Informatics 1: Data & Analysis
Session 2016/17, Semester 2

Assignment Feedback

This is a report on the written coursework assignment for *Informatics 1: Data & Analysis*. That assignment included exam questions from 2013 and 2016, and this report is based on solutions submitted by students. Please note the following:

- This is not a set of "model" answers. It does contain solutions, which can be used to check your own answers; but there are also notes on different possible answers, key points, possible errors, and comments on the ways people approached each question in the exam itself and as coursework.
- Not all the questions have a single "right" answer. There can be multiple correct ways to write a database query, explain a concept, or construct an example. This report includes some variants on answers, but still cannot cover every possible correct alternative.
- Practising past exam questions is one way to learn more about a subject, but it is quite limited and not enough on its own. Even when an exams routinely follow a fixed structure, the questions change and successful performance does essentially depend on a good understanding of the material in the course.

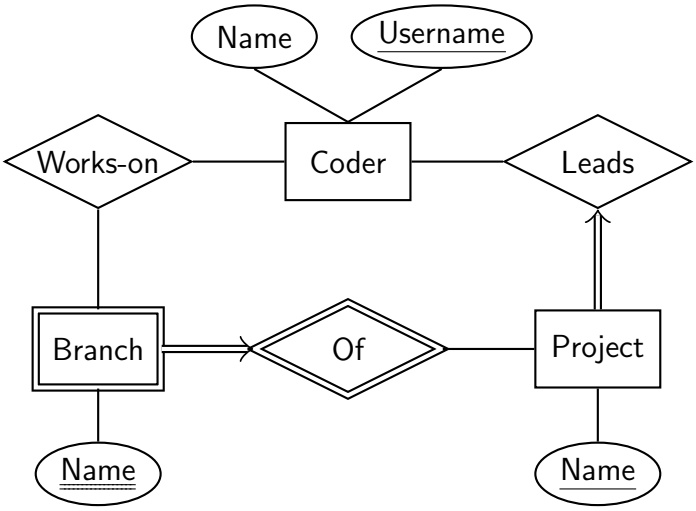
The assignment consisted of three questions, each with several subquestions. The rest of this report gives the full text of each question followed by notes on solutions and feedback on the answers given by students.

Where you find errors in these notes, please send them to me at Ian.Stark@ed.ac.uk

Ian Stark
2017-05-06

Question 1 [*This question is worth a total of 30 marks.*]

The following entity-relationship diagram captures information about a number of *coders* who work together on a wide range of software *projects*. A project may have a number of different *branches* under active development at any one time. Branches are named, often using standard descriptions such as "master", "stable", or "unstable".



(a) What is the meaning of the double line around Branch? Why is this needed? What is the primary key for a Branch? [*6 marks*]

(b) Construct SQL data declarations for a set of tables to represent this entity-relationship diagram. Assume that usernames are limited to 64 characters and all other names fit within 200 characters. Make sure to use **not null** where necessary; however, you do not need to include **on delete** declarations. [*24 marks*]

Notes on Question 1

(a) The double line indicates that Branch is a *weak entity*. This is necessary because a branch name is only unique within its project — many different projects will have their own "master" branch. The primary key for a Branch is the composite {Branch.name, Project.name}.

Note that there are three separate parts to this question: which mean there need to be three clear answers. That's not necessarily difficult, but it's easy to miss when concentrating on the content.

(b) Here is a suitable set of data declarations.

```
create table Coder (
  username varchar(64),
  name     varchar(200),
  primary key (username)
)

create table Project(
  name  varchar(200),
  leader varchar(64) not null,
  primary key (name),
  foreign key (leader) references Coder(username)
)

create table Branch(
  name   varchar(200),
  project varchar(200),
  primary key (name,project),
  foreign key (project) references Project(name)
)

create table WorksOn(
  coder   varchar(64),
  branch varchar(200),
  project varchar(200)
  primary key (coder, branch, project),
  foreign key (coder) references Coder(username),
  foreign key (branch,project) references Branch(name,project)
)
```

There's quite a lot to do here, ranging from the very simple Coder table to the WorksOn relationship that references all three other tables.

Here a few features that need care in their solution.

- The one-to-many relationship between coders and projects needs to be represented the right way round: with a leader field in the Project table, not a project field in Coder.

  The ER diagram is clear that each project requires exactly one lead coder. Sometimes, though, people try to start a solution with the foreign key in the target table not the source; that doesn't work.

