CS 674/INFO 630: Advanced Language Technologies	Fall 2007
Lecture 1 — August 28	
Lecturer: Lillian Lee	*Scribe: Nikos Karampatziakis & Ainur Yessenalina*

1 Notes on the General Subject Area

In the previous lecture we saw two examples of research in the computational analysis of (textual) natural language data. This single area (sometimes divided up into subareas called Information Retrieval (IR), Natural Language Processing (NLP), Computational Linguistics, etc.) is relevant to and can draw from different communities. For example,

- Databases, Digital Libraries and "Web Science". One challenge in this set of areas is that natural language data are unstructured or semi-structured. This means that we cannot index or represent them as efficiently as more traditional types of data such as employee records.
- Human-Computer Interaction. Natural language would be a natural modality in which to communicate with the computer.
- Linguistics and Psychology. Our field of study can provide information about language use "in the wild". Conversely, insights from linguistics and psychology should prove very relevant to our field of study.
- Machine Learning. Some very natural problems that arise in natural language processing can be formulated as interesting machine learning problems. For example, there is a lot of research going on in predicting complicated objects such as parse trees, translations, etc.
- Sociology. Natural language data capture a lot of the behavior and emotions of people on the web.

The focus of this course will be on research fundamentals, concepts, derivations and attitudes. We will be concerned with the following questions: how did something come about? How can it be improved?

2 Introduction to Information Retrieval

There are three dominant paradigms in "standard IR" (the canonical example is the task that Google has to do when you type in a query).

- The Vector Space Model (VSM). This approach embodies a successful empirical tradition.

- Probabilistic Retrieval (Robertson-Spärck Jones version). This model is theoretically motivated. It started at about the same time as the VSM and both of them have converged somewhat nowadays.

- Language Modeling. This approach has the same roots as probabilistic retrieval but embodies a different perspective.

2.1 Preliminaries

In classic (also known as ad-hoc) IR we ideally want to rank (some) documents in the corpus (document collection) C by relevance to the query q, which expresses the user's information needs. To evaluate the performance of an IR system we assume that there is an evaluation corpus. This consists of documents, queries and relevance judgements on a (document, query) basis. Some well-known publicly available corpora are the TREC datasets.

2.2 Per-query quality measures

Nowadays most users will only take a look at the first page of results from an IR system. Hence, people are much more interested in the precision of an IR system than in recall[1]. Thus, these days, two important per-query measures are the following:

- Precision at k (prec@k) is the percentage of k top-ranked documents (according to the system) that are truly relevant. The advantage of this performance measure is that it reflects how good the first page of results will be. However it is sensitive to the choice of k and it does not average well. This means that a small change in the ranking may bring about a big change in its score, and also that the possible values come at large intervals, especially when k is small.

- Average precision. Let $p = \{p_1, p_2, \ldots, p_r\}$, where r is the number of truly relevant documents, be the ranks given by the system to the truly relevant documents. Then the average precision is

$$\frac{1}{r}\sum_{i=1}^{r} \text{prec@}p_i$$

  For example, say we have four documents and two systems have produced the following rankings

[1]The recall of an IR system is a quality measure that is defined as the proportion of relevant documents that are retrieved, out of all relevant documents. Note that there are cases where recall is an appropriate measure, such as if the user's task is to find all patents that relate to a particular idea.

rank	system-1	system-2
1	relevant	not relevant
2	not relevant	not relevant
3	relevant	relevant
4	not relevant	relevant
avg-prec	$\frac{1}{2}(100\% + 66.6\%)$	$\frac{1}{2}(33.3\% + 50\%)$

The average precision of system-1 is greater than that of system-2 which reflects the fact that the first ranking is better that the second one. The advantage of this measure is that small changes in the ranking do not produce big changes in its score (it averages better). But the disadvantage of this measure is that it's more expensive and may give too much weight to the relevant documents that have low ranks (prec@k completely ignores them). In practice, usually both of these measures are reported.

3 The Vector Space Model

The vector space model was initially proposed by Gerry Salton (CS Professor here at Cornell). The model is simple, intuitive and empirically works well. Our purpose is to rank documents whose content best matches the query. This description however doesn't specify how we represent the documents and how we determine the best match.

