Thomas Moser, Michael Heil

July 24, 2020

Introduction

The goal of this task is to work with simplified prewave data to extract specific information from text. The terms we want to extract we call query terms in the following. The object we want to extract from are called alerts.

For this task we supply two REST apis returning json objects which can be accessed via a GET request using your personal api key to obtain the relevant data. You can query these apis without any limitation.

https://services.prewave.ai/adminInterface/api/testQueryTerm?key=< your-key > Returns a json encoded list of query terms we want to extract. An example of this data can be found on the next page. The fields are described in more detail in the following:

id Unique id identifying the query term object

text The text we look for (multiple terms are separated by white spaces)

language Two letter code denoting the language of the query term text

keepOrder If true then the multiple parts (space separated) of the term have to appear consecutively without any separating tokens. Otherwise the parts can occur in any order and in any quantity, anywhere in the text.

This api is stable and its output will not change over time.

https://services.prewave.ai/adminInterface/api/testAlerts?key=< your-key > Returns a sequence of alert objects which structure can be easily deduced from some example data (see next page). None of the fields are nullable. The id should be unique and contents consists of an array of content objects.

Problem

Write a program which can query the data, given by the api for testAlerts, and determines in which alert a query term occurs. The query terms you can obtain from the testQueryTerm api.

The final product is a list of the matches and needs to contain at least the alert id and the id of the extracted terms. The choice of the coding language and the data format of the output list is your choice. Do not write duplicate matches in that list.

Hints

Choose a language and libraries you feel comfortable with.

Some questions in this task are formulated in an open way. Do not spend too much time on perfecting it but rather make a clear decision and if necessary clarify your choice in a comment.

The data obtained via testAlerts api is changing with each request and consider testing your program on multiple api calls to it (around a 100 should suffice).

If you need some inspiration for libraries here are some ideas for various languages:

Scala For the reference solution we used circe as a json, sttp as a api client and better files as a file library.

Java We recommend to look int some json library like org.json. For a http client it suffices to use the httpcomponents from apache.

Python Look into requests and json packages.

Example Data

In the following you see an example alert:

```
{
        "id": "6gbujhu89786",
        "contents": [
                {
                        "text": "Wolfgang Lemb, ig metall Germany stands
                           in solidarity with #StrikeForBlackLives",
                        "type": "text",
                        "language": "de"
                }
        ],
        "date": "2020-07-24T06:18:45.777Z",
        "inputType": "tweet"
}
```

and an example query term:

```
{
        "id": 101,
        "text": "IG Metall",
        "language": "de",
        "keepOrder": true
}
```