Assignment 05

Calculate Cosine Similarity Score

- Input
  - Query
  - Posting List
- Output
  - List of 10 top ranked documents

Calculate Cosine Similarity Score

- Remember what this is about
  - A query as a vector
  - A corpus as a term-document matrix
    - Where each document is a column in the matrix

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

Calculate Cosine Similarity Score

- We are not going to calculate the similarity score of a query with every document
  - That would be inefficient.
  - Many scores are zero.
- We are not going to actually create a term-document matrix
  - The posting list has all the information that we need to calculate the similarity scores

Assignment 05

Calculate Cosine Similarity Score

- We are going to calculate the cosine similarity score, but in a clever way.
- Here are some constants you will need in your posting list:
  - The number of documents in the posting list (aka corpus).
    - Figure this out when creating the corpus
  - The document frequency of a term
    - This should be the number of items in a row of the posting list. (each term has its own row)
  - The term frequency of a term in a document.
    - Different for every term document pair.

Calculate Cosine Similarity Score

- Steps
  - Get a query from the user
  - Convert it to TF-IDF scores

$$tfidf(t,q) \quad = \quad WTF(t,q) * log\left(\frac{|corpus|}{df_{t,q}}\right)$$

```
$\mathrm{WTF}(t, q)$
    if $tf_{t,q} = 0$
        then $return(0)$
        else $return(1 + log(tf_{t,q}))$
```

Calculate Cosine Similarity Score

- “UCI Informatics Professors”
  - 3 terms {“UCI”, “Informatics”, “Professors”}
  - 3 TF-IDF scores
    - Size of the corpus comes from the posting list
    - The document frequency of “UCI” comes from the number of entries in the posting list for “UCI”
    - The term frequency is 1

$$tfidf(\text{“}UCI\text{”}, \text{“}UCI\ Informatics\ Professors\text{”}) \quad = \quad (1 + log(1)) * log\left(\frac{|corpus|}{(df_{\text{“}UCI\text{”}} + 1)}\right)$$

Assignment 05

Calculate Cosine Similarity Score

- Steps
  - Get a query from the user
  - Convert it to TF-IDF scores
  - Use your binary posting list to create accumulator scores for the documents with the query words
  - For each term in the query
    - Get the posting list for the term
    - Scores[d] += TF-IDF(term,query) * TF-IDF(term, document)

Calculate Cosine Similarity Score

- At the end of this we will have the data structure Scores
- Which for “UCI Informatics Professors” required looking up 3 posting lists
- Optionally the scores may be normalized so we have a mathematically meaningful comparison.
    - Create a new data-structure like Scores called Magnitude
    - For each term in the entire posting list
        - For each document represented in Scores
            - Magnitude[document] += TF-IDF(term, document)^2

Calculate Cosine Similarity Score

- Now we have Scores and Magnitude
- Now we calculate the highest rankings
- For each document in Scores
    - Double x = Scores[document]/sqrt(Magnitude[document])

Calculate Cosine Similarity Score

```
$\textsc{CosineScore}(q)$
$\textsc{Initialize}(Scores[d \in D])$
$\textsc{Initialize}(Magnitude[d \in D])$
for $each\ term(t \in q)$
    do $p \leftarrow \textsc{FetchPostingsList}(t)$
       $df_t \leftarrow \textsc{GetCorpusWideStats}(p)$
       $\alpha_{t,q} \leftarrow \textsc{WeightInQuery}(t, q, df_t)$
       for $each\ \{d, tf_{t,d}\} \in p$
           do $Scores[d] += \ \alpha_{t,q} \cdot \textsc{WeightInDocument}(t, q, df_t)$
for $d \in Scores$
    do $\textsc{Normalize}(Scores[d], Magnitude[d])$
return $top\ K \in Scores$
```

Evaluation in IR

Introduction to Information Retrieval
CS 221
Donald J. Patterson

Content adapted from Hinrich Schütze
http://www.informationretrieval.org

Evaluation in IR

Outline

- Intro to Evaluation
- Standard Test Collections
- Evaluation of Unranked Retrieval
- Evaluation of Ranked Retrieval
- Assessing relevance
- Broader perspectives
- Result Snippets

Intro to Evaluation

- There are many implementation decisions to be made in an IR system
  - Crawler
    - Depth-first or breadth-first?
  - Indexer
    - Use zones?
    - Which zones?
    - Use stemming?
    - Use multi-word phrases? Which ones?

Intro to Evaluation

- There are many implementation decisions to be made in an IR system
    - Query
        - Ranked Results?
        - PageRank?
        - Which formula do we use in the TF-IDF Matrix?
        - Should we use Latent Semantic Indexing?
            - How many dimensions should we reduce?

Intro to Evaluation

- There are many implementation decisions to be made in an IR system
  - Results
    - How many do we show?
    - Do we show summaries?
    - Do we group them into categories?
    - Do we personalize the rankings?
    - Do we display graphically?