  It's possible to build a partially-correct attempt using a separate Leads table that records which coder leads each project. This can even support the key constraint

if done carefully, but there's no way to express the total participation constraint on Project.

Note that there's no constraint in the diagram on how many projects any one coder can lead: zero, one, or many.

- The weak Branch entity needs to reference the Project name, and include that as part of its primary key. This picks up from part (a), where you need to identify the primary key for a Branch.
- The Works-on relationship has no key constraint so requires its own table. Moreover, because Branch is a weak entity, it must also mention the Project name of its identifying owner as it appears in the Branch table.

  There's an error that turned up more than once here. Where the Works-on relationship mentions a Project name, it must be the correct one for the Branch being worked on. This means that the following is incorrect:

```
create table WorksOnDoneWrong(
  ...
  foreign key (branch) references Branch(name),  -- This branch name might not
  foreign key (project) references Project(name)  -- be used in this project
)
```

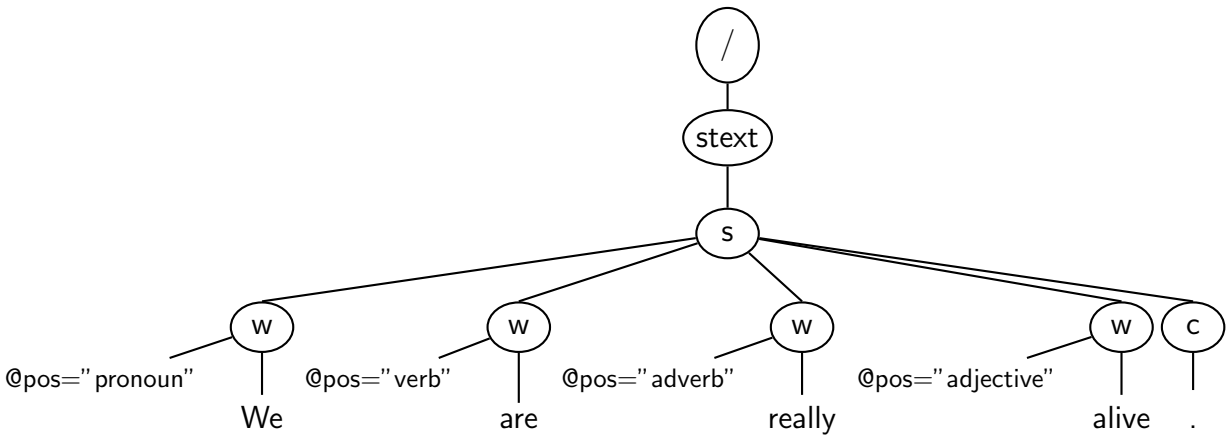
  The correct line is that given earlier

```
  foreign key (branch,project) references Branch(name,project)
```

  which makes sure the branch and project name match up correctly.

Question 2 [*This question is worth a total of 40 marks.*]

The following tree shows the XPath data model for a short XML document. It is a line of spoken text annotated using a simplified version of the British National Corpus mark-up scheme.



(a) Write out this tree as an XML document. [*9 marks*]

(b) This tree contains examples of all four of the main node types in XML v1.0. For example, the "/" at the top is a *root node*. Name the three other types of node in this tree, giving examples of each. [*3 marks*]

(c) In this mark-up scheme stext denotes spoken text, s indicates a sentence, w is a word and c punctuation. Every word is annotated with an appropriate part of speech (pos), taken from a long list of possibilities. A piece of spoken text may contain one or more sentences. Each sentence may contain words and punctuation but must begin with a word and end with punctuation.

Write a DTD to specify these constraints on the XML structure of a spoken text document. [*14 marks*]

(d) Write XPath expressions to return the following lists of text strings from any XML document that satisfies this mark-up scheme.

(i) All punctuation marks used.

(ii) All verbs.

(iii) Every adverb used in any sentence that uses an exclamation mark "!".

[*10 marks*]

(e) The document above is just one spoken line. Standard resources for linguistic research like the British National Corpus bring together work from many sources. Building such corpora requires *balancing* and *sampling* to ensure that they are representative. Explain the meaning of balancing and sampling here. [*4 marks*]

Notes on Question 2

(a) Here is the appropriate XML document.

```
<stext>
  <s>
    <w pos="pronoun">We</w>
    <w pos="verb">are</w>
    <w pos="adverb">really</w>
    <w pos="adjective">alive</w>
    <c>.</c>
  </s>
</stext>
```

(b) There are *element nodes* like stext or w; *text nodes* like really or alive; and *attribute nodes* like @pos="verb".

