Project Plan

Knowledge Graphs(KG01)

Entity Search

SUPERVISOR: DIEGO MOUSSALLEM

Submitted By:
Aman Sharma
Barun Kumar
Saud Afaq
Taslima Akter
Paaras Baru

DATE: 06-April-2020

Contents

Contents

0.1 OVERVIEW

The project plan comprises of several phrases, where we have been asked to create an NLP pipeline to parse the entity description generation and implement ranking functions for displaying the results as well as UI implementation which needs to be completed within a span of 1 year. We have divided the total available time in this project group in two phases, the first phase would be the time until end of the semester and the second phase would be the time till we start the integration phase. This plan describes in detail the tasks of only first phase and an abstract view of second phase.

We have categorized the tasks of phase one and described the timeline for each task.

0.2 PRIMARY TASKS

0.2.0.1 Document Retrieval

0.2.0.1.1 Fetching the documents using LD2NL framework.

0.2.0.2 Implementing indexing on documents

0.2.0.2.1 Calculation of all the parameters required in the ranking model.

0.2.0.2.2 Creating Index Table.

0.2.0.3 Scoring the documents/ranking

0.2.0.3.1 Implementation of BM25F Ranking function for each Query-Document pair.

0.2.0.4 Making it work for more than one KG

0.2.0.5 Implementation of UI

- Improvement of the existing UI to incorporate aforementioned features

0.3 FIRST PHASE

In the first phase, we take first two primary tasks of indexing and scoring the documents as our goal. We start this by get- ting hands-on on GENESIS (node JS), LD2NL Framework and SPARQL.

0.3.1 DOCUMENT FETCHING AND CREATE HIGH LEVEL DESIGN DOCUMENT

(2 WEEKS)

The very first task is to retrieve the text/documents (on which the further task of ranking will be implemented) that contains entity description, from the Avatar and Triple2NL packages within the LD2NL framework. Our initial approach is to build an interface which consumes Entity description text via APIs. Furthermore, each document is sent to next stage in the pipeline for Indexing.

3.2 INDEXING (4 WEEKS)

Information Retrieval engines rely on indices for efficient access to the information required for computing scores at query time. We are planning to use indexing table for ex- ample: ***R-vertical indexing*** [1] as the index structure for improved performance of the search. In this stage, each doc- ument is indexed based on the term(s) appearing in the con- tent.

The output of this stage is the index table for each document. The table contains all the relevant parameters required for the Scoring/ ranking the document.

MG4J [4], an open-source engine for text indexing, is our choice for implementing Indexing.

3.3 SCORING USING THE RANKING MODEL (4 WEEKS)

For ranking model, BM25F ranking model [1] is our choice. Using BM25F, document D is scored against a query Q using a summation over individual scores of query terms $q \in Q$:

$$score^{BM25F}(Q, D) = \sum_{q \in Q} w_i^{BM25F}$$

0.4 SECOND PHASE

In the second phase, we have planned to execute the remaining of the assigned tasks as follows:

0.4.1 Changing the data source and make it work for more than one KG:
(Task by Barun Kumar and Taslima Akter)

Fetching data from DBpedia

In this phase of summer semester, I would be working on fetching the data from another knowledge base. At present, we are using the index to get the data and on the next phase, we would be using DBPedia as our knowledge base. We will use SPARQL query to retrieve data for an entity. The query would return an RDF triple which is a set of subject, predicate and object. The triple would then be processed to extract a set of relevant entities. The entities would further be ranked using the existing ranking function.

0.4.2 Improvements to existing ranking algorithm (Task by Aman Sharma and Paaras baru)

We will add additional phases in the nlp pipeline like entity recognition to understand more about the entity to improve our search results. We plan to go with 2 approaches. One where we decide an entity by appropriate library and give the search results

The other where we give user a list of probable entities that the query term could be and based on user need refine our search results. This can be done by adding an entity type dropdown in UI, and hence user can decide what he wants(eg. whether dresden should be searched as a band or a place).

0.4.3 Integration to Kotlin(Task by Saud)

This task would involve integrating our created bm25 service to Genesis using kotlin.

- Integrating our created bm25 service to Genesis using kotlin/JAVA.
- Replace the current “LUCENE_SERVICE” with our own rest API.
- Update the ports in index file, and call our service in App.kt file.
- Else, create another service to integrate BM25 service with our project.

All the tasks will be performed independently and simultaneously by all team members and will be discussed weekly for any integration issues.

0.5 ORGANIZATION

- Responsibilities and Scrum Master: Aman Sharma.
- External Team Meeting: Every Monday, 11:00.

- Internal Team Meeting: Every Monday, 18:00 to 20:00
- Team members will fill the weekly work report of their tasks.
- Total weekly working hours: 20 hours.

0.6 Technologies and Tools used

We use the following environment in the project.

- Tools: Java8
- Eclipse
- Node JS

- LD2NL
- GENESIS

Communication: Slack and Trello. Versioning:

GitHub

MG4J

python

0.7 DEADLINES AND MILESTONES FOR 1ST PHASE AND 2ND PHASE

Serial. No.	Milestone	Deadline	Phase	Duration for tasks	Number of Person working
1.	Creation of the Project Plan	25-11-2019	FIRST	2 weeks	5
2.	Document Fetch-ing	06-12-2019	FIRST	2 weeks	5
3.	Indexing	27-01-2020	FIRST	3weeks	5
4.	Scoring using the ranking model	17-01-2020	FIRST	3 weeks	5
6.	First phase Presentation	20-01-2020	FIRST		5

References:

[1]https://drive.google.com/file/d/1C-aNUNcJtGlvg5mXDz2AterM7INnbIw-/view

[2]https://drive.google.com/open?id=1Tq4F5oJqXtH75pFqKvwgby3z5ieef5AZ

[3]https://dl.acm.org/citation.cfm?id=3106514

[4] https://dl.acm.org/citation.cfm?id=1863879.1863882