keyword categorization

Leadspace home assignment

Author: moran vaisberg

Creation Date: 11-Jun-18

Status: Final

Contents

[Prerequisites 3](#_Toc434572203)

[Technical Design 4](#_Toc434572203)

[Solution Overview 5](#_Toc434572209)

[Test Cases 6](#_Toc434572210)

# prerequiSiteS

property.json is a system property file and should present in the installation folder (“C:\leadspace”).

property.json file contains the path to dictionary.json and categorisation.log files.

dictionary.json file contains the predefined dictionary

# technical Design

## keyword categorization Feature ID: 1

#### Description

* Implementation of java service using REST protocol. The service provides a keyword categorization functionality:

Given a phrase, the service will provides an analysis of the phrase and resolves the keywords present in the phrase using predefined dictionary.

* Expected result:

{

   "Vice President”: 0,

   "Sales”: 18,

   "Marketing”: 28

 }

* Example of dictionary configuration:

President

Vise President

Sales

Marketing

IT

CFO

CTO

Banking

eComerce

* Example of url: <http://localhost:8080/categorize?phrase=Vice+President+of+Sales+and+Marketing>

# Solution Overview

A phrase, provided as a query parameter in GET method of REST protocol, represents a collection of Strings. The requirement is to find the strings, which are in the dictionary and their position in the phrase. The response will be in JSON format.

An algorithm to find categories and the positions in the phrase:  
Step1: Pick a String from an input categories (a primary expression) and find a match or prefix in the dictionary categories

Match: e.g. category = 'Sales', dictionary = 'Sales'

Prefix: e.g. category = 'Vice', dictionary = 'Vice President'

Step2: If found a match, add the category and the position to a map

Step3: If found a prefix, concatenate next String to the prefix

Prefix: e.g. category = 'Vice', dictionary = 'Vice President', next String = 'President'

Concatenated String = 'Vice President'

Proceed to Step2

Step4: During Step2 and Step3, perform the same action for secondary expression.

Primary Expression can be a concatenation of several input categories,

While Secondary Expression is always single category.

e.g. Primary Expression = 'Vice Manager', Secondary Expression = "Manager"

dictionary = {'Vice President', 'Manager Department'}

In case Primary returns false, proceed with Secondary on the same loop iteration.

if match, proceed to Step2 and so on.

Step5: No Match - proceed to Step1 to pick next String

# test cases

1. Find a matches between phase sub strings and the dictionary

Dictionary: ["President","Vice President","Sales","Marketing", "IT", "CFO", "CTO",“Banking","eComerce"]  
url: <http://localhost:8080/categorize?phrase=President+CTO+Sales+and+Marketing>  
"expected":{  
 "Vice President": 0,  
 "Sales": 18,  
 "Marketing": 28  
 }

1. Find a matches between phase sub strings and the dictionary

Dictionary: ["President","Vice President","Sales","Marketing", "IT", "CFO", "CTO",“Banking","eComerce"]  
url: <http://localhost:8080/categorize?phrase=President+CTO+IT+and+Banking>  
"expected":{  
 "President": 0,

"CTO": 10,

"IT": 14,

"Banking": 21  
 }

1. Case when Vice matches only a prefix, but a match on Manager Department

Dictionary: ["Vice President","Manager Department","company"]

url:

<http://localhost:8080/categorize?phrase=Vice+Manager+Department+of+company>

"expected":{

"Manager Department": 5,

"company": 27

}

1. Test a phrase with no matching in the dictionary

Dictionary: ["President","Vice President","Sales","Marketing", "IT", "CFO", "CTO", "Banking","eComerce"]

url:

[http://localhost:8080/categorize?phrase=This+is+not+matching+phrase](http://localhost:8080/categorize?phrase=Vice+Manager+Department+of+company)

"expected":{

}