Universität Stuttgart
IPVS
Data Engineering

Summer term 2018
Prof. Dr. Melanie Herschel
18.04.2018 – 09.05.2018

Excercise Sheet 01

1. Information Pyramid

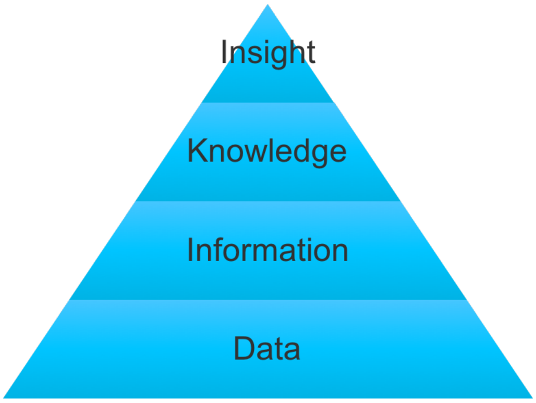


Figure 1: Information Pyramid

In class, the information pyramid is introduced to describe the relationships between data, information, knowledge, and insight. An example about a twitter feed from Bradley Scott illustrates these relationships.

(a) Give a general explanation of each layer of the pyramid.

(b) Assume you like to gain insight from the following three datasets:

(i) Database about olympic games: `https://www.olympic.org/`

(ii) Database about soccer world cup: `https://openfootball.github.io/`

(iii) Database about tourism: `http://www2.unwto.org/` and `http://ec.europa.eu/eurostat/web/tourism/data/database`

Propose an example insight derived from combining these datasets. Describe how you obtain the insight from the data following the steps of the information pyramid.

(c) Let us now focus on the data engineering part of the information pyramid. To this end, consider the data engineering pipeline discussed in class. Illustrate the problems that need to be solved by each (dark blue) component of the data-engineering pipeline, reusing your example from the previous subtask.

2. Knowledge Bases

A knowledge base is considered a collection of facts that describes information about entities and their properties. They rely on concepts that describe information about entity types and their properties. In class, we mentioned the following knowledge bases: Yago, Freebase, DBPedia, and Google Knowledge Graph.

(a) Find out how many facts the knowledge bases contain.

(b) Describe the APIs the knowledge bases support. Is there an API that is supported by all the knowledge bases?

(c) Compare the output of the knowledge bases on ambiguous terms like "Java".

3. Entity Annotation

Knowledge bases are widely used to tag words in written language. This is also called entity annotation. The University of Pisa offers an online tag manager called tagme (`https://tagme.d4science.org/tagme/`). It performs entity annotation online.

(a) Tag the sentence "Python is more fun than Java." Which words does tagme annotate? Do the annotations make sense?

(b) Tag the sentence "Java is smaller than Borneo." Which words does tagme annotate? Do the annotations make sense?

(c) Is Java associated with the same annotation in both sentences? Did tagme fail?

(d) Create a sentence that associates Java with the "wrong" entity.

(e) Find 3 more ambiguous words and describe how tagme handles them.

4. Data Selection: WebTables Ranking

A lot of structured data is available on the Web. Not only parsing the data is a challenge, but also selecting web-extracted relational data. While keyword ranking works well on web documents, it lacks the ability to appropriately rank structured data. Therefore, more appropriate ranking algorithms have been discussed in class. This task focuses on these ranking algorithms.

(a) What is an optimal ranking result? Describe the difference between the naiveRank algorithm and the filterRank algorithm. Which one of them yields better results? Why does none of them yield optimal results?

(b) Let us have a look at the more sophisticated featureRank approach discussed in class. The featureRank approach makes use of table features such as number of rows. Let us assume we have the following tables available in our table collection:

Relation-ID	#rows	#cols	has-header?	#NULLs in table	search rank
Beta1	150	10	TRUE	5	5
Beta2	10	5	TRUE	0	5
Oxford1	50	5	TRUE	0	4
Random1	60	7	FALSE	50	7
Random2	20	3	FALSE	10	8
Statista1	10	3	TRUE	0	3
Statista2	100	5	TRUE	0	3
Statista3	1000	20	TRUE	500	6
Wiki1	10	2	TRUE	0	1
Wiki2	500	10	TRUE	20	2

For which features are higher values considered better? For which ones are lower values better? Think of further relation features not shown in this table that could be considered for relation ranking (possibly beyond those seen in the lecture).

(c) In the original work, a linear regression estimator is trained to calculate a feature score. Creating a linear regression estimator is out of the scope of this exercise. Therefore, use simple mathematical operations to combine all features shown in the above table (except ID). Define two different score functions to rank the relations. Which formula do you think is more suitable to rank the given relations? Explain why.

(d) Given the scores for all relations, use the featureRank algorithm introduced in class to provide the top three relations. Give a step by step exlplanation.

(e) Now let us have a look at the schemaRank approach. What is the complexity of the algorithm introduced in class to calculate the coherency for a relation with n attributes? What is the value range the cohere function returns? What do negative values mean?

