Master 1 in computer science
INF 4188: Semantic Web and application
Practice /30

Dr. Azanzi Jiomekong

Objective: The main goal of the overall exercise is to allow students to understand semantic web and its application and to be able to build semantic web toolkits to solve real world problems. Thus, they will learn how to acquire knowledge from knowledge sources and use this knowledge to build knowledge graphs. Thereafter, by developing an ontology and an inference engine, they will learn how to use the knowledge graph to infer new facts given to existing ones. Finally, they will integrate these resources to solve real world problems.

NB: The overall exercises are out of more than 20 for the continuous assessment and more than 30 for the practice. However, students' marks will be maintained out of 20 for the continuous assessment and 30 for the practice.

Evaluation given to students: 22-03-2023

Evaluation of the work every week

Lecture starting: 10-04-2023

Final evaluation: 19-04-2023

Adequate nutrition is an essential catalyst for economic and human development as well as for achieving Sustainable Development Goals - Goal 2: Zero Hunger and Goal 3: Ensure healthy lives and promote well-being for all at all ages. However, understanding Food information can allow people to have a healthy diet. To this end, food information engineering involves the acquisition, the processing and the diffusion of up-to-date food information to different stakeholders. These information are compiled from several data sources and used for a variety of purposes such as food recommendation, recipe substitution, food image recommendation, nutritional agendate, etc. To recommend a food to a person, a task can consist of first recognizing this food (e.g., from a food image, from a Food Composition Table), and identifying the food components contained in this food. To this end, a set of data sources including food images and food composition tables should be annotated.

Comprehension of the different exercices proposed: (/6pts)

To solve a problem, the first step consists of reading and understanding the problem to be solved. Thus, the students should read carefully each exercise and will prove during the lectures that they understood what is expected from them.

1- Reading and understanding of exercise 2	2pts
2- Reading and understanding of exercise 3	2pts
3- Reading and understanding of exercise 4	3pts

Continuous assessment: (/20pts)

<u>Exercice 1</u> - Daily evaluation during the lecture: 6pts

All the questions of this exercise will be defined during the lecture.

Development environment, knowledge, Facts representation, modeling and serialization, RDF, RDFS, OWL, SPARQL, reasoning mechanism, Ontology modeling

<u>Exercice 2</u> - Annotating TSOTSATable dataset using existing knowledge graphs: 12pts

The TSOTSATable dataset in the form of CSV file can be used as input format in a data analytics pipeline. However, a lack of understanding of the semantic structure and meaning of the content may hinder the data analytics process. Thus gaining this semantic understanding will be very valuable for data integration (integrating food composition tables of different countries), data cleaning, data mining, machine learning (food recommendation), knowledge discovery, reinforcement learning (robot cooker), etc. For example, understanding food composition tables can allow for recipe substitution. Thus, the addition of semantics to TSOTSATable dataset may enhance a wide range of applications, such as food search, food question answering, Food Knowledge Graph construction, food recommendation, recipe substitution, Automatic Generation of Food Composition Tables etc.

To make the TSOTSATable dataset machine understandable, one should match the latter to a KG. This corresponds to the Tabular Data to Knowledge Graph matching research problem. Tabular data to KG matching is the process of assigning semantic tags from KG (FoodOn, Wikidata, DBpedia, etc.) to the elements in the table. This task is however difficult in practice due to the problems that the students will encounter during the exercise.

The goal of this exercise is to match the TSOTSATable dataset to FoodOn and Wikidata KG, so as to produce a semantic Food Composition Table. In this exercise, the students will consider the following types of annotation:

(i) Column Type Annotation (CTA): this task consists of the annotation of an entity column (i.e., a column composed of entity mentions) in a table with types from Wikidata and FoodOn. Each column can be annotated by multiple types and the one that is as fine grained as possible should be marked as perfect annotation. The one that is the ancestor of the perfect annotation is regarded as an OKAY annotation and all the others are wrong annotations. Each column should be annotated by at most one item. The annotation should be represented by its full IRI. The output file should be a CSV file. Each line should include a column identified by table id and column id and the column's annotation.

The annotation should be represented by its full IRI, where the case is NOT sensitive. Each output file should be the corresponding CSV file that is annotated. Each line should include a column identified by table id and column id, and the column's annotation (a Wikidata and FoodOn item). It means one line should include three fields: "Table ID", "Column ID" and "Annotation IRI". The headers should be excluded from the submission file.

It should be noted that:

- Table ID is the filename of the table data, but does not include the extension
- Column ID is the position of the column in the input, starting from 0, i.e., first column's ID is 0
- One submission file should have NO duplicate lines for each target column
- Annotations for columns out of the target columns are ignored

(ii) Cell Entity Annotation by Wikidata and FoodOn: this task consists of the annotation of column cells (entity mentions) in a table using Wikidata and FoodOn. It consists of annotating each target cell with an entity of Wikidata and FoodOn. Each output file should contain the annotation of the target cell. The output file should be in CSV format. Each line should contain the annotation of one cell which is identified by a table id, a column id and a row id. Each cell should be annotated with at most one entity.