Intro to Evaluation

Intro to Evaluation

- How can we evaluate whether we made good decisions or not?

Intro to Evaluation

- How can we evaluate whether we made good decisions or not?
  - Measure them

Measures for a search engine

- How fast does it index?
  - Number of documents per hour
  - Average document size
- How fast does it search
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries

Measures for a search engine

- We can measure all of these things:
  - We can quantify size and speed
  - We can make this precise
- What about user happiness?
  - What is this?
  - Speed of response/size of index are factors
  - But fast, useless answers won’t make a user happy
- Need to quantify user happiness also.

Measuring user happiness

- Issue: Who is the user we are trying to make happy?
  - It depends.

Measuring stakeholder happiness

- Issue: Who is the user we are trying to make happy?
  - Search engine:
    - The user finds what they want.
    - Measure whether or not they come back.

Measuring stakeholder happiness

- Issue: Who is the user we are trying to make happy?
    - eCommerce Site
        - User finds what they want
        - Are we interested in the happiness of the site?
        - Are we interested in the happiness of the customer?
        - Measure the $$ of sales per user
        - Measure number of transactions per user
        - Measure time to purchase
        - Measure conversion rate (lookers -> buyers)

Measuring stakeholder happiness

- Issue: Who is the user we are trying to make happy?
  - Enterprise site
    - Are the users “productive”?
    - Measure time savings when using site
    - Measure “things accomplished”
      - careful about confounding factors
    - Measure how much a user utilizes the site’s features

Evaluation in IR

Measuring stakeholder happiness

Measuring stakeholder happiness

- Can we measure happiness?

Measuring stakeholder happiness

- Can we measure happiness?
- Do we want to measure happiness?

Measuring stakeholder happiness

- Can we measure happiness?
- Do we want to measure happiness?
- What are some proxies for happiness?

Measuring stakeholder happiness

- Can we measure happiness?
- Do we want to measure happiness?
- What are some proxies for happiness?
  - Relevance of search results

Measuring stakeholder happiness

- Can we measure happiness?
- Do we want to measure happiness?
- What are some proxies for happiness?
  - Relevance of search results
    - How do we measure relevance?

Measuring Relevance Instead

- What do we need to measure relevance?
  - A document collection, a test corpus
  - A set of queries, benchmark queries
  - A set of answers, a gold standard
    - i.e., Document, d, {is, is not} relevant to query q
    - Alternatives to binary exist, but atypical
  - Cross-validation methodology
    - Parameter tuning

Information need

- Remember the user has an information need
  - not a query
- Relevance is assessed in relation to the information need, not the query
  - e.g., I am looking for information on whether drinking red wine is more effective than eating chocolate at reducing risk of heart attacks
  - Query: red wine heart attack effective chocolate risk
  - Does the document address the need, not the query

Relevance benchmarks

- TREC - National Institute of Standards and Testing (NIST) has run a large IR test bed for many years
- Reuters and other benchmark document collections
- Retrieval tasks which are specified
  - sometimes as queries
- Human experts mark, for each query and for each document
  - Relevant or Irrelevant

Evaluation in IR

Unranked retrieval

- Precision:
  - Fraction of retrieved documents that are relevant
- Recall:
  - Fraction of relevant documents that are retrieved

Unranked retrieval

- Precision:
  - Fraction of retrieved documents that are relevant
- Recall:
  - Fraction of relevant documents that are retrieved

	Relevant	*Not Relevant*
Retrieved	*TP*	*FP*
Not Retrieved	*FN*	*TN*

Evaluation in IR

Unranked retrieval

	Relevant	*Not Relevant*
Retrieved	*TP*	*FP*
Not Retrieved	*FN*	*TN*

- Precision:
  - Fraction of retrieved documents that are relevant

?

- Recall:
  - Fraction of relevant documents that are retrieved

?

Evaluation in IR

Unranked retrieval

	Relevant	*Not Relevant*
Retrieved	TP	FP
Not Retrieved	FN	TN

- Precision:
  - Fraction of retrieved documents that are relevant

? $$Precision = \frac{TP}{TP + FP}$$

- Recall:
  - Fraction of relevant documents that are retrieved

?

Evaluation in IR

Unranked retrieval

	Relevant	*Not Relevant*
Retrieved	*TP*	*FP*
Not Retrieved	*FN*	*TN*

- Precision:
  - Fraction of retrieved documents that are relevant

? $$Precision = \frac{TP}{TP + FP}$$

- Recall:
  - Fraction of relevant documents that are retrieved

? $$Recall = \frac{TP}{TP + FN}$$

Unranked retrieval - Accuracy

- The difficulty with measuring “accuracy”
  - In one sense accuracy is how many judgments you make correctly

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

- Documents A - F, Query q

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	√	
B		√
C		√
D	√	
E		√
F	√	

- If my system returns A,C,D,E to query q....
  - How many TP, TN, FP, FN do I have?

