich medextractor.py befindet. Alternativ können auch absolute Pfade angeben werden.

Es werden alle Textdateien (*.txt) analysiert, die sich in dem in der config.json-Datei angegebenen Ordner befinden. Die von Medextractor erzeugten xml- und Knowledgebase- Dateien enthalten ein über alle analysierten Texte akkumuliertes Ergebnis.

Wird festgelegt, dass die Knowledgebase-Datei nicht überschrieben werden soll, werden alle neu gefundenen Krankheit-Symptom-Beziehungen zu der vorhandenen Knowledgebase-Datei hinzugefügt.

THREE

VOKABULAR-DATEIEN

Die Vokabulardateien sind einfache Dateien im csv-Format und enthalten Einträge der folgenden Art:

C0010051 coronary aneurysm DISEASE

Der Eintrag C0010051 ist der CUI (Concept Unique Identifier) aus der MetaMapLite-Datenbank. Der CUI ist als Referenz enthalten, wird aber nicht weiter vom Medextractor verwendet.

FOUR

AUFRUF DES PROGRAMMS

4.1 Voraussetzungen

• Packages, die in requirements.txt aufgelistet sind, sind installiert (Installation aller Packages möglich mit dem Befehl pip install -r requirements.txt)

4.2 Aufruf

Das Programm wird gestartet, indem in die Windowseingabeaufforderung der Befehl

python medextractor.py

eingegeben wird.

Zu beachten ist, dass in der System-Path-Umgebungsvariable der Pfad zur (ggf. virtuellen) Umgebung des Python-Interpreters enthalten ist, in der spaCy installiert wurde. Ggf. sollte hierzu active.bat im Verzeichnis der virtuellen Umgebung der Python-Installation aufgerufen werden.

Da die Vokabulardateien umfangreich sind, dauert allein das Trainieren des Entity-Rulers typischerweise über eine Minute.

Nach Beendigung des Programms befinden sich die xml-Dateien mit der RDF-Repräsentation sowie die xml-Datei für den Entity Linker in dem in config.json angegebenen Ordner.

FIVE

ENTITY-LINKER

Das Jupyter-notebook entity_linker_demo.ipynb demonstriert, wie die Daten aus der xml-Export-Datei gelesen und für das Training von Entity Ruler und Entity Linker verwendet werden. Findet der Entity Ruler in einem Text Symptome, dann ordnet der Entity Linker diesen Symptome dazugehörige Krankheiten zu.

SIX

MEDEXTRACTOR

6.1 medextractor package

6.1.1 Subpackages

medextractor.dummy package

Submodules

```
medextractor.dummy.dummy module
```

```
class medextractor.dummy.dummy.DummyKnowledgeExtractor
    Bases: KnowledgeExtractorInterface
    get_knowledge_base()

class medextractor.dummy.dummy.DummyPreprocessor(doc_name)
    Bases: PreprocessingInterface
    get_preprocessed_text()

class medextractor.dummy.dummy.DummyRDFSerialiser(knowledgebase, namespace, namespace_prefix)
    Bases: RDFSerialiserInterface
    serialise_knowledgebase()
```

