Information Retrieval (5LN712)

Scoring, Term Weighting, and the Vector Space Model

Ali Basirat

Department of Linguistics and Philology
Uppsala University

March 6, 2020

Table of contents

Limitations of Boolean Retrieval[1]

- A document either matches or does not match a query
- Good for expert users with precise understanding of their needs and of the collection
- Not good for the majority of users
- Boolean queries often result in either too few (=0) or too many (1000s) results.
- Most users don't want to wade through 1000s of results returned by a Boolean retrieval system
- In Boolean retrieval, it takes a lot of skill to come up with a query that produces a manageable number of hits.
- Most users are not capable of writing Boolean queries

[1] Based on the Hinrich Schütze's slide

Table of contents

Ranked Retrieval

- To rank documents based on their relevance to a query
- With ranking, large result sets are not an issue

Relevance Ranking

- Assign a score to each query-document pair
- The score measures how well the document is relevant to the query
- More relevant documents are expected to get higher ranks than the less relevant documents
- We would like to assign each query-document pair a score in $[0, 1]$
- This score tells us how well the query and document match
- Once the query-document pairs are scored, we sort them

Structure of digital documents

- A digital document is not only a sequence of terms
- It may encode metadata such as authors, title, format, publication date, etc.
- The contribution of different fields may differ with information need
- The contribution of fields are weighted

Table of contents

Weighted zone scoring

To assign weights to pairs of (query, document)

- If the document has l zones
- Let $g_1, \ldots, g_l \in \mathbb{R}$ s.t. $\sum_{i=1}^{l} g_i = 1$ be the weight of each zone
- Let $s_i \in \{0, 1\}$ be the Boolean score between a query q and the ith zone
- The weighted zone score is $\sum_{i=1}^{l} s_i g_i$

Weighted zone scoring

If the query q is a two term query $q = (q_1 \wedge q_2)$ then the algorithm below computes the weighted zone scores

```
ZONESCORE(q1, q2)
 1  float scores[N] = [0]
 2  constant g[ℓ]
 3  p1 ← postings(q1)
 4  p2 ← postings(q2)
 5  // scores[] is an array with a score entry for each document, initialized to zero.
 6  // p1 and p2 are initialized to point to the beginning of their respective postings.
 7  //Assume g[] is initialized to the respective zone weights.
 8  while p1 ≠ NIL and p2 ≠ NIL
 9  do if docID(p1) = docID(p2)
10        then scores[docID(p1)] ← WEIGHTEDZONE(p1, p2, g)
11             p1 ← next(p1)
12             p2 ← next(p2)
13        else if docID(p1) < docID(p2)
14                then p1 ← next(p1)
15                else p2 ← next(p2)
16  return scores
```

Figure: Algorithm for computing the weighted zone score from two postings lists. Function WEIGHTEDZONE computes $\sum_{i=1}^{l} s_i g_i$

Learning Zone Weights

The zone weights $g_1, \ldots, g_L \in \mathbb{R}$ can be set by

- An expert that knows about the importance of each zone
- An algorithm that learns the weights from some training examples
  - The training examples are tuples of (query, document, relevance judgment)

$$\Phi_j = (q_j, d_j, r(d_j, q_j)) \quad j = 1, \ldots, n$$

  - The relevance judgements are usually collected by experts
  - A training algorithm sets the zone weights g_i such that the score of each pair q_j, d_i is close to the relevance judgement $r(d_j, q_j)$

Learning Zone Weights

Documents with two zones (title, body)

- Let $s_T(d, q)$ and $s_B(d, q)$ be Boolean variables indicating whether the title and the body zones of d match the query q
- Let $g \in [0, 1]$ be the weight of the title zone

$$s(d, q) = g s_T(d, q) + (1 - g) s_B(d, q)$$

- We look for a value of g that makes the values of $s(d, q)$ very close to the relevance judgements provided in a training data

