CS824 Information Retrieval

Assignment 2

Name: Chengjun Shi
Student#: 200380163

Q1 Discuss the main ideas of indexing.

The most fundamental and difficult issue in information retrieval is indexing. The main goals of indexing is to analyze the set of documents and recognize the contents of documents. We have two basic approaches of indexing: one is manual indexing, the other one is automatic indexing. Documents could be anything which one may search for, and could contain information in different media forms. In such case, we make the following assumption, if a document contains a word, then it is considered relevant to the word. The number of appearances of a word is related to its importance. Then, a document after indexing is represented as a list of keywords (or index terms) and their frequencies. The final result of indexing is a "Term-Frequency" matrix, shortly, *TF* matrix.

Q2 Describe different term weighting methods.

In order to consider the importance of a term in describing a document, our task is to change the *TF* matrix into a weight matrix through statistic methods. We have the following term weighting methods:

1. Document frequency of a term t: The number of documents in which t appears, $df(t) = |\{TF(i,j) \neq 0 \mid 1 \leq i \leq n\}|$.
2. Row-wise normalization: relative frequency of different terms in the same document, $Pr(t|d) = \frac{TF(t,d)}{\sum_t TF(t,d)}$.
3. Column-wise normalization: relative importance of the same term in different documents, $Pr(d|t) = \frac{TF(t,d)}{\sum_d TF(t,d)}$.
4. Weight matrix: $W(t,d) = Pr(t|d) \times Pr(d|t)$.
5. *TF*IDF*: Term frequency * Inverse document frequency, wher $IDF = \frac{1}{\log df}$.
6. Information entropy of a term t: $H_B(t) = \log df(t)$.

Q3 Describe the basic idea of IDF. Provide an explanation of IDF based on Shannon's information entropy.

The inverse document frequency (*IDF*) is a measure of specificity of a term based on B matrix. The formula of *IDF* is:

$$IDF(t) = \frac{1}{\log df(t)}.$$

It represents how much information a term provides. As we discussed in class on text analysis, it is known that stop words may appear a lot of times in one document but they have little importance. Thus, it is necessary to weigh down the frequent terms and weigh up the important ones. That is how *IDF* works. If a term t is too common, then it cannot represent or specify a document correctly. The rare ones are good to distinguish relevant and non-relevant documents and terms. Shannon information entropy is a measure of information uncertainty and orderness, and it is typically defined in probabilistic framework. Events with a lower-probability value carry more information than the ones with a higher-probability value. The formula of Shannon entropy is:

$$\begin{aligned} H_B(t) &= df(t)(-\frac{1}{df(t)} \log \frac{1}{df(t)}) \\ &= \log df(t) \\ &= \frac{1}{IDF(t)} \end{aligned}$$

The common part between *IDF* and Shannon information entropy is that they are measures of the amount of information carried by a term. We may interpret *IDF* as a particular kind of probabilistic function.

Q4 Describe system evaluation measures based on binary relevance judgments, namely, a document is either relevant or nonrelevant.

Given an information retrieval system, we have the following Boolean model to evaluate the system. First, we denote the set of documents as D. Based on user needs and content relevance, we have a bi-partition $R \cup \overline{R} = D$, $R \cap \overline{R} = \emptyset$, where documents in R are considered relevant and documents in $\overline{R}$ are considered nonrelevant. In the given IR system, we have another bi-partition $Ret \cup \overline{Ret} = D$, $Ret \cap \overline{Ret} = \emptyset$. Documents in Ret are retrieved by the system and $\overline{Ret}$ contains the remaining documents. Two parameters are introduces to evaluate the system, namely, precision and recall:

$$\begin{aligned} Precision &= \frac{|R \cap Ret|}{|Ret|} \\ Recall &= \frac{|R \cap Ret|}{|R|}. \end{aligned}$$

F-measure is a combination of these two parameters:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

Q5 Use an example to draw a recall-precision graph.

To draw a recall-precision curve, I generate a dataset with 300 objects and 10 attributes for binary classification problem. The SVM classifier generated the following graph:

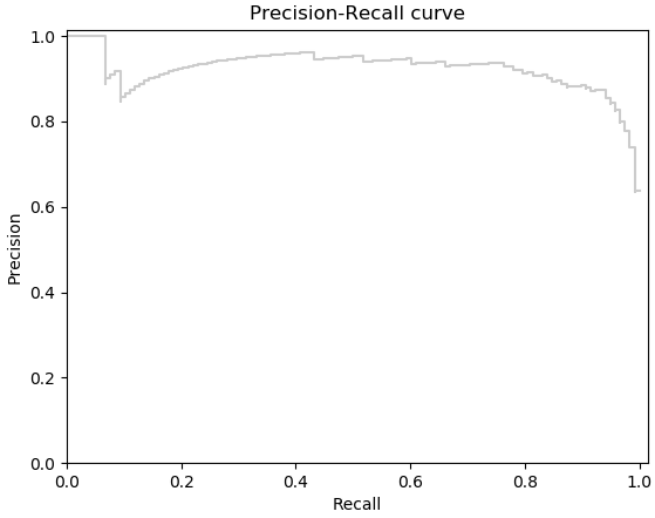

Figure 1:

Q6 Describe a method of representing relevance by using user preference relations, and system evaluation based on user preferences.

On the set of documents D, we have a user preference relation $\succ_u$ which is a weak order. Suppose we have two documents $d,\ d' \in D$, $d \succ_u d'$ if and only if user prefers d to d' or d is more relevant than d'. Induced by $\succ_u$, we have an equivalence relation $\sim_u$ and we say $d \sim_u d' iff \neg(d \succ_u d') \wedge \neg(d' \succ_u d)$. Since $\succ_u$ is a weak order, it must satisfy two properties, namely, asymmetry and negative transitivity. We can find a retrieval function $u : D \to \mathcal{R}$ such that $d \succ_u d' \iff u(d) > u(d')$. Then the retrieval function u reproduces the user ranking, and it produces a perfect ranking to represent relevance. In real design, we say u produces an acceptable ranking if $\forall d_1, d_2 \in D, d_1 \succ_u d_2 \implies u(d_1) > u(d_2)$.

From the above, we know that the retrieval function u produces an acceptable ranking on D. The evaluation of an IR system can be considered as how close is the user ranking to the system ranking, how close is $\succ_u$ to $\succ_s$. The distance of two rankings may be viewed as

a distance between two sets. By comparing user ranking and system ranking, the distance between two rankings on any one pair (d_1, d_2) can be classified into three cases:

$$\delta(d_1, d_2) = \begin{cases} 0, \text{ if user and system agree on } (d_1, d_2); \\ 1, \text{ if one is } \succ \text{ and the other one is } \sim; \\ 2, \text{ contradict each other.} \end{cases}$$

Then the total distance between $\succ_u$ and $\succ_s$ may be calculated by:

$$\begin{aligned} \beta(\succ_u, \succ_s) &= \sum_{d_1, d_2} \delta(d_1, d_2) \\ &= 2 * \text{the number of contradiction pairs} + \text{the number of compatible pairs} \\ &= 2 * |\succ_u{}^c \cap \succ_s| + |\succ_u \cap \sim_s|. \end{aligned}$$

Furthermore, we have normalized distance as:

$$\begin{aligned} n\beta(\succ_u, \succ_s) &= \frac{\beta(\succ_u, \succ_s)}{\beta(\succ_u, \succ_u{}^c)} \\ &= \frac{2 * |\succ_u \cap \succ_s{}^c| + |\succ_u \cap \sim_s|}{2 * |\succ_u|}. \end{aligned}$$