(c) The following DTD captures the constraints provided. The declarations can all appear in any order.

```
<!ELEMENT stext (s+) >
<!ELEMENT s  (w,(w|c)*,c) >
<!ELEMENT w (#PCDATA) >
<!ELEMENT c  (#PCDATA) >
<!ATTLIST w pos CDATA #REQUIRED >
```

The element that caused the most difficulty was "s". There are several possible alternate correct answers, such as "((w+,c+)+)", but there are also lots that don't work — either because they are too restrictive, or too broad. For example, it's essential to give an expression that allows arbitrary punctuation within a sentence, not just at the end.

(d) Here are some appropriate XPath expressions. The question explicitly asks for text strings, so the text() operator is important.

(i) All punctuation marks used.

```
//c/text()
```

(ii) All verbs.

```
//w[@pos="verb"]/text()
```

(iii) Every adverb used in any sentence that uses an exclamation mark "!".

```
//s[c/text()="!"]/w[@pos="adverb"]/text()

//c[text()="!"]/../w[@pos="adverb"]/text()

//w[@pos="adverb"][../c/text()="!"]/text()
```

For all of these it's possible to construct other correct solutions.

(e) **Balancing** means choosing a range of different types of sources for the corpus: books, newspapers, blogs, letters, etc.

Sampling refers to selecting texts at random from the chosen sources.

Question 3 [*This question is worth a total of 30 marks.*]

The non-existent start-up company *Find-a-Flick* has a website that makes film recommendations. The founders plan to monetise this some day by selling popcorn in flavours matched to individual movies through a deep-dive adaptive data-learning algorithm. For the moment, the site just suggests films based on keywords provided by a user.

For each of around 200,000 films, Find-a-Flick has a body of text built up from reviews, plot descriptions, and comments about the film. By counting occurrences of keywords in the text, the website makes recommendations of films to match user queries.

(a) The performance of an information retrieval system like Find-a-Flick can be evaluated in terms of its *recall* and *precision*. Informally, recall is the proportion of relevant results that are actually retrieved. Give a similar informal definition of precision.

(b) Which is more important for the Find-a-Flick service: recall or precision? Give a reason for your choice.

[*5 marks*]

The following table shows the keyword counts for text associated with three different films, computed to assist a search for "Exciting Scottish historical drama".

	Exciting	Scottish	Historical	Drama
Film A	30	15	0	10
Film B	2	2	4	1
Film C	0	0	4	3
Query	1	1	1	1

One way to identify which films are most relevant to the query is the *cosine similarity measure*, based on the *vector space model* of documents.

(c) Write out in full the formula for calculating the cosine of the angle α between the two four-dimensional vectors (x_1, x_2, x_3, x_4) and (y_1, y_2, y_3, y_4).

(d) Use this to rank these three films by relevance to the original query.

(e) The Find-a-Flick repository happens to have much more text about Film A than it does about either Film B or Film C — this is why the keyword counts are much higher for that film. Does this affect the ranking of Film A? If so, does it make it higher or lower? If not, why not?

[*15 marks*]

The Find-a-Flick team are testing out two possible information retrieval systems: *Hare* and *Tortoise*. These are being evaluated on a small test collection of just 4000 documents, with a single query for which there are 200 relevant documents. *Hare* returns 1200 documents from the collection, including 150 that are relevant; while *Tortoise* returns just 160, with 120 of them being relevant.

(f) Tabulate the results for each system and calculate their precision and recall on this test. Show your working.

(g) One way to combine precision and recall scores is to use their *harmonic mean*. Give the formula for this, and calculate its value for each of *Hare* and *Tortoise*.

[*10 marks*]

Notes on Question 3

(a) The *precision* of an information retrieval system is the proportion of results returned that are relevant to the original query.

(b) The precision of the Find-a-Flick service is more important: what matters is that a good proportion of the films suggested be relevant; not that every possible relevant film appears in the suggestion list. In a database with 200,000 there may be hundreds or even thousands of films relevant to a particular query.