Learning Zone Weights

If the training set contains n examples $q_j, d_j, r_j = r(d_j, q_j)$ $j = 1, \ldots, n$

- For a given value of g, the error of the scoring function for each training example is

$$\epsilon_j(g) = (r(d_j, q_j) - s(d_j, q_j))^2$$

- The total error is $E = \sum_{j=1}^{n} \epsilon_j(g)$
- Our goal is to minimize the total error E with respect to g

Learning Zone Weights

since s_T and s_B are Boolean variable, for each value of g the score function $s(d, q)$ have only four possibilities

s_T	s_B	$s(d,q) = gs_T(d,q) + (1-g)s_B(d,q)$
0	0	0
0	1	$1-g$
1	0	g
1	1	1

Learning Zone Weights

- Let n_{ijr} be the number of *relevant* training examples for which $s_T = i$ and $s_B = j$ $(i, j \in \{0, 1\})$
- Let n_{ijn} be the number of *non-relevant* training examples for which $s_T = i$ and $s_B = j$ $(i, j \in \{0, 1\})$
- Let r be the relevance judgement

s_T	s_B	r	$s(d, q)$	ϵ	n	$e = n\epsilon$
0	0	0	0	0	n_{00n}	0
0	0	1	0	1	n_{00r}	n_{00r}
0	1	0	$1 - g$	$(1 - g)^2$	n_{01n}	$n_{01n}(1 - g)^2$
0	1	1	$1 - g$	g^2	n_{01r}	$n_{01r}g^2$
1	0	0	g	g^2	n_{10n}	$n_{10n}g^2$
1	0	1	g	$(1 - g)^2$	n_{10r}	$n_{10r}(1 - g)^2$
1	1	0	1	1	n_{11n}	n_{11n}
1	1	1	1	0	n_{11r}	0

Learning Zone Weights

- The total training error is:

$$E = (n_{01n} + n_{10r})(1 - g)^2 + (n_{01r} + n_{10n})g^2 + n_{00r} + n_{11n}$$

- Differentiating E with respect to g and setting the result equal to zero, we get:

$$g = \frac{n_{10r} + n_{01n}}{n_{10r} + n_{10n} + n_{01r} + n_{01n}}$$

Table of contents

Term Frequency and Document Ranking

- A document that mentions the terms of a query more often has more to do with the query
- Such a document should receive a higher rank

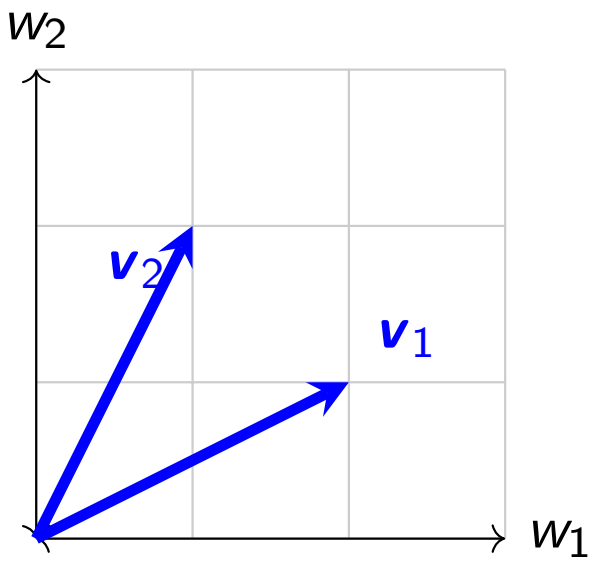
Term Frequency

- $\text{tf}_{d,t}$: the number of occurrence of term t in document d
- A document can be summarized by a list of term-frequencies
- The list $[\text{tf}_{d,t_1}, \ldots, \text{tf}_{d,t_n}]$ is viewed as a quantitative digest of document d
- This representation is considered as a bag-of-word representation of d
- The bag-of-word representation of document d is a real-valued vector whose elements are proportional to the frequency of a list of words in the document
- In this representation, all terms are considered equally important

