######################### README #########################

1. Siyang Wu (UNI: sw2848)

Hao-Hsin Shih (UNI: hs2762)

2. List of files:

\_\_init\_\_.py

Project2.py (This is the main python that users should run)

Info\_box\_group.py

info\_box\_person.py

setting.py

type\_check.py

run.sh

3. To run the program. You have two options:

**(Key)**

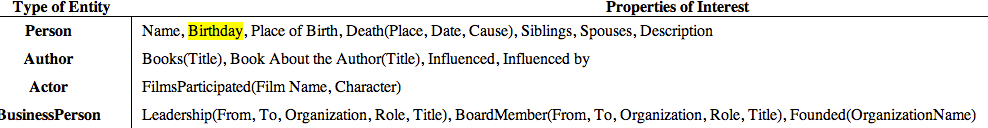


**/home/gravano/6111/Html/Proj2/code/run.sh -key <Freebase API key> -q <query> -t <infobox|question>**



**/home/gravano/6111/Html/Proj2/code/run.sh -key <Freebase API key> -f <file of queries> -t <infobox|question>**

4. The table for mapping:



For person: (Person, Actor, Author, Business person)

|  |  |  |
| --- | --- | --- |
| **Freebase Property** | **Property of interest** | **Comments** |
| Person.name | /type/object/name | I used only the first entry from the array of values provided by freebase. |
| Person.Birthday | '/people/person/date\_of\_birth' | I used only the first entry from the array of values provided by freebase. |
| Person.Place of birth | '/people/person/place\_of\_birth' | I used only the first entry from the array of values provided by freebase. |
| Person.death  (optional) | '/people/deceased\_person/date\_of\_death'  '/people/deceased\_person/place\_of\_death'  '/people/deceased\_person/cause\_of\_death' | I used only the first entry from the array of values provided by freebase. |
| Person.sibling  (optional) | ['/people/person/sibling\_s']['values'] | I used all the entries from the array of values provided by freebase because some people have many siblings |
| Person.spouses  (optional) | ['/people/person/spouse\_s']['values']  inside the dictionary: (for every spo)  spo['property']['/people/marriage/from']  spo['property']['/people/marriage/from'] | the array of values provided by freebase because some people have many spouses and they are different “from” and “to” |
| Person.description | '/common/topic/description' | I used only the first entry from the array of values provided by freebase. |
| Author.books | ['/book/author/works\_written']['values'] | I used all the entries from the array of values provided by freebase because author may have many books. |
| Author.books about the author | ['/book/book\_subject/works']['values'] | I used only the first entry from the array of values provided by freebase. |
| Author.influenced | ['/influence/influence\_node/influenced']['values'] | I used all the entries from the array of values provided by freebase because author may influence many people |
| Author.influenced\_by | ['/influence/influence\_node/influenced\_by']['values'] | I used all the entries from the array of values provided by freebase because author may be influenced by many people |
| Actor.films participated | ['/film/actor/film']['values']  inside the dictionary:  ['/film/performance/character']  ['/film/performance/film'] | I used all the entries from the array of values provided by freebase because actors may have many films and corresponding characters |
| Business\_person.Founded | ['/organization/organization\_founder/organizations\_founded']['values'] | I used all the entries from the array of values provided by freebase because Business people may have founded many companies |
| Business\_person.Leadership | ['/business/board\_member/leader\_of']['values']  inside the dictionary:  organization: '/organization/leadership/organization'  role: '/organization/leadership/role'  title: '/organization/leadership/title'  from: '/organization/leadership/from'  to: '/organization/leadership/to' | I used all the entries from the array of values provided by freebase because Business people may be the leaders of many organizaitions |
| Business\_person.Boardmember | ['/business/board\_member/organization\_board\_memberships']['values']  inside the dictionary:  organization: '/organization/organization\_board\_membership/organization'  role: '/organization/ organization\_board\_membership /role'  title: '/organization/ organization\_board\_membership /title'  from: '/organization/ organization\_board\_membership /from'  to: '/organization/ organization\_board\_membership /to' | I used all the entries from the array of values provided by freebase because Business people may have been board members in many organizations |