Module contents

medextractor.interfaces package

Submodules

medextractor.interfaces.interfaces module

```
class medextractor.interfaces.interfaces.KnowledgeExtractorInterface Bases: ABC {\bf abstract~get\_knowledge\_base()} \to {\rm str}
```

```
class medextractor.interfaces.interfaces.PreprocessingInterface(doc name)
     Bases: ABC
     abstract get_preprocessed_text() → str
class medextractor.interfaces.interfaces.RDFSerialiserInterface(knowledgebase, namespace,
                                                                            namespace_prefix)
     Bases: ABC
     abstract serialise_knowledgebase(output_path) → str
     abstract set_serialisation_format(serialisation_format)
Module contents
medextractor.knowledge package
Submodules
medextractor.knowledge.base module
class medextractor.knowledge.base.KnowledgeBase
     Bases: object
     The KnowledgeBase manages entities and relations.
     Functions: add_relation(SemanticRelation) has_relation(SemanticRelation) -> bool give_entities(str) -> [] ex-
     port_for_entity_linker(str) safe(str) load(str)
     add_relation(relation: SemanticRelation) \rightarrow None
          Add a SemanticRelation into the KnowledgeBase.
           @param relation: a SemanticRelation
     add_training_example_to_relation(relation: SemanticRelation, sent_text: str) → None
          Add a training sentence to a SemanticRelation
           @param relation: a SemanticRelation @param sent_text: a training sentence
     export_for_entity_linker(file_name: str)
          Save the KnowledgeBase data as an xml file that can be used to train an entity linker.
           @param alias: the name of the xml file
     give_entities(alias: str) \rightarrow []
          Return a list of entities that are related to symptoms in SemanticRelations stored in the KnowledgeBase
           @param alias: the name of a symptom
     has_relation(relation: SemanticRelation) \rightarrow bool
          Return True if the relation is in the KnowledgeBase. Return False otherwise.
           @param relation: a SemanticRelation
     load(file_name: str)
          Load the KnowledgeBase from file.
           @param file_name: the name of the file
```

```
save(file\_name: str) \rightarrow None
          Save the KnowledgeBase into a pickle file. If the KnowledgeBase does not contain any SemanticRelations,
          no file is saved.
          @param file_name: the name of the file
medextractor.knowledge.entity module
class medextractor.knowledge.entity.Entity(entity_name: str, entity_type: EntityType)
     Bases: object
     An entity consisting of its name string and its EntityType
class medextractor.knowledge.entity.EntityType(value)
     Bases: Enum
     Types of entities that can be stored in the KnowledgeBase.
          DISEASE SYMPTOM UNDEFINED
     DISEASE = 1
     SYMPTOM = 2
     UNDEFINED = 3
medextractor.knowledge.relations module
class medextractor.knowledge.relations.RelationType(value)
     Bases: Enum
     Types of relations to be used in the KnowledgeBase.
     HAS_SYMPTOM = 2
     IS_SYMPTOM_OF = 1
     NO_RELATION = 3
medextractor.knowledge.semantics module
class medextractor.knowledge.semantics.SemanticRelation(entity_1: Entity, entity_2: Entity,
                                                                  relation type: RelationType,
                                                                  training_sample: Optional[str] = None)
     Bases: object
     A semantic relation between two entities connected by a value of RelationType.
     Additionally, training samples can be saved that resulted in this semantic relation.
     add_training_sample(training_sample: str)
          Adds the training_sample into the list of training_samples.
          training sample: string
              The text sample/sentence to be added
          None
```

```
contains\_training\_sample(training\_sample: str) \rightarrow bool
```

Checks whether the training_sample given is already included in the list of training samples.

training_sample: string

A text sample/sentence

true, if training_sample is contained in the list, false otherwise

Module contents

knowledge package

contains the modules: base entity relations semantics

This is the form of a docstring.

It can be spread over several lines.

medextractor.knowledge extractor package

Submodules

medextractor.knowledge extractor.knowledge extractor module

class medextractor.knowledge_extractor.knowledge_extractor.KnowledgeExtractor(config: ConfigManager)

Bases: KnowledgeExtractorInterface

KnowledgeExtractor searches a text string for entities and for relations between these entities

call2(text)

export_for_entity_linker()

Exports all entities, aliases and example sentences into an xml-File. The data is prepared for easy import into spaCy's Entity Linker. The xml-File is human readable and allows reviewing the data that will be used by the Entity Linker. Path and filename are defined in config.json.

None

None

get_knowledge_base()

Returns the knowledgebase that contains all entities and sample sentences. Samples sentences can be used for training statistical models (e.g. Entity Linker)

None

KnowledgeBase

process_texts()

Analyzes all text documents in the folder specified in config.json

None

None

```
saveKB(*args)
```

Saves the database persistently. Optionally, path and file name are given as a string parameter when calling this function. If no path and file name are given, the function will use the path and file name in attribute self. knowledgebase filename.

```
(optional) file_name (string)
```

None

set_context(context)

This function allows defining a context. The context is described by named entities included in the Entity Ruler (self._ruler). These entities will be added to the set of entities when searching for disease/symptom relations between entities.

```
context: {} (set of spacy.Spans = Entities of Entity Ruler)
None
```

Module contents

knowledge_extractor package
contains the knowledge_extractor module
This is the form of a docstring.
It can be spread over several lines.

medextractor.preprocessor package

Submodules

medextractor.preprocessor.preprocessor module

```
class medextractor.preprocessor.preprocessor.RuleBasedPreprocessor(doc_name)
    Bases: PreprocessingInterface
    get_preprocessed_text() → str
    pysbd_sentence_boundaries()
```

Module contents

medextractor.rdf package

Submodules

medextractor.rdf.RDFSerialiser module

```
class medextractor.rdf.RDFSerialiser.RDFSerialiser(knowledgebase, namespace_prefix)
    Bases: RDFSerialiserInterface
    knowledgebase_to_graph()
```

```
serialise_knowledgebase(output_path)
set_serialisation_format(serialisation_format)

medextractor.rdf.graphmanager module

class medextractor.rdf.graphmanager.GraphManager(namespace_prefix, namespace_uri)
    Bases: object
    add_disease(disease)
    add_symptom(disease, symptom)
    get_serialized_graph(output_path, serialization_format='pretty-xml')
```