Exercise

$$Retrieved:\ A\ C\ D\ E$$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$$Retrieved: \ A\ C\ D\ E$$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved: \ A\ C\ D\ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	√	
B		√
C		√
D	√	
E		√
F	√	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved:\ A\ C\ D\ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

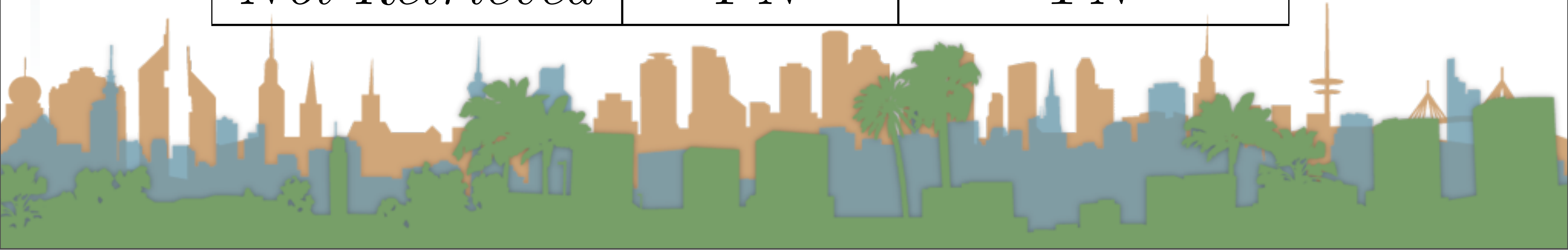
	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved:\ A\ C\ D\ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN



Exercise

$Retrieved: \ A \ C \ D \ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved: \ A\ C\ D\ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	√	
B		√
C		√
D	√	
E		√
F	√	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved: \ A\ C\ D\ E$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Exercise

$Retrieved: \ A \ C \ D \ E$

$Document$	$Relevant(q)$	$Not \ Relevant(q)$
A	$\surd$	
B		$\boxed{\surd}$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

	$Relevant$	$Not \ Relevant$
$Retrieved$	TP	FP
$Not \ Retrieved$	FN	$\boxed{TN}$

Exercise

$$Retrieved:\ A\ C\ D\ E$$

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

Evaluation in IR

Exercise

- What is our precision?
- What is our recall?
- What is our accuracy?

TP	2
FP	2
FN	1
TN	1

Exercise

- What is our precision?

$$Precision = \frac{TP}{TP + FP}$$

TP	2
FP	2
FN	1
TN	1

- What is our recall?
- What is our accuracy?

Evaluation in IR

Exercise

- What is our precision?

$$Precision = \frac{TP}{TP + FP}$$

TP	2
FP	2
FN	1
TN	1

- What is our recall?

$$Recall = \frac{TP}{TP + FN}$$

- What is our accuracy?

Exercise

- What is our precision?

$$Precision = \frac{TP}{TP + FP}$$

TP	2
FP	2
FN	1
TN	1

- What is our recall?

$$Recall = \frac{TP}{TP + FN}$$

- What is our accuracy?

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Exercise

- If my system returns A,C,D,E to query q....

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

$Precision$	$\frac{1}{2}$
$Recall$	$\frac{2}{3}$
$Accuracy$	$\frac{1}{2}$

- What do I want Precision to be?

Exercise

- If my system returns A,C,D,E to query q....

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

$Precision$	$\frac{1}{2}$
$Recall$	$\frac{2}{3}$
$Accuracy$	$\frac{1}{2}$

- What do I want Precision to be?

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

$$Precision = \frac{TP}{TP + FP}$$

Evaluation in IR

Exercise

- If my system returns A,C,D,E to query q....

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	√	
B		√
C		√
D	√	
E		√
F	√	

$Precision$	$\frac{1}{2}$
$Recall$	$\frac{2}{3}$
$Accuracy$	$\frac{1}{2}$

- What do I want Recall to be?

Evaluation in IR

Exercise

- If my system returns A,C,D,E to query q....

Document	Relevant(q)	Not Relevant(q)
A	✓	
B		✓
C		✓
D	✓	
E		✓
F	✓	

Precision	$\frac{1}{2}$
Recall	$\frac{2}{3}$
Accuracy	$\frac{1}{2}$

- What do I want Recall to be?

	Relevant	Not Relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

$$Recall = \frac{TP}{TP + FN}$$

Exercise

- If my system returns A,C,D,E to query q....

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

$Precision$	$\frac{1}{2}$
$Recall$	$\frac{2}{3}$
$Accuracy$	$\frac{1}{2}$

- What do I want Accuracy to be?