Term Frequency

Example

- Let's the vocabulary set be $\{w_1, w_2\}$.
- **Doc1**: $w_1 w_2 w_1 \rightarrow \mathbf{v}_1 = [2, 1]$
- **Doc2**: $w_2 w_2 w_1 \rightarrow \mathbf{v}_2 = [1, 2]$



Inverse Document Frequency

- All terms are not equally important
- Some terms have little or no discrimination power in determining the relevance of a document
- The terms that are uniformly distributed in many (all) documents should contribute less to the relevance determination

Example

Function words that are frequently seen in almost all documents do not have strong discrimination power

Inverse Document Frequency

- df_t: The document frequency of the term t is the number of documents that contain t
- Terms that have high df are less likely to be important for determining documents relevance
- We give lower weights to the terms with high value of df
- Inverse document frequency: $\text{idf}_t = \log \frac{N}{\text{df}_t}$
- N is the total number of documents in the collection
- Question: what is the idf of a term that is seen in all documents?

Inverse Document Frequency

Example

What is the idf of the terms forming the following collection?

- **Doc1**: the viruses spread quickly .
- **Doc2**: the man died .
- **Doc3**: the news spread quickly .

term	N	df	idf
the	3	3	$\log \frac{3}{3} = 0.00$
viruses	3	1	$\log \frac{3}{1} = 0.48$
spread	3	2	$\log \frac{3}{2} = 0.40$
quickly	3	2	$\log \frac{3}{2} = 0.40$
man	3	1	$\log \frac{3}{2} = 0.48$
died	3	2	$\log \frac{3}{2} = 0.48$
news	3	2	$\log \frac{3}{2} = 0.40$
.	3	3	$\log \frac{3}{3} = 0.00$

Table: Caption

Inverse Document Frequency

- The idf of a rare term is high.
- The idf of a frequent term is likely to be low.
- The idf of functional words is low

tf-idf Weighting

- The tf-idf of a term t in a document d is:

$$\text{tf-idf}_{d,t} = \text{tf}_{d,t} \times \text{idf}_t$$

- Highest: when t occurs many times in a small number of documents
- Lower: when the term occur fewer times in many documents
- Lowest: when the term occur in all documents

Inverse Document Frequency

Example

What is the idf of the terms forming the following collection?

- **Doc1**: the viruses spread quickly .
- **Doc2**: the man died .
- **Doc3**: the news spread quickly .

term	idf	tf-idf(d1)		tf-idf(d2)		tf-idf(d2)	
the	0.00	1	0.00	1	0.00	1	0.00
viruses	0.48	1	0.48	0	0.00	0	0.00
spread	0.40	1	0.40	0	0.00	1	0.40
quickly	0.40	1	0.40	0	0.00	1	0.40
man	0.48	0	0.00	1	0.48	0	0.00
died	0.48	0	0.00	1	0.48	0	0.00
news	0.40	0	0.00	0	0.00	1	0.40
.	0.00	1	0.00	1	0.00	1	0.00

Table: Caption

tf-idf for Document Representation

- Each document d is represented by a vector $[\text{tf-idf}_{d,t_1}, \ldots, \text{tf-idf}_{d,t_n}]$
- This is still a bag-of-word representation of the document
- Since the order of the terms are not taken into account
- What is the dimensionality of the vector space?

