



Monsoon 2022

WildCard Queries

- Different Queries and query with Incorrect Spellings

Dr. Rajendra Prasath

Indian Institute of Information Technology Sri City, Chittoor



26th September 2022 (rajendra.2power3.com)

> Topics to be covered

➤ Recap:

- Inverted Index Construction
- Boolean Retrieval
- Phrase Queries
- Proximity Search

➤ Query Processing

- Exact Match vs Relevance
- Wildcard Queries
- Permuterm Index
- Bi-gram Indexes
- Spelling Variations
 - Specific tasks in Spelling Correction

➤ More Topics to come up ... Stay Tuned ...!!

Recap: Information Retrieval

- **Information Retrieval (IR)** is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).
- During these days, we frequently think first of web search, but there are many other cases:
 - E-mail search
 - Searching your laptop
 - Corporate knowledge bases
 - Legal information retrieval
 - Images / Patent / Transportation Related Searches
 - and so on . . .

Recap: Boolean Incidence Matrix

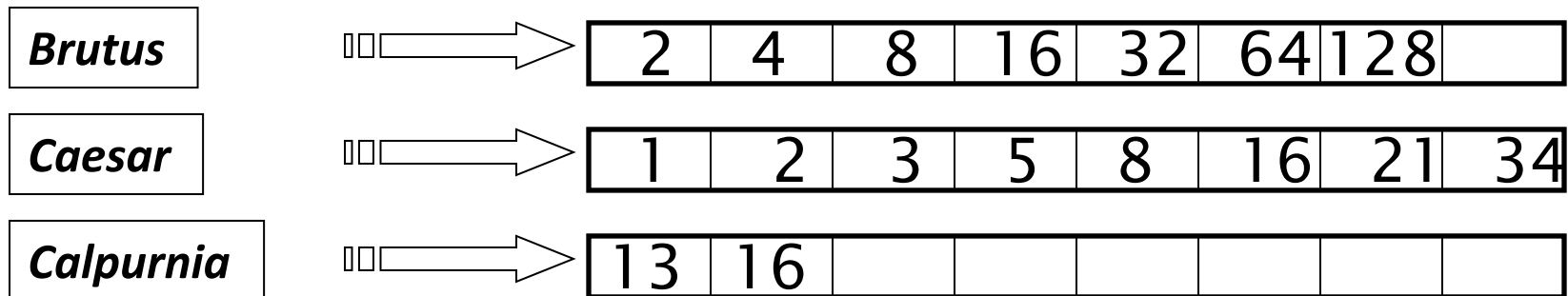
Terms	d_1	d_2	d_3	...	d_n
the	1	1	1	...	0
a	1	1	1	...	1
Darjeeling	1	1	1	...	0
is	1	1	1	...	0
of	1	1	1	...	0
in	1	0	0	...	1
and	1	1	0	...	0
Bengal	1	0	1	...	0
It	1	0	1	...	0
Its	0	1	0	...	1
state	1	0	1	...	0
West	1	0	1	...	1

Boolean Queries: Exact match

- ✧ The Boolean retrieval model is being able to ask a query that is a Boolean expression:
 - ✧ Boolean Queries are queries using AND, OR and NOT to join query terms
 - ✧ Views each document as a set of words
 - ✧ Is precise: document matches condition or not
 - ✧ Perhaps the simplest model to build an IR system on
- ✧ Primary commercial retrieval tool for 3 decades
- ✧ Many search systems you still use are Boolean:
 - ✧ Email, library catalog, Mac OS X Spotlight

Query Optimization

- ✧ What is the best order for query processing?
- ✧ Consider a query that is an AND of n terms.
- ✧ For each of the n terms, get its postings, then AND them together.



Query: *Brutus* **AND** *Calpurnia* **AND** *Caesar*

Phrase queries

- Want to be able to answer queries such as “stanford university” – as a phrase
- Thus the sentence “**I went to university at Stanford**” is not a match.
 - The concept of phrase queries has proven easily understood by users; one of the few “advanced search” ideas that works
 - Many more queries are *implicit phrase queries*
- For this, it no longer suffices to store only
 - *<term : docs>* entries

Wild Card Queries



Wild-card queries: *

- **mon***: find all docs containing any word beginning with “**mon**”.
 - Easy with binary tree (or B-tree) dictionary: retrieve all words in range: $\text{mon} \leq w < \text{moo}$
 - ***mon**: find words ending in “**mon**”: harder
 - Maintain an additional B-tree for terms backwards.
- Can retrieve all words in range: $\text{nom} \leq w < \text{non} ?$

From this, how can we enumerate all terms meeting the wild-card query pro*cent ?