3.1 Representation

We assume a fixed vocabulary $V = \{v_1, ...v_m\}$ of terms (proxies for the important concepts). This assumption already hides a lot of details under the rug. For example, is "dog" a term? (Yes, surely.) Is "data base" a term? (Less obvious: maybe it is two terms.) Is "ddddd" a term? (No? But what if it is someone's login?[2]) The answer depends on the particular application.

For every document $d \in C$ and for every $v_j \in V$ we compute a weight $d[j]$, measuring an estimate of the association between d and v_j. Hence, a document d is represented by the following vector $\vec{d}$

$$\vec{d} = \begin{pmatrix} d[1] \\ \vdots \\ d[m] \end{pmatrix} = (d[1], \ldots, d[m])^T \in \mathbb{R}^m.$$

To quantify the best match to a query q (which we also represent as a vector $\vec{q} \in \mathbb{R}^m$) we have to define a function $f : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ that assigns a score to every document-query pair. A natural choice is the inner product:

$$\vec{q} \cdot \vec{d} \triangleq \sum_{j=1}^{m} q[j] \times d[j].$$

[2]A search in Google for "ddddd" returns more than 1.5 million pages.

The inner product rewards documents that have a strong association with important query terms. Euclidean distance is not a good choice for a match function because it ignores the direction of the vectors. For example, assuming that the $d[j]$'s are based just on counts, a document that contains just "Google" is closer in Euclidean distance to the vector for a query "cat" than the vector for a document that consists of the term sequence "cat cat cat cat".

3.2 Term-weighting

Now we turn into the issue of setting the weight $d[j]$ for a document d and a term v_j. This is an important and open research problem, but there are three consensus points in the information retrieval community.

1. Corpus-distinctive terms are more important to match. One interpretation of this statement is that the user forms a query q to distinguish the relevant from the non-relevant documents so we should pay the most attention to distinguishing terms. Another view of this idea is that distinctive terms are likely to relate to important concepts. For example, suppose we have a query "computer modeling of neural processes". In a CS corpus, the word "neural" is more important than the other words. In a neurobiology corpus the word "computer" would be more important. We can conclude that the words that come up quite often are not really heplful for our task. Therefore we have to penalize somehow the words that appear in many documents. One way to do this is to use some notion of the *inverse document frequency* (IDF) of a term that varies inversely with the number of documents that contain the term.

2. The *term frequency* of v_j in document d ($TF_d(j)$) is correlated with some underlying document-concept association. If a term v_j appears in a document d many times, then the document is to some extent related to the concept that v_j represents. But ...

3. Some normalization $norm(d)$ is necessary e.g. to compensate for length bias. For example, consider a document that contains the word "cars" 100 times. Is it relevant to cars? Perhaps yes, if the document is, say, 300 words long, but what if it is 100000 words long?

The general formula for term weighting is thus

$$d[j] = \frac{TF_d(j) \times IDF(j)}{\text{norm}(d)}$$

but the exact functional forms of all three quantities have been intentionally unspecified.

4 Finger Exercise

1. Suppose we have two rankings which are identical except that in the first the j-th document is relevant and the $(j+1)$-th is not and in the second the j-th document is

not relevant and the $(j+1)$-th is relevant. Suppose further that there are r relevant documents in both rankings and p of them are ranked above position j. What is the difference in their precision at k for different values of k? What is the difference in average precision of the two rankings? How is it affected by p and j?

2. For ease of notation we represent a ranking as a string from the language $\{n, r\}^*$. For example the string $nrnnr$ represents a ranking of 5 documents where only the second and fifth are relevant. We also use c^j to denote j repetitions of character c. We will investigate the importance of getting the first document right. Say, we have the following rankings $A = rn^k r^{k-1}$ and $B = nr^k n^{k-1}$. How big should k be so that ranking B has better average precision than A? Repeat your calculations for rankings $C = n^4 rn^k r^{k-1}$ and $D = n^5 r^k n^{k-1}$.