(f) We are given the following coherences for the above relations:

Relation-ID	coherency score
Beta1	5.67
Beta2	3.55
Oxford1	6.22
Random1	1.78
Random2	-0.23
Statista1	5.21
Statista2	3.12
Statista3	1.56
Wiki1	4.33
Wiki2	0.88

Modify your favored score formula from task (c) so that it considers the coherency score. Pay special attention to negative coherency scores.

(g) Calculate the score for each relation with the help of your new score formula. Use the new score values and the featureRank algorithm to provide your top three relations. Is the result a different one than that from task (e)?

5. Data Extraction & Selection: Challenges

As discussed in class, three major challenges arise on finding and using relational data in search engines. This task is about understanding these challenges.

(a) Describe the three challenges discussed in class (slides 75-86 in chapter 2) in your own words.

(b) Let us have a closer look at the second challenge. Find three example relations (different web pages not mentioned in class) that provide high quality relational data. Find an example table that does not contain relational data (different from the ones in class).

(c) In the context of the second challenge, we have a closer look at precision and recall. How is precision and recall defined? Intuitively, what do precision and recall indicate?

(d) Let us assume we have analyzed 100 relations with identifiers (ID) 1...100. The ones in ID range [21, 42] are true high quality relations. The other ones do not contain relevant relational data. We have implemented two different WebTables extractors which yield the following IDs as result:

$$E_1 = \{3, 7, 19 - 22, 25 - 31, 33 - 37, 39 - 44, 51 - 55, 66\}$$

$$E_2 = \{8, 9, 19 - 33, 36 - 44, 50\}$$

Note that, for example, $19 - 22$ is an abbreviated notation for $19, 20, 21, 22$. Calculate precision and recall for the two extractors. Which extractor yields better results? Explain your answer.

(e) Relational data may not only be present in HTML tables, but also other structures like lists. What are possible challenges to obtain relational data from lists?

6. Data Extraction: Tables from Lists

As introduced in the last exercise, it is a challenge to obtain structured data from lists. This task illustrates the difficulty to impose a table layout on lists containing structured data. The concepts are introduced in detail in *Harvesting relational tables from lists on the web* by Elmeleegy et al. (VLDB Journal 2011).

(a) Given the following list of cartoons:

(1) Three Little Pigs (Disney/1933)

(2) Gertie The Dinosaur (MacCay)

(3) Popeye the Sailor (Fleischer/1936)

(4) ...

Suggest a good table schema for the example. Which characters from the example above are good characters to split the line string? What is an additional challenge regarding the completeness of the data?

(b) The following subtasks should give you an impression of the challenge to automatically derive field values from individual lines of the list. Let us assume the first line is split into 5 words [Three, Little, Pigs, Disney, 1933]. How many field candidates exist? For example, a line consisting of the three words [Three, Little, Pigs] has the following field candidates:

(1) [*Three*, *Little*, *Pigs*] (3 fields)
(2) [*Three Little*, *Pigs*] (2 fields)
(3) [*Three*, *Little Pigs*] (2 fields)
(4) [*Three Little Pigs*] (1 field)

In general, how many field candidates can be generated from a line with m words?

(c) Now let us find out how many unique word combinations exist among all combinations. That is important to efficiently calculate a field quality score for each combination. A field quality score is a value that indicates how well a word combination fits into a potential column. Let us reuse the line with the three words [Three, Little, Pigs]. The unique combinations are:

(1) [*Three*]
(2) [*Little*]
(3) [*Pigs*]
(4) [*Three*, *Little*]
(5) [*Little*, *Pigs*]
(6) [*Three*, *Little*, *Pigs*]

That makes 6 unique combinations. How many unique combinations can be derived from the first line with 5 words? How many unique combinations does a line with m words yield?

(d) The field quality score introduced in the previous subtasks is significantly influenced by the type score of a field candidate. The type score indicates if a candidate is recognized as a member of a type that commonly occurs in seperate table columns. Types may be *String, Int, Datetime, Float, ...* Let us have a closer look at the following field candidates:

(1) [*Disney*, 1933]
(2) [*Disney*] and [1933]

Which of the two options gets a higher field score? Explain your answer.

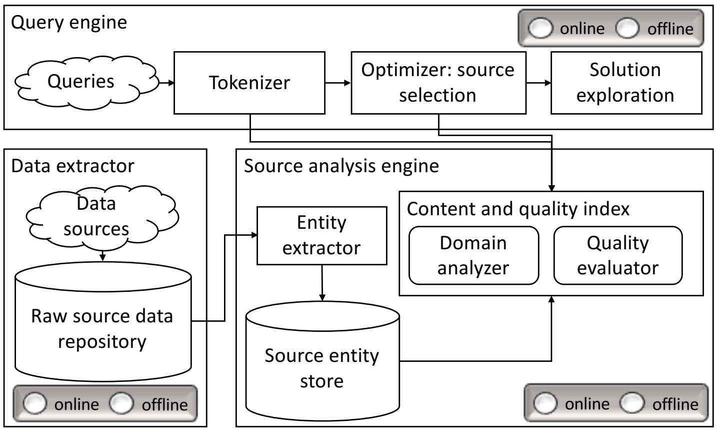
(e) *Optional:* From the previous tasks you gained a first impression on the challenges to lay table structures on lists. If you like to know the entire solution to this problem, read *Harvesting relational tables from lists on the web* by Elmeleegy et al. (VLDB Journal 2011).

