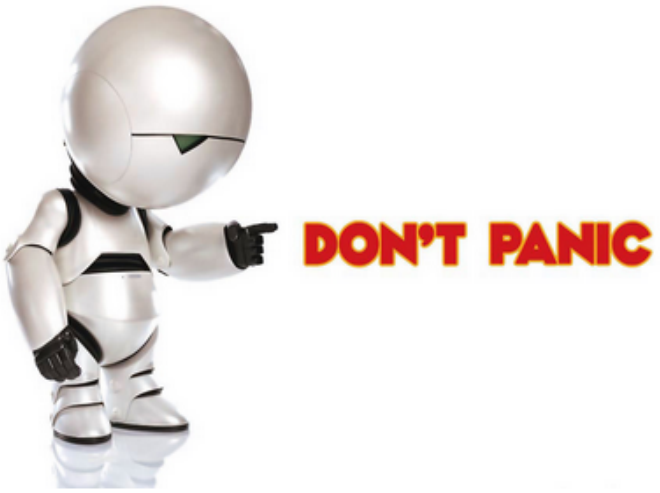
PensieroProfondo

Specifiche v1.1 del progetto di Metodologie di Programmazione 2014
Corso di Laurea Triennale in Informatica
Prof. Roberto Navigli

Il progetto consiste nello sviluppo di un piccol(issim)o sistema di Question Answering (QA) chiamato **PensieroProfondo**. Esempi avanzati di tali sistemi sono: START, IBM Watson e WolframAlpha.



Il sistema da progettare è, ovviamente, più semplice e consiste nei seguenti moduli: sistema di indicizzazione della base di conoscenza (Indexer), sistema di interrogazione (Searcher) e l'interfaccia in linguaggio naturale, ovvero il sistema di QA vero e proprio (PensieroProfondo).

La base di conoscenza

Per poter rispondere alle domande che gli verranno poste, il sistema ha bisogno di immagazzinare quante più informazioni possibili relativamente allo scibile umano. A tal fine, si è scelto di utilizzare una grande base di conoscenza collaborativa, Freebase (http://freebase.com). Gli elementi fondamentali nella base di conoscenza sono i **topic** (ovvero concetti e individui), **i tipi** dei topic e le **proprietà dei tipi**. Ad esempio, Homer Simpson ha il seguente id univoco: m.0h545 (sotto forma di URL si ottiene una pagina concatenando alla URL base http://www.freebase.com/ l'ID del topic sostituendo slash al punto, cioè: http://www.freebase.com/m/0h545 rappresenta Homer Simpson in HTTP)

Download e formato dei dati in input

Una versione di Freebase ridotta al dominio del cinema è scaricabile da:

http://babelware.org/pensieroprofondo/fb_triples_film.gz

Il file è strutturato come un elenco di relazioni (soggetto, predicato, oggetto) in formato N3 ovvero una tripla per ogni linea. Il file contiene solo topic collegati ai film, per un totale di 127 milioni di triple (17G scompattato e 1.7G compattato). Il numero di predicati unici è pari a 13258 (eventualmente scaribili da http://babelware.org/pensieroprofondo/fb_reltypes_film.gz) ed il numero di topic è 4555082 (http://babelware.org/pensieroprofondo/entityFilmUniq.gz).
Per la lettura del file delle triple di Freebase si consiglia di leggere direttamente il file .gz senza scompattarlo utilizzando la classe GZIPInputStream:

```
FileInputStream fin = new FileInputStream(FILENAME);
GZIPInputStream gzis = new GZIPInputStream(fin);
InputStreamReader isr = new InputStreamReader(gzis);
BufferedReader br = new BufferedReader(isr);
```

Esempi di predicati che coinvolgono i topic come soggetti

Ad esempio, un topic è m.0h545, che rappresenta Homer Simpson. Ecco alcune triple che lo riguardano:

```
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "Homer Simpson"@et  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "Homer Simpson"@pt  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "Homer Simpson"@cs  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "Homer Simpson"@da  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "Houmeris Simpsonas"@lt .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/2000/01/rdf-schema#label>  "호머 심슨"@ko  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/common.topic>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/tv.tv_character>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/fictional_universe.fictional_character>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/award.award_nominee>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/theater.theater_character>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/base.thesimpsons.topic>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/base.breakfast.breakfast_cereal_mascot>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/base.italiantv.adapted_tv_character>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/film.film_character>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/base.breakfast.topic>  .
<http://rdf.freebase.com/ns/m.0h545>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://rdf.freebase.com/ns/base.ontologies.ontology_instance>  .
```

La prima riga si traduce in label(m.0h545, “Homer Simpson”@et), ovvero che m.0h545 si dice in Estone “Homer Simpson” (ma va?). L’ottava riga, invece, specifica type(m.0h545, tv.tv_character), ovvero che Homer è di tipo tv.tv_character, così come, nelle righe seguenti, che è di tipo fictional_universe.fictional_character, film.film_character ecc.

