2019 - 2020 Past exam paper Information Retrival

Jiarong Li 20230033

Computer Science in Artificial Intelligence
National University of Ireland, Galway
j.li11@nuigalway.ie

1 Q1

1.1 a.

Outline, with an appropriate example, a suitable indexing strategy to deal with Boolean queries. Discuss an approach for handling queries:

- (term1 OR term2)
- (term1 AND term2)

Outline an efficient approach to dealing with general Boolean queries.

Answer: Dealing with Boolean queries, we need to find out that if a document contains a term or a set of terms that satisfy the query. The model assumes terms are present or absent, hence term weights $w_{i,j}$ are binary and discrete, i.e., $w_{i,j}$ is an element of 0, 1 For instance, if the query terms are "java" and "coffee", we need to find out the set of documents which contain the most "coffee" and "java" terms.

We can use the disjunctive normal form to handle the queries above. If a document 'satisfies' any of the components, the document is deemed relevant and returned.

1.2 b.

Outline a suitable data structure to allow searching for the presence of terms and prefixes of terms in a passage of text. Illustrate how the following terms would be placed in such a data structure:

stand, stood, standing, door, star

Answer: Use a set of terms with stemming techniques. Stemming algorithms attempt to remove common suffixes from terms occurring in the documents. The overall goal is to reduce sim- ilar words to a common root form by identifying morphological derivations of words. There are many approaches in the literature and in commercial systems. Lovin's algorithm and Porter's stemmer are two of the well- known classic stemmers.

The terms should be placed as stand, door, star.

1.3 c.

Normalisation is often used in term weighting schemes. Explain the term normalisation in this context and with reference to any well-known weighting scheme discuss any approach to normalisation. (7 marks)

Answer: tf

For all terms in a document, the can be weight assigned is calculated by: $w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$, where $f_{i,j}$ is the normalised frequency of term t_i in document d_j, N is the number of documents in the collection, n_i is the number of documents that contain term t_i.

Maximum term normalisation. Usually of the form: $ntf = a + (1-a) \times \frac{tf_{i,d}}{tfmax(d)}$, where a is a smoothing factor which can be used to dampen the impact of the second term.

Problems with maximum term normalisation:

(1). Stopword removal may have effects on distribution of terms; this normalisation is unstable and may require tuning per collection, (2). Possibility of outliers with unusually high frequency, (3). Those documents with a more even distribution of term frequencies should be treated differently to those with a skewed distribution

Normalisation because we observe higher frequencies in longer documents merely because longer documents tend to repeat same words more frequently. For instance, consider a document d‘ created by concatenating a document d to itself. d‘ is no more relevant to any query than document d, but yet according to vector space type similarity, $sim(d', q) >= sim(d, q) \forall q$.

Structure of modern weighting schemes:

Many, if not all, of the developed or learned weighting schemes can be epresented in the following format: $sim(q, d) = \sum_t \in q \wedge d(ntf(D) \times gw_t)(C) \times qw_t(Q)$, where $ntf(D)$ is the normalised term frequency in a document, $gwt(C)$ is the global weight of a term across a collection $qwt(Q)$ is the query weight of a term in Q, a query.

Pivoted Normalisation: Standard benchmark, Needs to be tuned for collection, Issues with normalisation

We can view most weighting schemes as comprising local factors, global factors (collection wide) and query related features. Many, if not all, of the developed or learned weighting schemes can be represented in the following format: $sim(q, d) = \sum_{t \in q \wedge d}(ntf(D) \times gw_t(C) \times qw_t(Q))$, where $ntf(D)$ is the normalised term frequency in a document $gwt(C)$ is the global weight of a term across a collection $qwt(Q)$ is the query weight of a term in Q, a query There have been many approach to term weighting that fits this template. Many term weighting schemes, normalisation schemes, document normalisation and global (idf-like) approaches have been developed and empirically investigated on collections.

For each document d, let tfmax(d) be the maximally occurring term in the document. The normalized term frequency can then be calculated as: $ntf_{i,j} = a + (1 - a) \times \frac{tft_{i,j}}{tfmax(d)}$, where a is a value between 0 and 1. The term a is a smoothing term to weaken the contribution of the second term.

This is a good method to penalise long documents but does suffer from some problems.

2 Q2

A company has a large set of scientific articles (each of which contains a title, abstract, authors, key words, year of release, main body of the paper and a bibliography).