For groups: (League, Sports Team)

|  |  |  |
| --- | --- | --- |
| **Freebase Property** | **Property of interest** | **Comments** |
| League.name | ‘/type/object/name’ | Both League and Sports Team have name property, so the tag is the same |
| League.championship  (optional) | "/sports/sports\_league/championship" | I used only the first entry from the array of values provided by freebase. |
| League.teams | "/sports/sports\_league/teams" | I used all the entries from the array of values provided by freebase. Because there are many teams in a league.  To extract the actual value I used the "value" tag (string format) from the entry. |
| League.sport | "/sports/sports\_league/sport" | I used only the first entry from the array of values provided by freebase.  To extract the actual value I used the "value" tag (string format) from the entry. |
| League.slogan | "/organization/organization/slogan" | I used only the first entry from the array of values provided by freebase.  To extract the actual value I used the "value" tag (string format) from the entry. |
| League.Description | /people/person/place\_of\_birth | I used only the first entry from the array of values provided by freebase, as it does not make sense for people to have multiple dates of birth.  To extract the actual value I used the "value" tag (string format) from the entry. |
| League.officialWebsite | "/common/topic/official\_website" ["values"] | I used all the entries from the array of values provided by freebase, as there may be many websites.  To extract the actual value I used the "value" tag (string format) from the entry. |
| Sports team.name | ‘/type/object/name’ | I used only the first entry from the array of values provided by freebase.  To extract the actual value I used the "value" tag (string format) from the entry. |
| Sports team.desctiption | Same as LEAGUE | I used only the first entry from the array of values provided by freebase.  To extract the actual value I used the "value" tag (string format) from the entry. |
| Sports team.championship  (optional) | "/sports/sports\_team/championships"[“values”] | I used all the entries from the array of values provided by freebase because a team may be gotten a lot of championships.  To extract the actual value I used the "value" tag (string format) from the entry. |
| Sports team.founded | "/sports/sports\_team/founded" | I used only the first entry from the array of values provided by freebase.  To extract the actual value I used the "value" tag (string format) from the entry. |
| Sports team.league | "/sports/sports\_team/league" | I used all the entries from the array of values provided by freebase.  Because a team may play in many leagues. |
| Sports team.locations | "/sports/sports\_team/location" | I used only one entry from the array of values provided by freebase. |
| Sports team.coach | "/sports/sports\_team/coaches”[“values”]  inside the dictionary:  "/sports/sports\_team\_coach\_tenure/coach" (name)  "/sports/sports\_team\_coach\_tenure/position"(position)  "/sports/sports\_team\_coach\_tenure/from" (from)  "/sports/sports\_team\_coach\_tenure/to " (to) | I used all the entries from the array of values provided by freebase. |
| Sports team.PlayerRoster | "/sports/sports\_team/roster"[“values”]  inside the dictionary:  "/sports/sports\_team\_roster/player" (name)  "/sports/sports\_team\_roster/position" (position)  "/sports/sports\_team\_roster/number" (number)  "/sports/sports\_team\_roster/from" (from)  "/sports/sports\_team\_roster/to " (to) | I used all the entries from the array of values provided by freebase. |

Additional information:

We classify all the types in “PERSON” and “not PERSON”. There is no overlap,

“PERSON” includes BusinessPerson, Actor and Author.

“not PERSON” includes League and SportsTeam.

Firstly we check whether this JSON file contains the ‘Person’, if it is, then we only output the attributes of PERSON. Otherwise it belongs to Group, then it should only output the attributes belonging to group.

For example, NFL is both LEAGUE and AUTHOR. But AUTHOR, BUSINESS\_PERSON, ACTOR all belong to the person kind, so they should not be printed because NFL is already a group type. Then it should be National Football League(LEAGUE).