Exercise Set 3

Enrico Buratto

Question 1

The **boolean retrieval model** is the most typical model for information retrieval. It has many **advantages**: first of all, it has a very clean formalism, *i.e.* the rules used to write a query are clear. It is quite intuitive and easy to implement, and it can give a sense of control over the system to expert users, which have a precise understanding of their needs and the data they're working with. It is also good for applications, because a computer can easily handle thousands of results.

However, the boolean model has also many **disadvantages**: most users are, in fact, incapable of writing boolean queries, or it's just too much work to think of a reasonable query. It also suffer from the "feast or famine" problem: the results are often either too few or too many, thus it's hard for a normal user to find their way around.

In **ranked retrieval models**, as opposed to boolean retrieval models, the system returns an ordering over the documents in the collection for a query. The queries for this model are, therefore, one or more words in a human language instead of boolean expressions. The main **advantages** of this model are that the users are more capable of writing significant queries and that the "feast or famine" problem is no more a problem: the system just shows the top k results, thus the user is not overwhelmed.

Question 2

The idf, inverse document frequency, is defined as

$$idf_t = log_{10}\left(\frac{N}{df_t}\right)$$

where df_t is the document frequency of t, *i.e.* the number of documents that contain t. The idf of a term that occurs in every document is then **0**, because $N = df_t$ and $log_{10}(1) = 0$. This fact is very useful for the use of stop words, since these words are not significant in disntinguishing between one and another document.

Question 3

We can compute the **idf values** for each term using the formula reported in the previous answer. The idf values are, then:

- car: $idf_t = log_{10}(806791/18165) = \mathbf{1.6475}$
- auto: $idf_t = log_{10}(806791/6723) = \mathbf{2.0792}$
- insurance: $idf_t = log_{10}(806791/19241) = \mathbf{1.6225}$
- best: $idf_t = log_{10}(806791/25235) = \mathbf{1.5048}$

The tf.idf weight of a term can be defined in different ways. For instance, in the slides by Manning and Raghavan it is defined as follows:

$$W_{t,d} = log_{10}(1 + tf_{t,d}) * log_{10}\left(\frac{N}{df_t}\right)$$

where $tf_{t,d}$ is the term frequency of t in document d. The **tf.idf weights** using this formula are, then:

- **Doc 1**
  - car: $W_{t,d} = log_{10}(1+27) * 1.6475 = 2.3842$
  - auto: $W_{t,d} = log_{10}(1+3) * 2.0792 = 1.2518$
  - insurance: $W_{t,d} = log_{10}(1+0) * 1.6225 = 0$
  - best: $W_{t,d} = log_{10}(1+14) * 1.5048 = 1.7698$
- **Doc 2**
  - car: $W_{t,d} = log_{10}(1+4) * 1.6475 = 1.1516$
  - auto: $W_{t,d} = log_{10}(1+33) * 2.0792 = 3.1843$
  - insurance: $W_{t,d} = log_{10}(1+33) * 1.6225 = 2.4848$
  - best: $W_{t,d} = log_{10}(1+0) * 1.5048 = 0$
- **Doc 3**
  - car: $W_{t,d} = log_{10}(1+24) * 1.6475 = 2.3031$
  - auto: $W_{t,d} = log_{10}(1+0) * 2.0792 = 0$
  - insurance: $W_{t,d} = log_{10}(1+29) * 1.6225 = 2.3966$
  - best: $W_{t,d} = log_{10}(1+17) * 1.5048 = 1.8889$

However, we can also consider the term frequency not logaritmically scaled, as also the book does; in this case, the td.idf weights are:

- **Doc 1**
  - car: $W_{t,d} = 27 * 1.6475 = \mathbf{44.4825}$
  - auto: $W_{t,d} = 3 * 2.0792 = \mathbf{6.2376}$
  - insurance: $W_{t,d} = 0 * 1.6225 = \mathbf{0}$
  - best: $W_{t,d} = 14 * 1.5048 = \mathbf{21.0672}$
- **Doc 2**
  - car: $W_{t,d} = 4 * 1.6475 = \mathbf{6.59}$
  - auto: $W_{t,d} = 33 * 2.0792 = \mathbf{68.6136}$
  - insurance: $W_{t,d} = 33 * 1.6225 = \mathbf{53.5425}$
  - best: $W_{t,d} = 0 * 1.5048 = \mathbf{0}$
- **Doc 3**
  - car: $W_{t,d} = 24 * 1.6475 = \mathbf{39.54}$
  - auto: $W_{t,d} = 0 * 2.0792 = \mathbf{0}$
  - insurance: $W_{t,d} = 29 * 1.6225 = \mathbf{47.0525}$
  - best: $W_{t,d} = 17 * 1.5048 = \mathbf{25.5816}$

Question 4

In order to study the logarithm base change we can use the change of base formula, that is defined as follows:

$$log_b(a) = \frac{log_x(a)}{log_x(b)}$$

so in our case we have that, if we want to change base from 10 to b:

$$idf_t = log_b\left(\frac{N}{df_t}\right) = log_b(10) * log_{10}\left(\frac{N}{df_t}\right)$$

The tf.idf score is the sum of the tf.idfs of all the terms, so changing the base just changes the score by a constant factor.

The relative scores of two documents on a given query changes with the base, but the proportion between them remains the same so it doesn't affect anything.

Question 5

In order to calculate the normalized euclidean document vectors we just need to calculate the L_2 norm, which is defined as:

$$||\bar{x}||_2 = \sqrt{\sum_i x_i^2}$$

and divide every tf.idf score by it. The vectors are different if we use the logarithmic scaling; for this reason, here are calculated only the normal ones (not scaled). However, the procedure is the same.

The L_2 norms are:

- Doc 1: $\sqrt{44.4825^2 + 6.2376^2 + 0^2 + 21.0672^2} = 49.6128$
- Doc 2: $\sqrt{6.59^2 + 68.6136^2 + 53.5425^2 + 0^2} = 87.2815$
- Doc 3: $\sqrt{39.54^2 + 0^2 + 47.0525^2 + 25.5816^2} = 66.5715$

And so the euclidean normalized vectors are:

- Doc 1: $(\mathbf{0.8966}, \mathbf{0.1257}, \mathbf{0}, \mathbf{0.4246})$
- Doc 2: $(\mathbf{0.0756}, \mathbf{0.7861}, \mathbf{0.6134}, \mathbf{0})$
- Doc 3: $(\mathbf{0.5939}, \mathbf{0}, \mathbf{0.7068}, \mathbf{0.3842})$

Question 6

Case a

With the weight of a term t equals to 1 if present in the query q and 0 otherwise, we have that:

- car: $W(t, q) = 1$
- auto: $W(t, q) = 0$
- insurance: $W(t, q) = 1$
- best: $W(t, q) = 0$

So, applying the product between the weight and the term weights we have that:

	Doc 1	Doc 2	Doc 3
car	0.8966	0.0756	0.5939
auto	0	0	0
insurance	0	0.6134	0.7068
best	0	0	0

And then we sum these products for every document:

- $Score(q, Doc1) = 0.8966 + 0 + 0 + 0 = 0.8966$
- $Score(q, Doc2) = 0.0756 + 0 + 0.6134 + 0 = 0.689$
- $Score(q, Doc3) = 0.5939 + 0 + 0.7068 + 0 = 1.3007$

So the ranking is **d3,d1,d2**.

Case b

This time we need first calculate the weights; in order to do that we can just follow the same procedure we did previously for weights, but this time with the idf. So we have that the norm is equal to $3,4530$, and then the weights are as follows:

	Weight
car	0.4778
auto	0.6021
insurance	0.4699
best	0.4358

Using the same table of Case a, we have the following situation:

	Doc 1	Doc 2	Doc 3
car	0.4284	0.0361	0.2838
auto	0	0	0
insurance	0	0.2882	0.3321
best	0	0	0

Then the scores are:

- $Score(q, Doc1) = 0.4284$
- $Score(q, Doc2) = 0.3243$
- $Score(q, Doc3) = 0.6159$

So the ranking is still **d3,d1,d2**.