2.1 a

Suggest a means to measure the similarity between two documents based on:

- Content of the document
- Authors.
- Bibliographies.
- Content, authors and bibliographies (8 marks)

Answer:

citation analysis. Measure of similarity of documents (Kessler in 1963). The bibliographic coupling of two documents A and B is the number of documents cited by both A and B. Co-citation If a paper cites two papers A and B, then they are related or associated. (websearch)

2.2 b

The company wishes to rank all papers that are relevant to a given query and to then re-order the papers in the answer set according to how authoritative or influential the papers are. Outline an approach that could be used to give a suitable solution for this requirement. (10 marks)

Answer:

Usually wish to select the most “authoritative” pages. Hence the search entails identifying pages that have relevancy and quality. In addition to content, web-pages also contain many links that connect one page to another. This web-structure contains implicitly a large number of human annotations which can be exploited to infer notions of authority (and by extension quality). Any link to a page, p, is a positive recommendation for that page p.

The HITS algorithm analyses hyper-links to identify: Authoritative pages (best sources); Hubs (collections of links)

HITS Algorithm:

Computes hubs and authorities for a particular topic specified by a normal query. First determines a set of relevant pages for the query called the base set S. Analyse the link structure of the web sub-graph defined by S to find authority and hub pages in this set.

There exists problems with identifying authoritative pages: (1)authoritative pages do not necessarily refer to themselves as such, (2) many links are purely for navigational purposes, (3) advertising links.

HITS Update Rules:

Authorities are pointed to by lots of good hubs: $a_p = \sum_{q:q->p} h_q$

Hubs point to lots of good authorities: $h_p = \sum_{q:p->q} a_q$

Mutually recursive heuristics used: a good "authority page" is one which is pointed to by a number of sources. a good "hub" is one that contains many links

2.3 c

Outline a suitable approach to cluster these documents in the collection into useful sub clusters that may be of use in user search tasks. Briefly list and limitations of the approach.

The HITS algorithm analyses hyper-links to identify: Hubs (collections of links)

HITS Algorithm:

Computes hubs and authorities for a particular topic specified by a normal query. First determines a set of relevant pages for the query called the base set S. Analyse the link structure of the web sub-graph defined by S to find authority and hub pages in this set.

There exists problems with identifying authoritative pages: (1)authoritative pages do not necessarily refer to themselves as such, (2) many links are purely for navigational purposes, (3) advertising links.

Constructing the Subgraph For a specific query Q, let the set of documents returned by a standard search engine (e.g. vector space approach) be called the root set R.

Step: (1) Initialize S to R. (2) Add to S all pages pointed to by any page in R. (3) Add to S all pages that point to any page in R.

limitations:

(1). on narrowly focussed query topics, there may not be many exact references and the hubs may provide links to more general pages (2). potential drift from main topic. All links are treated as being equally important. If there is a range of topics in a hub, the focus of the search may drift (3). timeliness of recommendation is hard to identify (4). sensitivity of malicious attack (5). edges with wrong semantics

3 Q3

3.1 a

The Boolean model doesn't consider term weights in queries, and the result set of a Boolean query is often either too small or too big. The idea of the extended model is to make use of partial matching and term weights as in the vector space model. It combines the characteristics of the Vector Space Model with the properties of Boolean algebra and ranks the similarity between queries and documents.

In the Extended Boolean model, a document is represented as a vector (similarly to in the vector model). Each i dimension corresponds to a separate term associated with the document.

The weight of term K_x associated with document d_j is measured by its normalized Term frequency and can be defined as: $w_{x,j} = f_{x,j} \times \frac{Idf_x}{max_i idf_i}$

3.2 b

Given a ranked answer set and evaluation judgements, explain how you would generate a precision-recall graph. (5 marks)

We can chose a threshold (top k documents, all documents above a certain score) and use that to define a set and then define precision and recall accordingly. However, we obtain a more accurate representation of the quality of the ranking by plotting precision against recall for a number of points. We can calculate these pairs of values for different points along the ranked list. To illustrate the development of a precision-recall graph, consider the fol- lowing 'toy' scenario. Given a document collection of size 20 (—D— = 20) which contains 10 relevant documents for a query ($|R| = 10$) and the top 10 ranked documents in the answer as follows. $d1, d2, d3, d4, d5, d6, d7, d8, d9, d10$ Also, assume that given the human judgements that those marked in bold are the relevant ones (i.e $d1, d2, d4, d5, d9$). Considering the first document only, we can calculate the a precision-recall pair. The precision-recall pair is $(1, 0.1)$. Note if we are considering the first document, hence $|A| = 1$. Considering the first two documents only, we can again calculate another precision-recall pair - $1, 0.2$. Our precision is still 1 (2/2) and the recall in- creases to 0.2 (2/10). For the first three documents we have a precision-recall pair $(0.67, 0.2)$. As we have encountered an non-relevant document, our precision falls and our recall remains the same. We continue and generate a set of points and the plot precision versus recall.