Esempi di predicati che coinvolgono i tipi come soggetti

I tipi sono normalmente specificati da stringhe separate da punto, ad esempio tv.tv_character oppure base.thesimpsons.topic, theater.theater_character. Anche i tipi possono avere un tipo a loro volta, ad es.:

```
<http://rdf.freebase.com/ns/award.award_winner> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2000/01/rdf-schema#Class>
```

Moduli del progetto

Il progetto consta di 3 parti, ciascuna implementata in un package separato:

- it.uniroma1.lcl.pensieroprofondo.index
- it.uniroma1.lcl.pensieroprofondo.search
- it.uniroma1.lcl.pensieroprofondo.qa

Il package it.uniroma1.lcl , invece, dovrà contenere le classi Type, Topic (e, se implementate, le classi Language e Graph).

Indicizzatore (10 punti)

Questo modulo legge il file delle triple di Freebase e crea uno o più indici atti a contenere le informazioni della base di conoscenza, in particolare i topic, i tipi dei topic e le proprietà dei tipi.

Al fine dell'indicizzazione si consiglia, ma non si richiede necessariamente, l'utilizzo della libreria Apache Lucene. Il punto di accesso di questo modulo è la classe FreebaseIndexer che possiede il seguente costruttore:

public **FreebaseIndexer**(String nomeFileGz, String cartellaDestinazione)

e il metodo:

public void **index**()

che procede all'indicizzazione, ovvero alla creazione nella cartella di destinazione specificata della struttura dati su disco da accedere in seguito (vedi sotto).
La classe implementa anche la possibilità di restringere l'indicizzazione a un elenco di identificativi di topic e/o un elenco di identificativi tipi specificati.

Nel progetto, **deve** esistere la classe eseguibile it.uniroma1.lcl.pensieroprofondo.index.Indexer. All'esecuzione di

```
java -cp lucene.jar:matricola.cognome.jar it.uniroma1.lcl.pensieroprofondo.index.Indexer nomeFileTriple.gz
```

la classe **deve** leggere le triple freebase da nomeFileTriple.gz, e deve generare i file di indice (da posizionare nella directory corrente). Per testare il programma si userà una versione ridotta del file di triple di freebase.
Il file lucene.jar conterrà i class files di Lucene 4.8.1 (che potranno essere utilizzati dal vostro progetto).

Searcher (10 punti)

Questo modulo legge l'indice o gli indici creati su disco per rispondere a singole interrogazioni. La classe Searcher è il punto di accesso. Di tale classe può esistere una sola istanza. La classe espone i seguenti metodi:

- **getTopic** che, dato in input un id, restituisce un oggetto della classe **Topic** che permette di accedere a tutte le informazioni del concetto o individuo specificato. Ad esempio:

  ```
  Searcher s = Searcher.getInstance();
  Topic t = s.getTopic("m.0h545");
  ```

  restituisce il topic relativo a Homer Simpson.

- **getTopics** che, dato in input un tipo, restituisce una collezione iterabile di **Topic.** Ad esempio:

  ```
  Type ty = new Type("film.film_character");
  for (Topic t : s.getTopics(ty))
      for (String label : t.getLabels())
          System.out.println(t.getId()+" ["+label+"]: "+t.getDescription());
  ```

  stampa l'elenco dei personaggi di film nel formato "id [nome]: descrizione", uno per ogni riga. Sia i tipi che i topic possono essere specificati senza il prefisso dell'URL (ad es. http://www.w3.org/1999/02/ e http://rdf.freebase.com/ns/ possono essere omessi, risparmiando così molto spazio nell'indice).

  Topic.getLabels restituisce una collezione di etichette associata al topic (ovvero, le parole utilizzate per esprimere il topic) per la lingua di default del vostro progetto **(inglese)**. **+2 punti** se si implementa il metodo **Topic.getLabels(Language)** che restituisce le etichette del topic per la lingua specificata (utilizzare tutte le lingue del dump scaricato).

