Search Engines

Gertjan van Noord

September 30, 2024

Plan for today

- Summary of last week
- This week: chapter 8 (evaluation)

Last week: chapter 6

- Ranked retrieval
- TFIDF
- Documents as vectors
- Normalized vectors
- Cosine Similarity

TF-IDF

- Score for a term t in document d: $\text{tf}_{t,d} \times \text{idf}_t$.
- Score for a query $q = t_1 \dots t_n$ is the sum of the tf-idf values for each search term t_i.

$$\text{score}(q, d) = \sum_{t_i} (\text{tf}_{t_i,d} \times \text{idf}_{t_i})$$

Example

TF	ape	child	food	of	panther
doc1	8	2	2	10	2
doc2	1	5	9	20	0
Query	1	0	1	1	0
IDF	5	2	2	0.001	6

score(food of ape,doc1): 8*5 + 2*2 + 10*0.001 = 44.01
score(food of ape,doc2): 1*5 + 9*2 + 20*0.001 = 23.02

We can immediately build vectors with tf-idf weights.

Vectors

TF-IDF	ape	child	food	of	panther
doc1	40	4	4	0.01	12
doc2	5	10	18	0.02	0

Remaining problem: the frequencies do not take document length into account.

Solution: **normalize vectors**

Normalize vectors

- Divide each cell in the vector by its length
- Length of a vector: square root of the sum of the squares of all the elements

	ape	child	food	of	panther
doc1	0.95	0.09	0.09	0	0.3
doc2	0.24	0.48	0.86	0	0
Query	1	0	1	1	0

Score for each document?

$$\sum_i \text{doc}_i \text{query}_i$$

Sum of products

score(food of ape,doc1): 0.95*1 + 0.09*1 + 0*1 = 1.04
score(food of ape,doc2): 0.24*1 + 0.86*1 + 0*1 = 1.10

Query can be weighted too

This also works if the terms in the query are weighted with tf-idf. Perhaps some terms occur twice in the query.

query: ape food of ape

	ape	child	food	of	panther
TF	2	0	1	1	0
IDF	5	2	2	0.001	6
TFIDF	10	0	2	0.001	0

Normalized vector for query too:

Query	0.98	0	0.20	0.0	0

Normalized vectors

	ape	child	food	of	panther
doc1	0.95	0.09	0.09	0	0.3
doc2	0.24	0.48	0.86	0	0
Query	0.98	0	0.20	0.00	0

Score for each document?

$$\sum_i \text{doc}_i \text{query}_i$$

Sum of products.

score(food of ape,doc1): 0.95*0.98 + 0.09*0.2 + 0*0.0 = 0.95
score(food of ape,doc2): 0.24*0.98 + 0.86*0.2 + 0*0.0 = 0.41

Cosine similarity

Score for each document?

$$\sum_i \text{doc}_i \text{query}_i$$

Sum of products.

If doc and query are length-normalized vectors, then this is the *cosine similarity* of the two vectors!

It is a similarity measure ranging from 0 to 1 which indicates how similar two normalized vectors are.

(This is a useful notion in many other situations)

Retrieval: find a document that is most similar to the query

Assignment this week

Build a Wikipedia-search-engine in which the user can pose queries consisting of one or more words. Your engine should then return the single **best** Wikipedia article for that query by computing the cosine similarity of the normalized tf-idf query and the normalized tf-idf values of all documents in which at least one of the search terms occur.

In order to be able to do this efficiently, you need to prepare the data in a pre-processing phase. This preparation will create a data-structure which will allow you to find efficiently:

- for a given doc-id the text of the document (including its title)
- for a given term the doc-id(s) in which that term appears, including its normalized tf-idf value

Assignment this week

After that, your program reads queries from standard input. For each query:

1. it builds the normalized tf-idf query vector

2. it then needs to go over all doc-ids in which at least one of the search terms occur

   - for each doc-id, the cosine similarity is computed

3. the title of the document with the highest similarity is printed to standard output (prefixed by the actual cosine similarity).

Assignment this week

Note: the idf value of a term that does not exist in the data at all is 0, hence the normalized tf-idf value will be 0 too. Such a query term can thus be ignored.