Exercise

- If my system returns A,C,D,E to query q....

$Document$	$Relevant(q)$	$Not\ Relevant(q)$
A	$\surd$	
B		$\surd$
C		$\surd$
D	$\surd$	
E		$\surd$
F	$\surd$	

$Precision$	$\frac{1}{2}$
$Recall$	$\frac{2}{3}$
$Accuracy$	$\frac{1}{2}$

- What do I want Accuracy to be?

	$Relevant$	$Not\ Relevant$
$Retrieved$	TP	FP
$Not\ Retrieved$	FN	TN

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

Unranked retrieval - Accuracy

Unranked retrieval - Accuracy

- Welcome to my search engine

Unranked retrieval - Accuracy

- Welcome to my search engine
  - I guarantee a 99.9999% accuracy.

Unranked retrieval - Accuracy

- Welcome to my search engine
  - I guarantee a 99.9999% accuracy.
  - Bring on the venture capital

Unranked retrieval - Accuracy

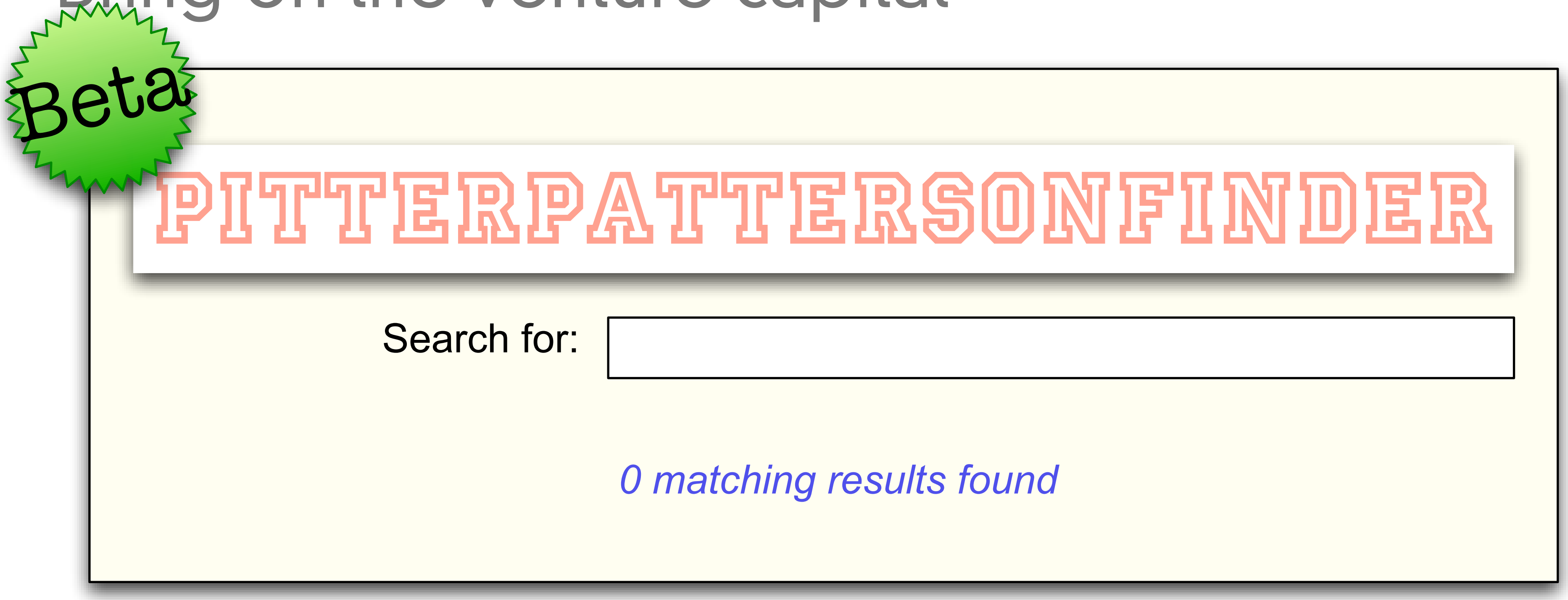
- Welcome to my search engine
  - I guarantee a 99.9999% accuracy.
  - Bring on the venture capital

Unranked retrieval - Accuracy

- Welcome to my search engine
  - I guarantee a 99.9999% accuracy.
  - Bring on the venture capital

Unranked retrieval - Accuracy

- Welcome to my search engine
  - I guarantee a 99.9999% accuracy.
  - Bring on the venture capital



Search for:

0 matching results found

Unranked retrieval - Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = \frac{0 + \uparrow}{0 + 0 + \epsilon + \uparrow}$$

Unranked retrieval - Accuracy

- Most people want to find something and can tolerate some junk

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = \frac{0 + \uparrow}{0 + 0 + \epsilon + \uparrow}$$