7. Data Collection (exam 2016, 25 points)

This exercise focuses on **Data Source Management Systems (DSMS)**, as they have been recently proposed and discussed in the chapter on Data Collection. In this context, please answer the following questions.

a) The main goal of DSMSs is to semi-automatically support data engineers in selecting relevant sources. Name the **three steps that typically need to be performed manually** and that such systems may thus simplify.

b) Below, you see the general architecture of a possible DSMS as well as keywords describing relevant concepts within DSMS. Answer the following questions.



○ Knowledge Base

○ Keywords

○ Data quality metrics

○ Web data

○ Context cluster

○ Pareto frontier

○ User interaction

(i) What do online and offline processing mean in the context of DSMS?

(ii) **Tick** the correct box (online / offline processing) for each main component in the figure.

(iii) Connect each **keyword** with the **component(s)** it is associated with through lines to the figure. Make sure that your association is **as specific as possible**, e.g., if a keyword associates with the Tokenizer, connecting it to the more general query engine will be considered incorrect.

c) We now focus on knowledge bases.

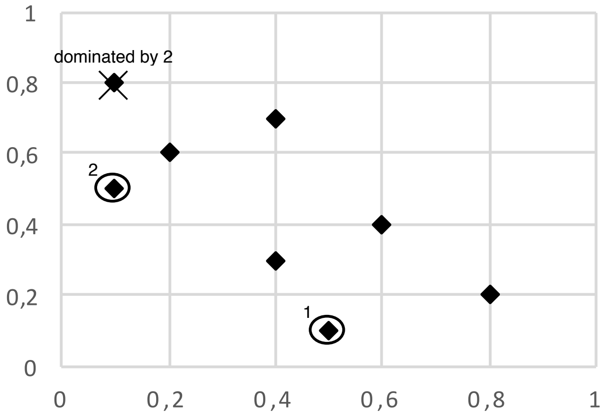
(i) How is the term **knowledge base** defined?

(ii) How is a knowledge base used in a DSMS.

d) Concerning the Pareto frontier (also called Pareto front), answer the following questions.

(i) Provide the definition for the Pareto frontier.

(ii) We have seen that **skyline queries** can be used to compute the Pareto frontier. One algorithm to compute a skyline is **plane sweep**. We are going to apply this algorithm on the sample 2-dimensional data shown below.
Two points of the skyline are already circled and labeled in the order they have been found by the algorithm. In addition, the figure also shows one point that to be pruned from the skyline, as it is dominated by point 2.



A. Provide labels for the x- and y-axis that are reasonable labels in the context of DSMS.

B. Using the plane sweep algorithm discussed in class, determine the remaining points on the skyline. Your answer should include the following annotations on the figure above:
   - Circle each point of the skyline and provide a label indicating at which iteration it was obtained (similar to points 1 and 2).
   - Cross through points that are discarded from the skyline during processing and label them with the points dominating them (similar to point dominated by 2).
   - Draw the "sweep lines" where processing can stop as it is guaranteed that all points on the skyline have been found.

8. Programming Exercise: Skyline Operator

The skyline operator filters results from data related queries in a way that it keeps only those data items that are not worse than any other with respect to a set of dimensions. In class, the skyline operator was discussed in detail. Remember that this is a group exercise and that among other things you have to present and pass two out of three programming exercises in order to be admitted to the exam.

a) Implement a skyline query algorithm that takes as a set of two-dimensional points $p_i = (x_i, y_i)$ as input. After executing the skyline operator your algorithm should return the set of points that is not worse than any other with respect to the dimensions x and y. Assume, that in both dimensions x and y higher values mean better quality. You can use any approach (e.g. those described and referenced in the lecture) and implement the method in a programming language of your choice.

   For demonstration purposes your program should be able to read a csv file which contains the input set of data points. In the csv file each row represents a data point specifying values for x and y. An example row is $42; 4711$. You can test your program with the offered test.csv and compare your result to the solution in the test_sol.csv.

b) Sketch your solution approach, i.e., the algorithm you have implemented, the data structures you used, etc. Also show the key parts of the code.

c) Demonstrate your working algorithm in class. Calculate the solution for a new dataset which will be given to you right before the demonstration.

d) How could you extend your approach to processing both dimensions where smaller values are preferable and other dimensions where higher values are preferable?

e) What is the runtime complexity of your algorithm? In which parts of the algorithm does the complexity reside? Sketch a proof.