This week: chapter 8

- Evaluation

Evaluation

Performance of a system:

- speed
- security
- usability
- . . .
- core functionality:
  - are we happy with the documents that the system presents?
  - i.e., is our *information need* satisfied?

User Queries

- Same query terms, different information need:
  - `check my cash`
  - `cash my check`
- Different query terms, same information need:
  - `Can I open an interest-bearing account`
  - `open savings account`
  - `start account for saving`
- Gap between users' language and official terminology

Automatic testing IR systems

We need:

- Document collection
- Test suite of information needs, expressed as *queries*
- Relevance judgments: relevant or not relevant for each query / document pair

Determining relevance is not easy

- Human annotators will decide
- In large document sets, it is hard to go through all documents

Experimental test sets

- ADI
- CACM
- Cranfield
- TREC
- . . .

Basic performance measures

- Precision: relevant proportion of returned documents
- Recall: returned proportion of relevant documents

Basic performance measures

- Precision: relevant proportion of returned documents
- Recall: returned proportion of relevant documents

Examples:

- My system has produced 12 results for a particular query. Of those, 8 were judged relevant. In total, there were 10 relevant documents for this query in the document set.
- Precision: 8/12
- Recall: 8/10

Precision and Recall

- What if a system returns all documents?

Precision and Recall

- What if a system returns all documents? Recall high, Precision low.
- What if a system only returns a single best matching document?

Precision and Recall

- What if a system returns all documents? Recall high, Precision low.
- What if a system only returns a single best matching document? Recall low, Precision high.
- We need to take both numbers into account

F-measure

- Combination of recall and precision in a single number
- Harmonic mean
- Somewhat similar to "ordinary" mean, but prefers numbers that are closer to each other
- F-measure, or F-score

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Harmonic mean: hyper-realistic example!

You bike to university. 10 kilometers. In the morning, you go fast because of the wind: 30 kilometers/hour. In the afternoon, you are tired, and the wind turned into a real storm. You bike only 10 kilometers/hour.

Your average speed is **not** 20 kilometers/hour.

In the morning, it took 20 minutes for 10 kilometers. In the afternoon, it took 60 minutes for 10 kilometers. In total, it took 80 minutes for 20 kilometers. Your average speed therefore was 15 kilometers/hour.

Harmonic mean:

$$2 \times \frac{30 \times 10}{30 + 10} = 15$$

F-measure

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

- Harmonic mean of recall and precision
- If recall is higher, F-measure is higher
- If precision is higher, F-measure is higher
- F-measure is maximal when both precision and recall are high
- F-measure ranges from 0 to 1
- F-measure always between precision and recall

Now it gets a bit more complicated

Evaluation of *ranked* retrieval

- Recall and precision at rank
- List **relevant** document ids and corresponding rank

Evaluation of ranked retrieval

- Recall and precision at rank
- List **relevant** document ids and corresponding rank

Set of 80 docs, 4 relevant. Ordered answer set:
NRNNNNNRRNNNNNN....N...R.....NNNNN

relevant doc	rank	recall	precision
1	2		
2	8		
3	9		
4	40		

Evaluation of ranked retrieval

Set of 80 docs, 4 relevant. Ordered answer set:
NRNNNNNRRNNNNNN....N...R.....NNNNN

relevant doc	rank	recall	precision
1	2		
2	8		
3	9		
4	40		

Evaluation of ranked retrieval

Set of 80 docs, 4 relevant. Ordered answer set:
NRNNNNNRRNNNNNN....N...R.....NNNNN

relevant doc	rank	recall	precision
1	2	.25	.50
2	8		
3	9		
4	40		

Evaluation of ranked retrieval

Set of 80 docs, 4 relevant. Ordered answer set:
NRNNNNNRRNNNNNN....N...R.....NNNNN

relevant doc	rank	recall	precision
1	2	.25	.50
2	8	.50	.25
3	9	.75	.33
4	40	1.00	.10

Evaluation of ranked retrieval

Set of 80 docs, 4 relevant. Ordered answer set:
NRNNNNNRNNNNNNN....R...R.....NNNNN

relevant doc	rank	recall	precision
1	2	.25	.50
2	8	.50	.25
3	9	.75	.33
4	40	1.00	.10