Module contents

6.1.2 Submodules

6.1.3 medextractor.config_manager module

```
class medextractor.config_manager.ConfigManager
Bases: object
```

6.1.4 medextractor.manual_rdf_graph module

```
medextractor.manual_rdf_graph.add_symptom(symptom_str: str)
```

6.1.5 medextractor.medextractor module

6.1.6 medextractor.ruler creator module

```
class medextractor.ruler_creator.RulerCreator 
Bases: object \mathbf{load()} \mathbf{save()} \rightarrow \mathbf{None}
```

6.1.7 Module contents

Main Medextractor

This is the form of a docstring.

It can be spread over several lines.

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m

```
medextractor, 18
medextractor.config_manager, 18
medextractor.dummy, 13
medextractor.dummy.dummy, 13
medextractor.interfaces, 14
medextractor.interfaces.interfaces, 13
medextractor.knowledge, 16
medextractor.knowledge.base, 14
medextractor.knowledge.entity, 15
medextractor.knowledge.relations, 15
medextractor.knowledge.semantics, 15
medextractor.knowledge_extractor, 17
medextractor.knowledge_extractor.knowledge_extractor,
medextractor.manual_rdf_graph, 18
medextractor.preprocessor, 17
medextractor.preprocessor.preprocessor, 17
medextractor.rdf, 18
medextractor.rdf.graphmanager, 18
medextractor.rdf.RDFSerialiser, 17
medextractor.ruler_creator, 18
```

22 Python Module Index

INDEX

A add_disease() (medextrac- tor.rdf.graphmanager.GraphManager method), 18 add_relation() (medextrac- tor.knowledge.base.KnowledgeBase method), 14 add_symptom() (in module medextrac-	export_for_entity_linker() (medextrac- tor.knowledge.base.KnowledgeBase method), 14 export_for_entity_linker() (medextrac- tor.knowledge_extractor.knowledge_extractor.KnowledgeExtracto method), 16 G
tor.manual_rdf_graph), 18	<pre>get_knowledge_base()</pre>
add_symptom() (medextrac-	tor.dummy.dummy.DummyKnowledgeExtractor
tor.rdf.graphmanager.GraphManager method),	method), 13
18	<pre>get_knowledge_base()</pre>
<pre>add_training_example_to_relation() (medextrac-</pre>	tor.interfaces.interfaces.KnowledgeExtractorInterface
tor.knowledge.base.KnowledgeBase method),	method), 13 get_knowledge_base() (medextrac-
14 add_training_sample() (medextrac-	tor.knowledge_extractor.knowledge_extractor.KnowledgeExtracto
tor.knowledge.semantics.SemanticRelation	method), 16
method), 15	get_preprocessed_text() (medextrac-
	tor.dummy.dummy.DummyPreprocessor
C	method), 13
call2() (medextractor.knowledge_extractor.knowledge_e.method), 16	tor.tnierjaces.tnierjaces.FreprocessingInierjace
ConfigManager (class in medextrac-	method), 14
tor.config_manager), 18	get_preprocessed_text() (medextrac-
<pre>contains_training_sample() (medextrac- tor.knowledge.semantics.SemanticRelation</pre>	tor.preprocessor.preprocessor.RuleBasedPreprocessor method), 17
method), 15	get_serialized_graph() (medextrac-
D	tor.rdf.graphmanager.GraphManager method), 18
-	give_entities() (medextrac-
DISEASE (medextractor.knowledge.entity.EntityType attribute), 15	tor.knowledge.base.KnowledgeBase method), 14
DummyKnowledgeExtractor (class in medextrac-	GraphManager (class in medextrac-
tor.dummy.dummy), 13 DummyPreprocessor (class in medextrac-	tor.rdf.graphmanager), 18
tor.dummy.dummy), 13	
DummyRDFSerialiser (class in medextrac-	Н
tor.dummy.dummy), 13	has_relation() (medextrac-
_	tor.knowledge.base.KnowledgeBase method),
E	14 HAS_SYMPTOM (medextrac-
Entity (class in medextractor.knowledge.entity), 15 EntityType (class in medextractor.knowledge.entity), 15	tor.knowledge.relations.RelationType attribute), 15