Several students proposed recall as more important on the grounds that people would like a longer list of related films, or to include films that were only slightly relevant to the query. However, without precision you don't necessarily get either of these things: less precision means extending the recommendation list with films that aren't at all relevant to the query. You might remember the video of IBM Watson on Jeopardy!, and how the possible "answers" where it had low confidence were not just less relevant but sometimes absolutely terrible.

(c) This is the cosine formula for the two four-vectors.

$$\cos(\alpha) = \frac{x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4}{\sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}\sqrt{y_1^2 + y_2^2 + y_3^2 + y_4^2}}$$

The question specifically names the coordinates of the two vectors, and it's important to use those in the answer. Solutions that used other vectors or a generic formula like $\vec{u}.\vec{v}$ received at most partial marks — repeating a memorised formula is not the same as demonstrating fluency in applying that formula to a particular situation.

(d) For the three films given, the appropriate calculation is the cosine between each film's text vector and the query keywords.

$$\cos(\text{Film A}, \text{Query}) = \frac{30 + 15 + 10}{\sqrt{4}\sqrt{30^2 + 15^2 + 10^2}} = \frac{55}{2\sqrt{1225}} = \frac{55}{70} = 0.79$$

$$\cos(\text{Film B}, \text{Query}) = \frac{2 + 2 + 4 + 1}{\sqrt{4}\sqrt{2^2 + 2^2 + 4^2 + 1^2}} = \frac{9}{2\sqrt{25}} = \frac{9}{10} = 0.90$$

$$\cos(\text{Film C}, \text{Query}) = \frac{4 + 3}{\sqrt{4}\sqrt{4^2 + 3^2}} = \frac{7}{2\sqrt{25}} = \frac{7}{10} = 0.70$$

I did suggest in lectures that using multiple digits of precision here is a poor choice: the original data doesn't support it, and it makes no difference to the ordering. Most answers given by students followed that; a few still gave four or even five decimal places, which really isn't appropriate or necessary.

Some students calculated the cosines between pairs of films (A, B), (B, C) and (A, C). That's not helpful at all in ranking their relevance to the query, and didn't earn any marks.

These cosines rank the three films in order of relevance as follows.

1. Film B
2. Film A
3. Film C

It's possible to calculate the angles between the vectors and rank those, but it's enough to work with the cosines themselves (larger cosine = smaller angle = better match).

Notice that the question asks for a ranking of all three films: a few students just gave the highest-rated film, which only achieved partial marks.

(e) Having generally higher keyword counts does not make a difference to the ranking. This is because the cosine measure looks at the angle between document vectors and is not affected by their length.

(f) The following tables give all the necessary figures for the calculation.

Hare	Relevant	Not relevant	Total
Retrieved	150	1050	1200
Not retrieved	50	2750	2800
Total	200	3800	4000

Tortoise	Relevant	Not relevant	Total
Retrieved	120	40	160
Not retrieved	80	3760	3840
Total	200	3800	4000

From these we can evaluate the performance of each system on this single query.

$$\textit{Hare}\text{ precision } P = \frac{150}{1200} = 1/8 = 0.125 \qquad \textit{Tortoise}\text{ precision } P = \frac{120}{160} = 3/4 = 0.75$$

$$\textit{Hare}\text{ recall } R = \frac{150}{200} = 3/4 = 0.75 \qquad \textit{Tortoise}\text{ recall } R = \frac{120}{200} = 3/5 = 0.6$$

This answers the question, which asks "Tabulate ... calculate ... Show your working". In particular, "tabulate" means to write out a table — so you need a table. Here, this *contingency table* with row and column totals makes it straightforward to calculate the answers: for each of them we need only pick out two values from the table and divide one by the other.

When this question appeared in an exam, a few students used a different table showing "True" and "False" against "Positive" and "Negative". That turns out not to be so useful in calculating precision and recall. It's possible, but a bit more awkward as you'll find you need to add up a pair of diagonal elements rather than just picking out single values.

(g) The formula for calculating the harmonic mean has many equivalent presentations. One of the simplest is

$$H = \frac{2PR}{P+R}$$

but you could also use

$$H = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} \quad \text{or} \quad H = \frac{2}{\frac{1}{P} + \frac{1}{R}} \quad \text{or} \quad \frac{1}{H} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right).$$

Using these we can calculate the harmonic mean of precision and recall for *Hare* as $3/14 = 0.21$ and for *Tortoise* as $2/3 = 0.67$.