- recall always goes up
- precision typically goes down, but not always
- interpolated precision: use higher value from below
- *The justification is that almost anyone would be prepared to look at a few more documents if it would increase the percentage of the viewed set that were relevant (that is, if the precision of the larger set is higher).*

Interpolated precision

relevant doc	rank	recall	precision	interpolated prec
1	2	.25	.50	.50
2	8	.50	.25	.33
3	9	.75	.33	.33
4	40	1.00	.10	.10

Single value summaries for ordered results

- 11 pt average precision: the average of the interpolated precision at recall levels 0, 0.1, 0.2, . . . 1

- 3 pt average precision: the average of the interpolated precision at recall levels 0.2, 0.5, 0.8

Single value summaries for ordered results

- p@n: precision at document cut-off value (n=5, 10, 20)
- r@n: recall at document cut-off value
- useful for web query, e.g., n = the number of hits on the first page
- R-precision: precision on rank R where R is the total number of relevant documents

Single value summaries for ordered results

- Average precision: average of precision on the ranks of relevant docs (non-interpolated)

relevant doc	rank	recall	precision
1	2	.25	.50
2	8	.50	.25
3	9	.75	.33
4	40	1.00	.10

Average precision $= (.50 + .25 + .33 + .10)/4 = 0.295$

- MAP: Mean of the Average Precisions for a set of queries

So far

- Evaluation by means of test sets
- Precision, Recall, F-measure
- Evaluation of ranked retrieval

Annotator Agreement

- Determining relevance of a document for a query is **very hard**
- Perhaps even for humans?
- If human annotators do not agree on this task, then this whole set-up does not make much sense
- Annotator agreement: find out if annotators agree on the task

Annotator Agreement

- Annotator agreement: find out if annotators agree on the task
- Have two annotators perform the same task, and check the agreement

Annotator Agreement

- Annotator agreement: find out if annotators agree on the task
- Have two annotators perform the same task, and check the agreement
- For the current task, almost all documents will be non-relevant for almost all queries
- Therefore, for almost all decisions the annotators will agree
- Agreement score is not enough - take into account "chance agreement"

Annotator Agreement: Kappa

- Annotator agreement: find out if annotators agree on the task
- Take into account $chance\ agreement$:
- Kappa statistic

Kappa

$$\text{kappa} = \frac{A - E}{1 - E}$$

- A: fraction of agreemeent cases
- E: fraction of chance agreement cases

Kappa

$$\text{kappa} = \frac{A - E}{1 - E}$$

- A: fraction of agreemeent cases: **count them**
- E: fraction of chance agreement cases: **estimate them**

Estimate chance agreement

Suppose judge A answers "N" in 9 out of 10 cases, and judge B answers "N" in 8 out of 10 cases.

The probability that the judges agree is: the probability that they both say "N" + the probability that they both say "R".

judge A says N: $P(A, N) = 0.9$
judge A says R: $P(A, R) = 0.1$
judge B says N: $P(B, N) = 0.8$
judge B says R: $P(B, R) = 0.2$
both judges say N: $P(A, N) \times P(B, N) = 0.72$
both judges say R: $P(A, R) \times P(B, R) = 0.02$
judges agree by chance: 0.74

NB. this is slightly different from table 8.2 in the book on page 152 (the book version is actually wrong imho - although in practice the difference is unimportant)

Of course, for the exercise you are requested to use the correct version!

Kappa

$$\text{kappa} = \frac{A - E}{1 - E}$$

- Suppose the judges agree in 0.85 of the cases
- And we calculated that the chance agreement is 0.74
- In that case, kappa: 0.42
- So what?

Kappa

$$\text{kappa} = \frac{A - E}{1 - E}$$

- Maximum: 1 perfect agreement
- 0: they agree as often as expected by chance. No meaningful agreement.
- ≤ 0: they disagree more often as expected by chance . . .

Kappa: rules of thumb for interpretation

$$\text{kappa} = \frac{A - E}{1 - E}$$

- Maximum: 1 perfect agreement
- > 0.8: good agreement
- > 0.67: fair agreement
- < 0.67: dubious

What to do if your annotators only show "dubious" agreement?

Kappa

What to do if your annotators only show "dubious" agreement?

- Clarify the task (better instructions to annotator)
- Adapt the task - apparantly the current task is not well-defined