- **getTypes** che, dato un topic, restituisce una collezione dei tipi associati a quel **Topic**.
- **getQuery** che, data in input un'interrogazione, restituisce una collezione di topic e/o tipi che rispondono all'interrogazione. Le interrogazioni sono oggetti che esprimono i seguenti tipi di vincoli (anche in congiunzione e/o disgiunzione):
  - L'oggetto in questione è un tipo, un topic o qualsiasi dei due
  - L'oggetto deve possedere una certa proprietà valorizzata (ad es. un topic deve avere tipo film.film_character oppure deve avere come attrice Nicole Kidman) o meno (ad es. l'oggetto deve avere un attore).

Obbligatorio per gruppi da due: La classe espone anche un metodo **getGraph** che, dato in input un insieme di Topic e/o Type, restituisce il grafo indotto da tali elementi.

Nel progetto, deve esistere la classe it.uniroma1.lcl.pensieroprofondo.search.Searcher

che espone i metodi di cui sopra. **Questa classe deve leggere automaticamente i file di indice generati dall'Indexer** (sia l'Indexer che il Searcher saranno eseguiti nella stessa directory - quindi i file di indice devono essere scritti e letti nella directory corrente al tempo di esecuzione). Questo significa che non dovrà essere configurato alcun file a mano per poter eseguire in eseguenza in modo corretto Indexer e Searcher.

Se si implementa la classe it.uniroma1.lcl.Language, questa deve avere un costrutore Language(String) che, presa in input una stringa rappresentate una lingua (cioè una stringa come "@en", "@it", "@ko", ecc.), restituisce un oggetto Language relativo a quella lingua.

Question Answering (10 punti)

Il terzo modulo è l'interfaccia di Question Answering verso l'utente. Esso rende disponibile una classe **PensieroProfondo** che espone metodi per due modalità di interazione:

- il metodo **run** entra in un ciclo infinito: richiede un input da tastiera, ovvero la domanda da sottoporre al sistema, la elabora, restituisce la risposta all'utente e torna alla richiesta successiva.
- il metodo **query** che, data in input una singola domanda sotto forma di stringa, restituisce la o le risposte a tale domanda.

Il tipo di domande che possono essere specificate è aperto. Inizialmente il sistema permette i seguenti tipi di domande:

- Che cosa è X? (ad es. "what is Matrix?", "what is Lucasfilm?")
- Chi è X? (ad es. "who is Homer Simpson?")
- Dove è nato X?
    - **Extra (+2 punti):** se il topic richiesto non è rispettivamente un'entità non vivente ("what is X?") o una persona ("who is X?", "where was X born?"), PensieroProfondo risponde adeguatamente.
- **Obbligatorio per gruppi da 2:** Chi ha **azione** X? (ad es. "who has directed Matrix?") dove **azione** ("girato") è un'azione associata a un determinata proprietà (ad es. "girato" = "film.film.directed_by")
- **Obbligatorio per gruppi da 2:** Dimmi **X tipo** (ad esempio, "tell me 10 cities") dove **X** è un numero e **tipo** è un tipo (ad es. "tell me 10 cities", dove cities è associato al tipo "location.citytown"; oppure, "tell me 10 movies", dove movies è associato al tipo "film.film".
- **Obbligatorio per gruppi da 2:** Dimmi **Y** di **X** dove **X** è un topic e **Y** è una stringa mappata a un predicato (ad esempio, "tell me the genre of Pink Floyd – The Wall", dove "genre" è mappato al predicato "film.film_genre" e "Pink Floyd – The Wall" è un topic)

Obbligatorio per gruppi da 2: Il sistema può essere espanso aggiungendo nuovi tipi di domande, utilizzando il meccanismo della reflection.

Tutti gli studenti devono comunque gestire l'impossibilità di rispondere; ad esempio, se **Y** non è un predicato, PensieroProfondo deve dire di non conoscere quell'informazione. Le risposte sono fornite dal metodo **query** come elenco dei topic o tipi che rispondono alla domanda, mentre il metodo **run** restituisce le etichette (.**getLabels**()) di ciascun risultato (topic o tipo) all'utente.

