



H2020-SC1-DTH-2018-2020

Type of Action: RIA Research and Innovation action

Topic: **Adaptive smart working and living environments supporting active and healthy ageing**

Grant Agreement no: 826266

Deliverable

# D5.5 Open Source Release of Knowledge Base System



**COADAPT**

Start date of the project: December 1, 2018

Duration: 42 months

Project funded by the European Commission within the Horizon 2020 programme for research, technological development and demonstration		
Dissemination Level		
PU	Public, fully open	<b>x</b>
CO	Confidential, restricted under conditions set out in Model Grant Agreement	<input type="checkbox"/>
CL	Classified	<input type="checkbox"/>



## Notices

For information, please contact the project coordinator, Prof Giulio Jacucci, e-mail [giulio.jacucci@helsinki.fi](mailto:giulio.jacucci@helsinki.fi)

This document is intended to fulfil the contractual obligations of the CO-ADAPT project, which has received funding from the European Union's Horizon 2020 Programme, concerning deliverable D5.5 described in contract 826266.

All intellectual property rights are owned by CO-ADAPT consortium and are protected by the applicable laws. Except where otherwise specified, all document contents are: “©CO-ADAPT Project - All rights reserved”.

## Table of Revisions

Version	Date	Description and reason	Author	Affected sections
v0.1	11 Nov 2020	Draft	UH, ETSH	ALL
V1.0	30 Nov 2020	1st version	UH, ETSH	ALL

## Partners

- 1 HELSINGIN YLIOPISTO (UH)
- 2 TYÖTERVEYSLAITOS (FIOH)
- 3 INNOVATION SPRINT (INNO)
- 4 UNIVERSITA DEGLI STUDI DI TRENTO (UNITN)
- 5 UNIVERSITA DEGLI STUDI DI PADOVA (UNIPD)
- 6 IDEGO SRL (IDEGO)
- 7 BNP SRL (BNP)
- 8 AALTO KORKEAKOULUSAATIO SR (AALTO)
- 9 ETSIMO HEALTHCARE OY (ETSH)
- 10 ELECTROLUX ITALIA SPA (ELUX)

## Author(s)

Tung Vuong (UH), Thomas Grandell (ETSH)

## Reviewer(s)

- Giulio Jacucci (UH)

# 1 Executive Summary

The COADAPT Project releases code of knowledge base (KB) as open source release software under the Apache License version 2.0. The goal of the release is to facilitate researchers to integrate the KB with: 1) conversational agent to recommend most probable behavioral change programs; 2) electronic medical records analysis system to help therapists with symptom recommendations.

This document accompanies the deliverable D5.5, Open source release of knowledge base implementation. The actual deliverable is the open source code of the model, public API, and its documentation.

## 2 Open Source Release

The knowledge base system is a HTTP-based REST API that provides access to the pre-trained topic model. The API is implemented using Nodejs and the source is released in [https://github.com/tungvuong/Coadapt\\_CA](https://github.com/tungvuong/Coadapt_CA).


The screenshot displays the GitHub repository page for `tungvuong/Coadapt_CA`. The repository is private and has 1 star, 0 forks, and 0 watchers. The main branch is selected, showing 1 branch and 0 tags. The file list includes:

File	Commit Message	Time
build	server	3 months ago
node_modules	num_recs parameter added	2 months ago
src/usingCA/controllers	num_recs parameter added	2 months ago
.babelrc	server	3 months ago
LICENSE	Initial commit	3 months ago
README.md	Update README.md	now
package-lock.json	1st version	2 months ago
package.json	1st version	2 months ago
server.js	remove comments in js files	2 months ago

The README.md content is as follows:

**Knowledge Base (KB) of symptoms.**

The Knowledge Base (KB) system is a HTTP-based REST API that provides access to the pre-trained topic model. KB implementation is part of CO-ADAPT project. We hope the KB can be useful, adapted for implementation of conversational agent or electronic medical records analysis system to help therapists with symptom recommendations.



**COADAPT**

Installation instruction:  
npm install

The right sidebar shows the repository's metadata: About (CO-ADAPT project, Readme, MIT License), Releases (No releases published, Create a new release), Packages (No packages published, Publish your first package), and Languages (Python 55.8%, JavaScript 44.2%).

Installation guide is included in the repository:

- Download the package from github.
- Navigate to the downloaded folder in the command prompt (window OS) or terminal (Mac OSX, Linux).
- Run “npm install”
- Run “npm run dev-start” to deploy REST API service.

The API is composed of the three main files which are described in the following sections.

## server.js

This file contains code for running REST API via HTTPS.

Make sure the private key and certificate are properly installed on the deployed server:

```
var privateKey = fs.readFileSync('/etc/ssl/key.pem', 'utf8');
var certificate = fs.readFileSync('/etc/ssl/cert.pem', 'utf8');
```

## ca.js

This file contains code for communication with the CA intelligence. Make sure python3 is installed for the topic modeling to work.

```
var process = spawn('python3', ['./src/usingCA/controllers/topicmodel/app.py',
req.body.content, req.body.num_recs]);
```

The API accepts all kinds of request methods (GET, PUT, POST, DELETE, PATCH) as below:

```
res.header("Access-Control-Allow-Methods", "GET, PUT, POST, DELETE, PATCH");
```

## app.py

This file contains code for

1. Loading the pre-trained topic model,
2. Loading the program change recommendations,
3. Pre-processing user input,
4. Ranking change program recommendations

DataLoader class is used to load the pre-trained topic model and the recommendations:

```
class DataLoader:
    def __init__(self, dir):
        self.id2word = corpora.Dictionary.load(dir+'/dict1')
        self.corpus = corpora.MmCorpus(dir+'/corpus1.mm')
        self.lda_model = gensim.models.ldamodel.LdaModel.load(dir+'/lda1.model')
        self.num_topic = 100
        self.max_num_recs = 50
```

```
with open(dir+"/recommendations.pickle", "rb") as i_file:
    self.recs = pickle.load(i_file)
```

The system preprocesses the user input as below:

```
preprocessed_input = lemmatization(remove_stopwords(list(sent_to_words([u_input])))
```

For ranking the change program recommendations, the system first encode the user input into a topic distribution as below:

```
doc_vec = lda_model[ans[0]][0]
```

The resulted topic distribution is used for ranking the recommendations

```
for rec in recs:
    for i,z in enumerate(rec[0]):
        response.append([rec[1], ans_vec[i]*z[1]])
response = sorted(response, key=takeSecond, reverse=True)
```

The system outputs the ranked recommendations as a response



### 3 Public API

Alternatively, the API is also publicly available in ["https://reknowdesktopsurveillance.hiit.fi/predict"](https://reknowdesktopsurveillance.hiit.fi/predict), and provides developers with programmatic access to the web service.

The API takes an input as text and outputs top ranked change program recommendations. All communication made to the API is encrypted (i.e. only HTTPS is allowed). The API service only accepts the input in JSON format as below:

```
{
  "participantID": "123",
  "description_Problems": "I'm taking more naps, I skip breakfast, i can not do things i
  enjoy to de-stress, as that stressed feeling is still there",
  "num_recs": 10
}
```

*The parameter "num\_recs" is optional and the default value is 5.*

The API will reply with top-5 program change recommendations:

```
{
  "change_programs": { "<recommendation 1>": <probability1>,
    "<recommendation 2>": <probability2>,
    "<recommendation 3>": <probability3>,
    "<recommendation 4>": <probability4>,
    "<recommendation 5>": <probability5> }
}
```