It should be noted that:

- Table ID does not include filename extension; m

- Column ID is the position of the column in the table file, starting from 0, i.e., first column’s ID is 0
- Row ID is the position of the row in the table file, starting from 0, i.e., the first row’s ID is 0.
- One submission file should have NO duplicate lines for one cell
- Annotations for cells out of the target cells are ignored

Column Property Annotation by Wikidata and FoodOn: this task consists of the annotation of column relationships in a table with properties of Wikidata and FoodOn. This is the annotation of each column pair with a property of Wikidata and FoodOn. Each output file should contain an annotation of a target column pair. Note that the order of two columns (domain - range) matters. Each line of the output file should contain the annotation of two columns which is identified by a table id, “column id one” and “column id two”. One line should have four fields: “Table ID”, “Column ID 1”, “Column ID 2” and Property IRI”. Each column pair should be annotated by at most one property.

It should be noted that:

- Table ID does not include filename extension
- Column ID is the position of the column in the table file, starting from 0, i.e., first column’s ID is 0
- One submission file should have NO duplicate lines for one column pair
- Annotations for column pairs out of the targets are ignored

NB: Templates files annotated are given as associated resources to this exercise.

Each student should solve the following tasks:

1- Annotate the 250 files that is given to you using Wikidata and FoodOn **(2.5 x 2 =5pts)**

2- Extract tables from the scientific papers given to you and annotate these tables using Wikidata and FoodOn **(2 + 1.5 + 1.5 = 5pts)**

3- Cataloging of all problems identified in the tables using the following table **(3pts)**

Problem identified	**Tables IDs**

4- Take inspiration of the manual annotation and propose a system for annotating TSOTSATable dataset **(5pts)**

5- Evaluate your system using the following metrics: Precision, Recall and F-score **(1 x 3 = 3pts)**

Practice (/30pts)

During the first semester, you downloaded and annotated several food images. In the previous exercise, you annotated the TSOTSATable dataset. In this exercise, these resources are going to be used to build a FoodKG for food search and food recommendation.

Exercice 3: TSOTSAOnto construction 20pts

1- Considering the TSOTSA project (confer the introduction), identify the problem and the context of this work and explain how an ontology can help to solve this problem**1pt**

2- Considering the description of the TSOTSATable dataset, the resources produced in INF 4077 (table 4 of the exercise 1), propose a UML model for describing food and its components. **2pts**

Img_id	Dish name	Objects found (dish content)	Ingredients	Source		
				Author	Title	Link

3- Define a set of axioms that can be used to infer new facts using this ontology **(0.25 / 12 = 2pts)**

4- Use Protégé to implement the ontology using the OWL serialization and enrich the ontology using the results of the table annotated **(2pts)**

5- Populate the ontology using some data that you automatically generated (5000 instances can be generated for the purpose of this exercise) **(2pts)**

6- Use the existing reasoner of your choice to infer new facts from this ontology - what is the limit of these reasoners? (you can put in a table) **(2 + 1 = 3pts)**

7- Propose a new reasoner (involving a SPARQL query engines that support entailment regimes such as RDFn RDFS or OWL 2) and test this reasoner on TSOTSAOntology **(5pts)**

8- Evaluate the reasoner by filling the table below **(1pts per reasoner)**

Axiom	**Reasoner**	**Memory consumption**	**Reasoning time**	**Comments**
Reasoner i				
Reasoner i+n				
Your reasoner				

9- Deploy your ontology using BlazeGraph **(1pt)**

10- Apply some SPARQL query on the ontology **(1pt)**

11- Bonus: **(+10 pts)**

- Performance of the tools that you built

- Metric of the ontology

Exercice 4 - Improving food search: 10pts

To help people to have a healthy habit, the improvement of the food search results can be of great help. However, correctly classifying food items for a particular user search query can be challenging. The presence of noisy information in the results, the difficulty of understanding the query intent, and the diversity of the foods and nutrients available and user health profile are some of the reasons that contribute to the complexity of this problem. The primary objective of this exercise is to build a ranking strategy and simultaneously, identify interesting categories of results (i.e., substitutes) that can be used to improve the consumer experience when searching for relevant food. This problem is well known problem of system recommendation

When developing food recommendation systems, extremely high accuracy is needed to help people to have healthy habits. Even more, when deploying food search in mobile phones, voice and image search applications, where a small number of irrelevant foods can be used for better recommendation.