I	module, 17		
IS_SYMPTOM_OF (medextrac-	medextractor.preprocessor.preprocessor		
tor.knowledge.relations.RelationType attribute),	module, 17		
15	medextractor.rdf		
	module, 18		
K	medextractor.rdf.graphmanager		
KnowledgeBase (class in medextractor.knowledge.base),	module, 18		
14	medextractor.rdf.RDFSerialiser		
knowledgebase_to_graph() (medextrac-	module, 17		
tor.rdf.RDFSerialiser.RDFSerialiser method),	medextractor.ruler_creator		
17	module, 18		
KnowledgeExtractor (class in medextrac-	module		
tor.knowledge_extractor.knowledge_extractor),	medextractor, 18		
16	medextractor.config_manager,18		
KnowledgeExtractorInterface (class in medextrac-	medextractor.dummy, 13		
tor.interfaces.interfaces), 13	medextractor.dummy.dummy,13		
<i>y</i> //	medextractor.interfaces, 14		
	medextractor.interfaces.interfaces, 13		
load() (medextractor.knowledge.base.KnowledgeBase	medextractor.knowledge, 16		
method), 14	medextractor.knowledge.base,14		
load() (medextractor.ruler_creator.RulerCreator	medextractor.knowledge.entity, 15		
method), 18	medextractor.knowledge.relations, 15		
mentou), 10	medextractor.knowledge.semantics, 15		
M	medextractor.knowledge_extractor,17		
medextractor	medextractor.knowledge_extractor.knowledge_extractor,		
module, 18	16		
medextractor.config_manager	medextractor.manual_rdf_graph, 18		
module, 18	medextractor preprocessor, 17		
medextractor.dummy	medextractor.preprocessor.preprocessor,		
module, 13	17		
medextractor.dummy.dummy	medextractor.rdf, 18		
module, 13	medextractor.rdf.graphmanager, 18		
medextractor.interfaces	medextractor.rdf.RDFSerialiser, 17		
module, 14	medextractor.ruler_creator, 18		
medextractor.interfaces.interfaces	N		
module, 13			
medextractor.knowledge	NO_RELATION (medextrac-		
module, 16	tor.knowledge.relations.RelationType attribute),		
medextractor.knowledge.base	15		
module, 14	P		
medextractor.knowledge.entity			
module, 15	PreprocessingInterface (class in medextrac-		
medextractor.knowledge.relations	tor.interfaces.interfaces), 13		
module, 15	process_texts() (medextrac-		
medextractor.knowledge.semantics	tor.knowledge_extractor.knowledge_extractor.KnowledgeExtracto		
module, 15	method), 16		
medextractor.knowledge_extractor	<pre>pysbd_sentence_boundaries()</pre>		
module, 17	tor.preprocessor.preprocessor.RuleBasedPreprocessor		
medextractor.knowledge_extractor.knowledge_ex	method), 17		
module, 16	R		
medextractor.manual_rdf_graph			
module, 18	RDFSerialiser (class in medextrac-		
medextractor.preprocessor	tor.rdf.RDFSerialiser), 17		

24 Index

```
RDFSerialiserInterface (class
                                                                                                                 medextrac-
                        tor.interfaces.interfaces), 14
RelationType
                                                        (class
                                                                                                                 medextrac-
                        tor.knowledge.relations), 15
RuleBasedPreprocessor
                                                                            (class
                                                                                                                 medextrac-
                        tor.preprocessor.preprocessor), 17
RulerCreator (class in medextractor.ruler creator), 18
S
save()
                         (medextractor.knowledge.base.KnowledgeBase
                        method), 14
                                     (medextractor.ruler\_creator.RulerCreator
save()
                        method), 18
\verb+saveKB()+ (medextractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extractor.knowledge\_extracto
                        method), 16
SemanticRelation
                                                                                                                 medextrac-
                                                                 (class
                                                                                              in
                        tor.knowledge.semantics), 15
serialise_knowledgebase()
                                                                                                               (medextrac-
                        tor.dummy.dummy.DummyRDFSerialiser
                       method), 13
serialise_knowledgebase()
                                                                                                               (medextrac-
                        tor.interfaces.interfaces.RDFSerialiserInterface
                       method), 14
serialise_knowledgebase()
                        tor.rdf.RDFSerialiser.RDFSerialiser method),
                        17
set_context()
                                                                                                               (medextrac-
                       tor.knowledge_extractor.knowledge_extractor.KnowledgeExtractor
                       method), 17
set_serialisation_format()
                                                                                                               (medextrac-
                       tor.interfaces.interfaces.RDFS erialiserInterface
                       method), 14
set_serialisation_format()
                                                                                                               (medextrac-
                       tor.rdf.RDFSerialiser.RDFSerialiser method),
SYMPTOM (medextractor.knowledge.entity.EntityType at-
                        tribute), 15
U
UNDEFINED
                                    (medextractor.knowledge.entity.EntityType
                       attribute), 15
```

Index 25