3.3 c

Learning mechanisms has been used successfully in information retrieval. Using an approach of our choice, suggest a learning mechanism to identify good weights to apply to terms in documents. Discuss any limitations of your approach (10 marks)

Clustering mechanism with Genetic Algorithms. Uses operators based on crossover and mutation as the basis of the algorithm to sample space of solutions. The effectiveness of an IR system is dependent on the quality of the weights assigned to terms in documents. We have a definition of relevant and non-relevant documents; can use MAP or precision@k as fitness. Each genotype can be a vector of length N (the size of the lexicon). Set all rates randomly initially. Run system with a set of queries to obtain fitness; select good chromosomes; crossover; mutate. Effectively searching landscape for weights to give good ranking.

Learning mechanisms and feedback mechanisms are implicit in many other IR models - neural and probabilistic approaches incorporate the notion of feed- back in their models. There are many issues and problems related to obtaining use feedback; this is due to a number of reasons: • Users tend not to give a high degree of feedback • Users are typically inconsistent with their feedback • Explicit user feedback does not have to be strictly binary (a range of values may be more appropriate)

4 Q4

4.1 a

Query expansion is often used to improve the performance of information retrieval systems. Outline an approach to suggesting terms to add to a query in the absence of user feedback. (8 marks)

Answer:

Document Set Analysis: Local Analysis, Global Analysis(Global analysis involves performing an analysis on the whole document collec- tion and not just the returned set. The degree of correlation between two terms is calculated based on their occurrences across the whole collection. The weights assigned to each term-document value is based on the term weight assigned by our weighting scheme. It is then possible to calculate similarity between any two terms by taking some measure of similarity between their two vectors (cosine measure). We can use this to expand our queries. clustering of documents and on word embeddings).

In local analysis, documents which are retrieved are examined at query time to determine terms for query expansion. Most systems typically involve the devel- opment of some form of term-term correlation matrix to quantify the connection between two terms. This correlation can be calculated over the whole document set or in the context of the current query. The current query is then expanded by including terms correlated to the query terms. Different means are available to develop the correlations. (1). Association Clusters (2). Metric Clusters. (3). Scalar Clusters. Scalar clusters represent an approach which tries to improve upon the previous approaches by attempting to pay attention to the context in which a term occurs. The neighbourhood of terms surrounding a term is considered when calculating the correlation between these terms. It is based on the heuristic that if two terms have similar neighbourhoods then there is a high correlation between terms. The size of the neighbourhood to be considered can be taken as a fixed sized set of terms either side of the term in the question or can be taken as the current sentence of paragraph. Similarity can be based on comparing the two vectors representing the neigh- bourhoods using the cosine similarity measure. This measure can be used to define term-term correlation matrix and the procedure continues as before.

4.2 b

Collaborative filtering systems often struggle to make recommendations to new users to a system (as the new users will not have given any recommendations). Propose and approach to help overcome this problem. (7 marks)

Answer:

The data sparsity challenge appears in several situations, specifically, the cold start problem occurs when a new user or item has just entered the system, it is difficult to find similar ones because there is not enough information (in some literature, the cold start problem is also called the new user problem or new item problem [21, 22]). New items cannot be recommended until some users rate it, and new users are unlikely given good recommendations because of the lack of their rating or purchase history.

To alleviate the data sparsity problem, many approaches have been proposed. Dimensionality reduction techniques, such as Singular Value Decomposition (SVD) [23], remove unrepresentative or insignificant users or items to reduce the dimensionalities of the user-item matrix directly. The patented Latent Semantic Indexing (LSI) used in information retrieval is based on SVD [24, 25], in which similarity between users is determined by the representation of the users in the reduced space. Goldberg et al. [3] developed eigentaste, which applies Principle Component Analysis (PCA), a closely-related factor analysis technique first described by Pearson in 1901 [26], to reduce dimensionality. However, when certain users or items are discarded, useful information for recommendations related to them may get lost and recommendation quality may be degraded [6, 27].

Hybrid CF algorithms, such as the content-boosted CF algorithm [16], are found helpful to address the sparsity problem, in which external content information can be used to produce predictions for new users or new items. In Ziegler et al. [28], a hybrid collaborative filtering approach was proposed to exploit bulk taxonomic information designed for exact product classification to address the data sparsity problem of CF recommendations, based on the generation of profiles via inference of super-topic score and topic diversification [28]. Schein et al. proposed the aspect model latent variable method for cold start recommendation, which combines both collaborative and content information in model fitting [29]. Kim and Li proposed a probabilistic model to address the cold start problem, in which items are classified into groups and predictions are made for users considering the Gaussian distribution of user ratings [30].

4.3 c

The majority of information retrieval systems return a ranked list of results for a given query. Propose an alternative means to present the return set that:

- shows the relationship between the query and the returned documents
- shows the relationship between the returned documents.

cosine-similarity.