3. Professor Martingale wants to use some probability theory for evaluating a ranking and he comes up with this. Assuming an infinite number of documents, the probability that the user of the IR system will select to view a document at rank k is given by

$$Pr[k\text{-th doc viewed}] = \frac{1}{2^k}$$

Assuming that the user looks at only one document, we can then define the expected observed (and one-time) relevance of a ranking r as:

$$E[Rel(r)] = \sum_k Pr[k\text{-th doc viewed}] r_k$$

where r_k takes the value 1 if the k-th document is relevant and 0 otherwise. Briefly explain why this measure may not be a satisfying way to evaluate a ranking.

4. What would be good way to define a probability distribution on the quality of a ranking (which may be thought as a random variable since different users have different needs), perhaps in terms of other things such as the actual relevances of the documents, the apparent relevance of the k-th snippet shown to the user, the user taking a look at the k-th snippet and any other factor you think it should be taken into account?

5 Solution

1. If $k \neq j$ then the difference is zero. For $k = j$ the first ranking will have prec@$k = \frac{p+1}{k}$ and the second will have prec@$k = \frac{p}{k}$ so their difference is $\frac{1}{k}$. Let $i = \{i_1, i_2, \dots, i_r\}$ and i' where $i'_\ell = i_\ell$ for $\ell \neq p+1$ and $i'_{p+1} = j+1$ be the ranks of the relevant documents in the first and second ranking respectively. The difference in average precision comes only from the j-th document. Indeed if we subtract the two average precision expressions

$$\frac{1}{r}\sum_{\ell=1}^{r} \text{prec@}i_\ell - \frac{1}{r}\sum_{\ell=1}^{r} \text{prec@}i'_\ell =$$

$$\frac{1}{r}\left(\sum_{\ell\neq p+1} \text{prec@}i_\ell - \sum_{\ell\neq p+1} \text{prec@}i'_\ell + \text{prec@}i_{p+1} - \text{prec@}i'_{p+1}\right)$$

the first two terms cancel and we get

$$\frac{1}{r}(\text{prec@}j - \text{prec@}(j+1)) = \frac{1}{r}\left(\frac{p+1}{j} - \frac{p+1}{j+1}\right) = \frac{p+1}{j+1}\frac{1}{rj}$$

We see that the change in average precision due to a swap at position j is proportional to how big p is relative to j and inversely proportional to j.

2. Probably the least tedious way to find the appropriate value of k is by writing a small script that tries different values of k. We can verify that for the first case for $k = 3$ we have the rankings $A = rn^3r^2$ $B = nr^3n^2$.

$$\text{avg-prec}(A) = \frac{1}{3}\left(1 + \frac{2}{5} + \frac{3}{6}\right) = 0.633$$

$$\text{avg-prec}(B) = \frac{1}{3}\left(\frac{1}{2} + \frac{2}{3} + \frac{3}{4}\right) = 0.638$$

For $k = 2$ we would get

$$\text{avg-prec}(A) = \frac{1}{2}\left(1 + \frac{2}{4}\right) = 0.75$$

$$\text{avg-prec}(B) = \frac{1}{2}\left(\frac{1}{2} + \frac{2}{3}\right) = 0.583$$

For C and D and for $k = 2$ we have the rankings $C = n^4rn^2r$ $D = n^5r^2n$

$$\text{avg-prec}(C) = \frac{1}{2}\left(\frac{1}{5} + \frac{2}{8}\right) = 0.225$$

$$\text{avg-prec}(D) = \frac{1}{2}\left(\frac{1}{6} + \frac{2}{7}\right) = 0.226$$

and for $k = 1$ it is obvious that C is better than D. We observe that even though the difference between rankings A and B and rankings C and D follows the same pattern, the latter is "easier" to compensate for because it happens further down the list of ranked documents.

3. The expected relevance measure is not good because it emphasizes the first few documents too much. For example, if the first document is not a relevant one, then no matter how good the rest of the ranking is, the expected relevance will not exceed $\frac{1}{2}$. In general, a mistake at position i cannot be compensated for even if all documents after position i are relevant.