Esempi

Seguono esempi di triple che rispondono alle domande specificate:

Q: tell me the genre of Pink Floyd – The Wall

La seguente tripla contiene una possibile risposta (m.04t36, che corrisponde a Musical, è un genere di m.058ymm che corrisponde a Pink Floyd – The Wall):

<http://rdf.freebase.com/ns/m.04t36> <http://rdf.freebase.com/ns/film.film_genre.films_in_this_genre> <http://rdf.freebase.com/ns/m.058ymm> .

Q: who has directed Pink Floyd – The Wall?

La seguente tripla contiene una possibile risposta (m.1ycck è Alan Parker, il regista):

<http://rdf.freebase.com/ns/m.058ymm> <http://rdf.freebase.com/ns/film.film.directed_by> <http://rdf.freebase.com/ns/m.01ycck> .

Q: who has written Pink Floyd – The Wall?

La seguente tripla contiene una possibile risposta (m.017g21 è Roger Waters):

<http://rdf.freebase.com/ns/m.058ymm> <http://rdf.freebase.com/ns/film.film.written_by> <http://rdf.freebase.com/ns/m.017g21> .

Q: tell me the actors of Pink Floyd – The Wall

La seguente tripla contiene una possibile risposta. In questo caso la risposta non è specificata direttamente nella tripla, perché il topic oggetto del predicato (m.0y58gz5) rappresenta una collezione di proprietà, ovvero nello specifico una film performance:

<http://rdf.freebase.com/ns/m.058ymm> <http://rdf.freebase.com/ns/film.film.starring> <http://rdf.freebase.com/ns/m.0y58gz5>

La collezione di proprietà in m.0y58gz5 è accessibile mediante altri predicati che mettono in relazione la film performance m.0y58gz5 con diversi topic. Ad esempio:

<http://rdf.freebase.com/ns/m.0y58gz5> <http://rdf.freebase.com/ns/film/performance/actor> <http://rdf.freebase.com/ns/m.0gczs7>
<http://rdf.freebase.com/ns/m.0y58gz5> <http://rdf.freebase.com/ns/film/performance/film> <http://rdf.freebase.com/ns/m.058ymm>
<http://rdf.freebase.com/ns/m.0y58gz5> <http://rdf.freebase.com/ns/film/performance/character> <http://rdf.freebase.com/ns/m.0y58gz8>
<http://rdf.freebase.com/ns/m.0y58gz5> <http://rdf.freebase.com/ns/type/object/type> <http://rdf.freebase.com/ns/film/performance>

I topic in questione sono: l'attrice Jenny Wright (m.0gczs7), il film Pink Floyd – The Wall (m.058ymm), il personaggio American Groupie (m.0y58gz8) e il tipo della collezione (film/performance).

Q: who is Jenny Wright?

PensieroProfondo sa che e' una persona:

<http://rdf.freebase.com/ns/people.person> <http://rdf.freebase.com/ns/type.type.instance> <http://rdf.freebase.com/ns/m.0gczs7> .

e che è una attrice:

<http://rdf.freebase.com/ns/film.actor> <http://rdf.freebase.com/ns/type.type.instance> <http://rdf.freebase.com/ns/m.0gczs7> .

Q: where was Jenny Wright born?

PensieroProfondo sa che è nata a New York:

<http://rdf.freebase.com/ns/m.02_286> <http://rdf.freebase.com/ns/location.location.people_born_here> <http://rdf.freebase.com/ns/m.0gczs7> .

Valutazione

Saranno oggetto di valutazione:

- Progettazione secondo il paradigma di progettazione orientato agli oggetti
- Facilità nell'estensibilità (ad esempio, nuovi tipi di interrogazioni nel secondo modulo, nuovi tipi di domande nel terzo)
- Pulizia del codice
- Documentazione del codice in javadoc
- Facilità di utilizzo delle API

È possibile consegnare anche solo i primi due moduli, puntando al punteggio massimo di 20.
È molto importante posizionare le classi nei package richiesti, dare il nome richiesto ai metodi, e fare in modo che le classi eseguibili interagiscano con l'utente come richiesto.
È molto importante presentare un unico file jar contenente: i file sorgenti, i file class, ed i file html del javadoc.

La consegna del file jar deve essere effettuata alla seguente URL:

http://151.100.179.33/metodologie2014/

Plagio

Nel caso di plagio accertato, dal Web o da colleghi di corso, **non sarà consentito di consegnare nuovamente il progetto** in questo A.A., il che implicherà dover sostenere nuovamente anche la prova scritta nell'anno seguente. **Anche chi ha permesso il plagio (per esempio fornendo il proprio codice) potrebbe subire conseguenze.**