Introduction to Data Science

Course 094201

Lab 9 &10:

Term weighting for textual classification and retrieval

Spring 2017

First Part (Lab 9)

Goal:

- To improve Lab 8 results!

Means:

1. tf based representation.

2. tf*idf based representation.

3. Standardization of terms – by transforming to lowercase.

4. Stopwords removal.

5. Using cosine similarity instead of Euclidian distance.

The dataset and the code (reminder)

- Sentiment analysis: The process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within a mention.
- In our case each line contains amazon products reviews and a **class** (0 for **negative tone** and 1 for **positive tone**) separated by a tab.
- **Examples**:
  - I love this thing!	1
  - VERY DISAPPOINTED.	0
- Our goal is to use the Rocchio classifier in order to predict whether a given sentence represents a positive tone or a negative tone.

tf based representation

- $tf_{t,d}$ is the number of occurrences of a term *t* in a document *d*
- While there is a large difference between 0 and 1, the increase in importance of this signal with respect to the topic is not growing linearly
- *tf v*ariants:
  1. Raw count of term *t* in document *d*
  2. ***wf (implement this variant)***

$$wf_{t,d} = 0 \text{ if } tf_{t,d} = 0,\ 1 + \log tf_{t,d} \text{ otherwise}$$

idf

- One of the most important measures of informativeness of a term: its rarity across the whole corpus
  - Widely used in practice in different IR applications today
- Variant 1:

  inverse of the raw count of number of documents the term occurs in ($idf_i = 1/df_i$)
- Variant 2 (widely used):

$$idf_i = \log\left(\frac{n}{df_i}\right)$$

  where *n* is the total number of documents in the corpus

*tf*idf* based representation

- Assign a *tf*idf* weight to each term *i* in each document *d*

$$\boxed{w_{i,d} = tf_{i,d} \times \log(n / df_i)}$$

$$tf_{i,d} = \text{frequency of term } i \text{ in document } d$$

$$n = \text{total number of documents}$$

$$df_i = \text{the number of documents that contain term } i$$

- Increases with the number of occurrences *within* a doc
- Increases with the rarity of the term *across* the whole corpus

Text pre-processing

1. **Lowercase: change all the words in the documents to lower case letters.**

2. **Remove punctuation marks.**

3. Stopwords are extremely common words that can be considered noise. E.g.: *the*, *and*, *or*. **Stopword removal** reduces the dimension of the vectors.

The file "stop_words.txt" contains a list of stop words, use it in order to remove stopwords from the documents.

Use cosine similarity

- A vector can be *normalized* (given a length of 1) by dividing each of its components by its length – here we use the L_2 norm

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

- This maps vectors onto the unit sphere: $|| \vec{d}_j ||_2 = \sqrt{\sum_{i=1}^{n} w_{i,j}^2} = 1$
- There is no bias towards longer documents:

Dot product

Unit vectors

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}$$

Assignment For First Part (Lab 9)

1. Implement each of the improvement phases.

2. For each phase of improvement report:
   - Improvement over baseline (using the boolean model with Euclidian distance – Lab 8)
   - Improvement over the previous stage (for example how much did changing to lowercase improve the results in comparison to tf-idf representation)
   - Explain in your own words the reason for each change in performance

Second Part (Lab 10)

Goal:

- To experiment with ad-hoc retrieval and evaluation

- Write a script which receives as **input** 3 parameters:
  - K – number of documents to retrieve.
  - query – a requested query
  - query-representation method (see below): (1) or (2)

- The **output** is a ranked list of k documents ordered by decreasing values of cosine similarity between the query and the document
  - The document is represented using a *tf*idf* vector
  - The query is represented using:

  (1) a boolean vector (2) *tf*idf* based representation

- Please use the code of the previous labs

Assignment For Second Part (Lab 10)

1. The format of the output is:
   document_id cosine_similarity document_text

2. Run the script and produce the requested output for each of the queries in the file. Parameters’ values: 20 documents, query representation methods 1 and 2.

   The naming convention of the output files:

   “Output_”queryID”_”methodID

   where queryID is the ID of the query in the file and methodID is either 1 or 2

3. For each output compute precision@5 (p@5). To do so you need to judge yourself which of the top retrieved documents are relevant. Remember: the judgment is based on the information need, not on the query. Therefore, use the queries file, where the information need is specified

Lab 10 – contd.

4. Report which query terms weighting method resulted in better retrieval based on average p@5. Explain your calculations.
5. For the last query “good camera” please calculate recall and average-precision. To do so we need to find all the documents relevant for the information need in the corpus.

   Use grep (via the terminal) to identify all documents which contain the word *camera* and describe some good features (qualities) of a camera, then create a list of such documents (qrels file). Use this list to calculate recall@20 and average-precision@20 for the output of the best method you found in the previous question. Show your calculations in detail.

Assignment For Second Part (Lab 10)

Example:

Input: 10 Great product

```
1 .  doc568  -  Great Product.  Score:  1.0
2 .  doc768  -  Great product.  Score:  1.0
3 .  doc397  -  Great product and price.  Score:  0.736444183991
4 .  doc556  -  Great product for the price!.  Score:  0.526768386285
5 .  doc792  -  Great Phone.  Score:  0.500186948891
6 .  doc290  -  Great Phone.  Score:  0.500186948891
7 .  doc647  -  Great phone.  Score:  0.500186948891
8 .  doc896  -  Great phone.  Score:  0.500186948891
9 .  doc718  -  It was a great phone.  Score:  0.500186948891
10 .  doc971  -  Excellent product.  Score:  0.483219652836
```

Assignment For Second Part (Lab 10)

Example:

Input: 10 very high price buy something else

```
1 .  doc543  -  Don't buy this product.  Score:  0.27700916645
2 .  doc180  -  Don't buy this product.  Score:  0.27700916645
3 .  doc291  -  Don't buy it.  Score:  0.26934524838
4 .  doc303  -  Good price.  Score:  0.254515500537
5 .  doc894  -  This product is very High quality Chinese CRAP!!!!!!  Score:  0.219875562438
6 .  doc397  -  Great product and price.  Score:  0.216962526818
7 .  doc534  -  Great case and price!  Score:  0.202724388262
8 .  doc212  -  Great price also!  Score:  0.199595091105
9 .  doc645  -  Linksys should have some way to exchange a bad phone for a refurb unit or something!  Score:  0.199306210806
10 .  doc892  -  Excellent product for the price.  Score:  0.195823715594
```