In these systems, the notion of binary relevance limits the customer experience. For example, for the query “Koki”, would “Koki banana” be relevant, irrelevant, or somewhere between? In fact, many people may search “Koki” to find “Koki banana”: they expect the search engine to understand their needs. But, “Koki” can also be eaten with “Potatoes”, etc. For this reason, the search relevance can be break down into the following four classes (ESCI) which are used to measure the relevance of a food item in the search results:

(i) Exact (E): the food item is relevant for the query, and satisfies all the query specifications (e.g., “Koki banana” matching all attributes of a query “Koki banana orange juice”)

(ii) Substitute (S): food item is somewhat relevant: it fails to fulfill some aspects of the query but the item can be used as a functional substitute (e.g., “Koki potato” for “Potatoes” query)

(iii) Complement (C): the food item does not fulfill the query, but could be used in combination with an exact item (e.g., “Eru” for “Fufu cassava”)

(iii) Irrelevant (I): the food item is irrelevant, or it fails to fulfill a central aspect of the query (e.g., “Orange juice” for “Eru” query).

Question 1

Given the lack of a food dataset, the first step of this work consists of proposing a “Food Queries Data Set”. This dataset should contain difficult search queries which aim to foster research in the area of semantic matching of queries and food products / dishes. In this dataset, for each query, the dataset should provide a list of up to 40 potential relevant results, together with ESCI relevance judgment indicating the relevance of the food product to the query. Each query-food pair is accompanied by additional information. The information accompanying every food product including title, food description. This should be a multilingual dataset containing queries in English and French. **(1 x 5 foods = 5 pts)**

Using the datasets defined in question 1, the following tasks can be solved:

(i) Ranking results list: Query-food Ranking

Given a user-specified query and a list of matched foods, the goal is to rank the food products so that the relevant foods are ranked above the non-relevant ones. The input of this task will be a list of queries and the output a table containing as the first column the query_id and the food_id in the second column. In this result, each query_id in the first row will be the most relevant food and the last row the least relevant food. The input data for each query will be sorted based on Excacts, Substitutes, Compliments, and irrelevants. In the following example, for query_1, food_50 is supposed to be the most relevant food item and food_80 is the least relevant one.

Input:				**Output:**	
query_id	query	query_locale	food_id	query_id	food_id
Query_1	"Query_1"	us	food_23	Query_1	food_50
Query_2	"Query_2"	us	food_234	Query_1	food_900
				Query_1	food_80
				Query_2	food_32

Evaluation

Normalized Discounted Cumulative Gain (nDCG) is commonly used as the relevance metric. Highly relevant documents appearing lower in a search results list should be penalized as the graded relevance is reduced logarithmically proportional to the position of the result. In this exercise, we have four degrees of relevance for each query and food pair: Exact, Substitute, Complement, and irrelevant. We set a gain of 1.0, 0.1, 0.01 and 0.0 respectively.

DCG_p shows how to compute the Discounted Cumulative Gain (DCG) for a list of the first p relevant food retrieved by the system. IDCG_p computes DCG for the list of p relevant foods sorted by their relevance (|REL_p|), therefore, IDCG_p returns the maximum DCG score.

Search results lists vary in length depending on the query. Comparing a search engine’s performance from one query to the next cannot be consisted achieved using DCG alone, so the cumulative gain for each position for a chosen value of p must be normalized across queries: where nDCG_p is obtained dividing DCG_p by IDCG_p.

(ii) Classifying the query/food pairs into E, S, C, or I categories: this is a multiclass food classification

Given a query and a result list of foods retrieved for this query, the goal of this task is to classify each food as being an Exact, Substitute, Complement, or Irrelevant match for the query.

The input to this task will be pairs, along with food metadata. Specially, rows of the dataset will have the form:

Input:

example_id	query	food_id	query_locale
example_1	11 degrees	food0	us
example_2	11 degrees	food1	us

Output:

example_id	esci_label
example_1	exact
example_2	complement
example_3	irrelevant
example_4	substitute

Evaluation

F1 score is commonly used as a metric for multi-class classification. The micro-F1 score computed a global average F1 Score by counting the sums of the True Positives, False Positives and False Negative values across all classes. It can also be used to evaluate the algorithm used.

(iii) Identifying substitute foods for a given query: this consists of food substitute identification

This task measures the ability of the system to identify the substitute foods in the list of results for a given query. The notion of “substitute” is exactly as in **(ii)**. The input of this task is the same as **(ii)**. The system will output a table where the example_id will be in the first column and the substitute_label in the second column. See the table below.

Input:

example_id	query	food	query_locale
example_1	query_1	food0	us
example_2	query_2	food1	us
example_3	query_3	food2	jp
example_4	query_4	food3	jp

Output:

example_id	substitute_label
example_1	no_substitute
example_2	no_substitute
example_3	substitute
example_4	substitute

Evaluation

The F1 score can be used to evaluate and rank the algorithms used. Particularly, the micro-F1 can be used to evaluate proposed systems.

Question 2

Apply the basic retrieval models such as BM25 along with BERT model for the first task on your dataset (**2.5pts)**

Question 3

Run the multilingual BERT-based models for the two other tasks **(2.5 + 2.5 = 5pts)**