Entity Cohort Matching

Every day, our data scientists at Socure need to analyze and categorize a large number of entities to support effective model training. An ***entity*** is a real human with a set of associated attributes such as first name, last name, emails, age, addresses, country, etc. Data scientists define ***entity cohorts*** to categorize entities in different ways. An entity cohort is a group of entities with a set of pre-specified conditions.

Entities

There is a TSV (Tab-Separated Value) file containing a set of entities with their attributes:

eid	*first_name*	*last_name*	*age*	*country*	*zip_code*	*emails*
1	John	Lee	22	US	91003	[jlee@yahoo.com,johnl@aol.com,jl123@gmail.com]
2	Sam	Smith	50	US	92123	[ssmith@aol.com,sams@example.com]
3	Sandy	Jones	19	CA	94666	[sj22@gmail.com,sjones@comcast.com]
4	Lily	Chen	35	CH	310002	[lcc12@hotmail.com,lc12345@qq.com]
5	Tom	Tan	81	CH	349999	[]

In the above Entities example,

- Fields are tab-separated.
- Each line represents an entity, which has an unique ID as specified in the field ***eid***.
- The field "emails" can be a list of email addresses (separated by comma) belonging to the entity. If no emails, the entry is "[]"

Entity Cohorts

Entity Cohorts are defined as below examples:

cohort:1	last_name:Chen	age:[10,50]	country:US
cohort:2	age:(15,45]	country:CH	emails:hotmail.com
cohort:3	first_name:John	zip_code:91003	
cohort:4	country:US	emails:gmail.com	

In the above Entity Cohorts example:

- Fields are tab-separated.
- The first field **cohort** is the unique ID for that particular cohort (e.g., cohort:1).

- We can define one or multiple conditions for each cohort, with the format: ***field_name:condition***.
    - first_name: exact match of the entity's first name.
    - last_name: exact match of the entity's last name.
    - age: match of the age range. "(" or ")" indicates that it is exclusive, while "[" or "]" indicates that it is inclusive. (e.g., [10,50] means within the range of age 10 and 50, inclusive, while (10,50] means older than 10, but younger than 50, including 50.)
    - country: exact match of the entity's country
    - zip_code: exact match of the entity's zip code.
    - emails: The email domain (e.g, yahoo.com) matches any emails that the entity owns.
- An entity is said to match a cohort, if ALL of its conditions are met.
- Depending on definitions of entity cohorts, an entity could match one or multiple or none cohorts.
- For example,
    - Lily Chen (eid:4) as mentioned in the above example can match cohort:2.
    - John Lee (eid:1) can match both cohort:3 and cohort:4.

Problem Statement

You are asked to implement a program that can provide below functions:

1. ***void init(String entityFilepath, String entityCohortFilepath)*** *//This is to pass two TSV files into the program.*
2. ***List<String> findEntityCohorts(int eid)*** *//Input an entity ID (eid), find which cohorts it can match to. Keep in mind an entity could match to multiple cohorts. Return all matched cohort IDs.*
3. ***boolean addEntityCohort(String cohort)*** *//Add a new cohort (fields delimited by TAB) or change/overwrite an existing cohort's rule. e.g,*
    a. *INPUT: "cohort:5 last_name:Jackson age:(18,26)",*
    b. *RETURN: true (if successful)*

Requirements

1. Programming Language: Scala, Java or Python. If you want to use other languages, please reach out to jainik@socure.com for discussion.
    a. We have provided a code skeleton below using JAVA for your reference.
2. Please submit a README file:
    a. A brief summary of your approaches to solve the problem.
    b. Steps on how to build and run your program, which should follow Input and Output as described above.
    c. Steps on how to test your program with your provided test cases.
3. Submit a tarball including all your source files, test cases and a README file. Please make sure you have tested with your own test cases.

4. Send the final submission through email.
5. On average, you should be able to finish this exercise within **~3 hours**. However, we value more of a submission with high quality/efficient codes and comprehensive test cases. Feel free to use extra time if you need to deliver us high quality codes. We do expect you to submit your codes within the same day.
6. We will evaluate your submissions by correctness, coding styles, readability and efficiency. Make sure you add enough comments to your codes to help us understand your logic.
7. If you have additional questions, please reach out to jainik@socure.com

Code Skeleton (JAVA)

```java
import java.util.*;

public class EntityCohortMatch {
    private String entityFilepath;
    private String entityCohortFilepath;
    // other variables to add if needed...

    public EntityCohortMatch() {
    }

    public void init(String entityFilepath, String entityCohortFilepath) {
            //Add your codes...
            this.entityFilepath = entityFilepath;
            this.entityCohortFilepath = entityCohortFilepath;
    }

    public List<String> findEntityCohorts(int eid) {
            //Add your codes...
            return new ArrayList<>();
    }

    public boolean addEntityCohort(String cohort) {
            //Add your codes...
            return true;
    }

    private static void test(Service s) {
        System.out.println(s.findEntityCohorts("eid1"));  // call API 2
        System.out.println(s.findEntityCohorts("eid2"));  // call API 2
        System.out.println(s.addEntityCohort("..."));  // call API 3
        System.out.println(s.findEntityCohorts("user_1"));  // call API 2
        System.out.println("Pass!");
    }

    public static void main(String[] args) {
        EntityCohortMatch s = new EntityCohortMatch();
        s.init("entity.tsv", "cohort.tsv");  // call API 1
        test(s);
    }
}
```