Document Similarity

- Each document d is represented by a vector $\vec{d}$
- The cosine similarity of two documents d_1 and d_2 is:

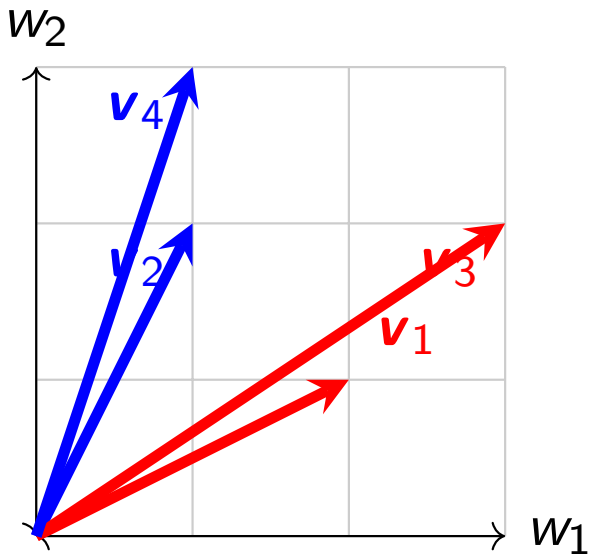
$$\text{sim}(d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{|\vec{d_1}||\vec{d_2}|}$$

- $\vec{d_1} \cdot \vec{d_2}$ is the inner product between $\vec{d_1}$ and $\vec{d_2}$ (i.e., $\vec{d_1} \cdot \vec{d_2} = \sum_{i=1}^{v} \vec{d_{1i}} \vec{d_{2i}}$)
- $|\vec{d_i}|$ is the norm of $\vec{d_i}$
- $\text{sim}(d_1, d_2)$ is equal to the cosine of the angle between the two vectors
- The larger the degree of similarity is, the smaller the angle between the vectors is

Term Frequency

Example

- Let's the vocabulary set be $\{w_1, w_2\}$.
- **Doc1**: $w_1 w_2 w_1 \rightarrow \mathbf{v}_1 = [2, 1]$
- **Doc2**: $w_2 w_2 w_1 \rightarrow \mathbf{v}_2 = [1, 2]$
- **Doc3**: $w_2 w_1 w_2 w_1 w_1 \rightarrow \mathbf{v}_3 = [3, 2]$
- **Doc4**: $w_2 w_2 w_1 w_2 \rightarrow \mathbf{v}_4 = [1, 3]$



Queries as Vectors

- Queries can be considered as short documents
- The elements of a query vector can be the tf or tf-idf of the query terms
- This will be a bag-of-word representation of a query

Queries and Document Vectors

- The degree of relevance between a query and a document can be measured by their vector similarity
- If a document vector is similar enough to a query vector, the document can be considered as a relevant document
- In order to rank a set of documents for a query, we measure the similarity between each document d and the query q

$$\text{sim}(d, q) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}||\vec{q}|}$$

Queries and Document Vectors

```
CosineScore(q)
 1  float Scores[N] = 0
 2  Initialize Length[N]
 3  for each query term t
 4  do calculate w_{t,q} and fetch postings list for t
 5      for each pair(d, tf_{t,d}) in postings list
 6      do Scores[d] += wf_{t,d} × w_{t,q}
 7  Read the array Length[d]
 8  for each d
 9  do Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```

Figure: The basic algorithm for computing vector space scores. $w_{t,q}$ is the weight of the term t in the query q. $\text{wf}_{t,d}$ is the weight of the term t in document d.

Summary

- Ranked retrieval: to augment Boolean retrieval with relevance rankings
- We assign a score to each query-document pair
- Weighted zone scoring: the presence of terms in different document zones may contribute differently to the document relevance
- The weights of document zones can be learned
- Term-frequency, inverse document frequency (tf-ifd) as a technique to measure document relevance
- Documents and queries can be represented as vectors
- The relevance between documents and queries if a function of the degree of similarity between their vectors

Reading and Exercises

- Reading: Sec 6-1 to 6-3 of the Introduction to IR book.
- Exercises: 6-1 to 6-6; 6-8 to 6-11; 6-15 to 6-17

References I

- Manning, Christopher D. and Raghavan, Prabhakar and Schütze, Hinrich.
  Introduction to Information Retrieval.
  Cambridge University Press, 2008.
- Wolfram Math World Article.
  Inner Product
  https://mathworld.wolfram.com/InnerProduct.html