Query processing

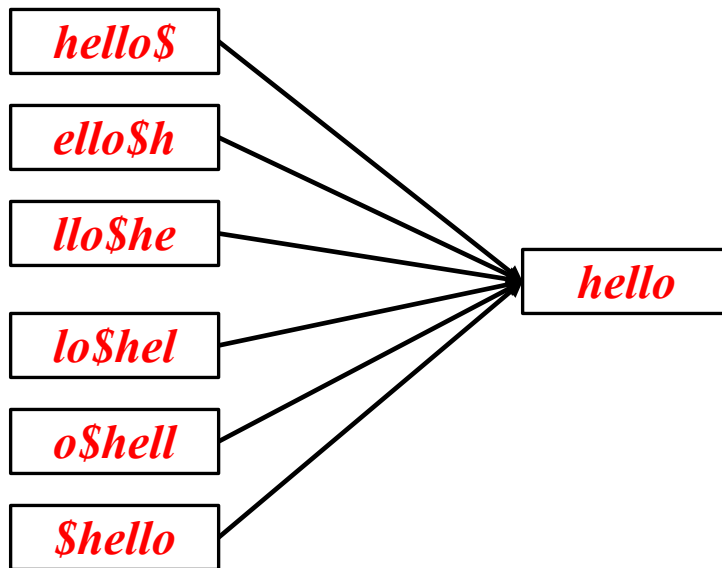
- At this point, we have an enumeration of all terms in the dictionary that match the wild-card query.
- We still have to look up the postings for each enumerated term
- E.g., consider the query:
 se*ate AND fil*er
- This may result in the execution of many Boolean AND queries.

B-trees handle *'s at the end of a query term

- How can we handle *'s in the middle of query term?
 - co*tion
- We could look up co* AND *tion in a B-tree and intersect the two term sets
 - Expensive
- The solution: transform wild-card queries so that the *'s occur at the end
- This gives rise to the PermuTerm Index.

Permuterm index

- Add a \$ to the end of each term
- Rotate the resulting term and index them in a B-tree
- For term hello, index under:
 - **hello\$, ello\$h, llo\$he, lo\$hel, o\$hell, \$hello**where \$ is a special symbol.



**Empirically, dictionary
quadruples in size**

Permuterm query processing

- (Add \$), rotate * to end, lookup in permuterm index

- Queries:

■ X	lookup on X\$	hello\$ for hello
■ X*	lookup on \$X*	\$hel* for hel*
■ *X	lookup on X\$*	llo\$* for *llo
■ *X*		
■ X*Y	lookup on X*	ell* for *ell*
■ X*Y*Z	lookup on Y\$X*	lo\$h for h*lo

treat as a search for X*Z and post-filter

For h*a*o, search for h*o by looking up o\$h*
and post-filter hello and retain halo

Bigram (k-gram) indexes

- Enumerate all k-grams (sequence of k chars) occurring in any term
- E.g., from text “April is the cruelest month” we get the 2-grams (bigrams)

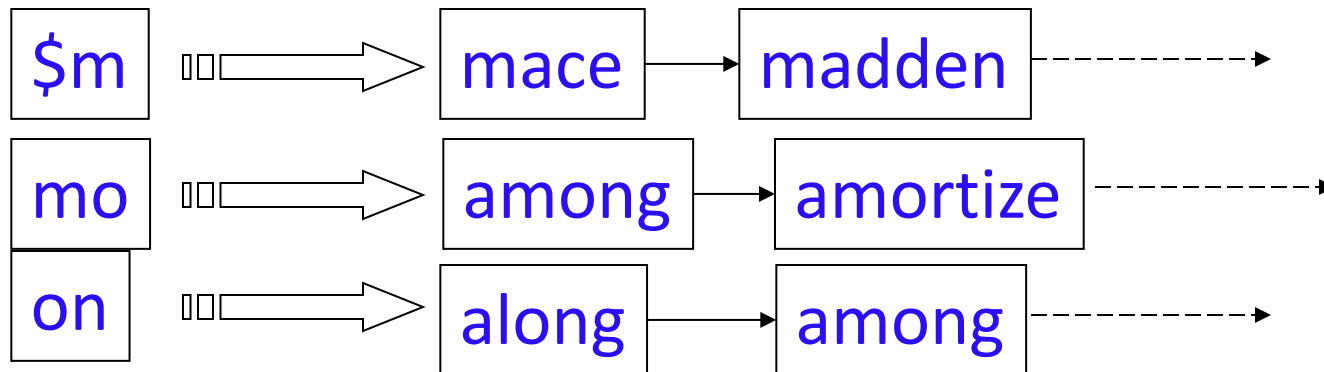
\$a,ap,pr,ri,il,l\$, \$i,is,s\$, \$t,th,he,e\$, \$c,cr,ru,
ue,el,le,es,st,t\$, \$m,mo,on,nt,h\$

\$ is a special word boundary symbol

- Maintain a second inverted index from bigrams to dictionary terms that match each bigram.

Bigram index example

- The k-gram index finds terms based on a query consisting of k-grams (here k=2).



Processing Wild-Cards

- Query **mon*** can now be run as
 - \$m **AND** mo **AND** on
- Gets terms that match **AND** version of our wildcard query
- But we would enumerate moon.
- Must post-filter these terms against query.
- Surviving enumerated terms are then looked up in the term-document inverted index
- Fast, space efficient (compared to permuterm)

Processing wild-card queries

- As before, we must execute a Boolean query for each enumerated, filtered term.
- Wild-cards can result in expensive query execution (very large disjunctions...)
 - `pyth*` **AND** `prog*`
- If you encourage “laziness” people will respond!

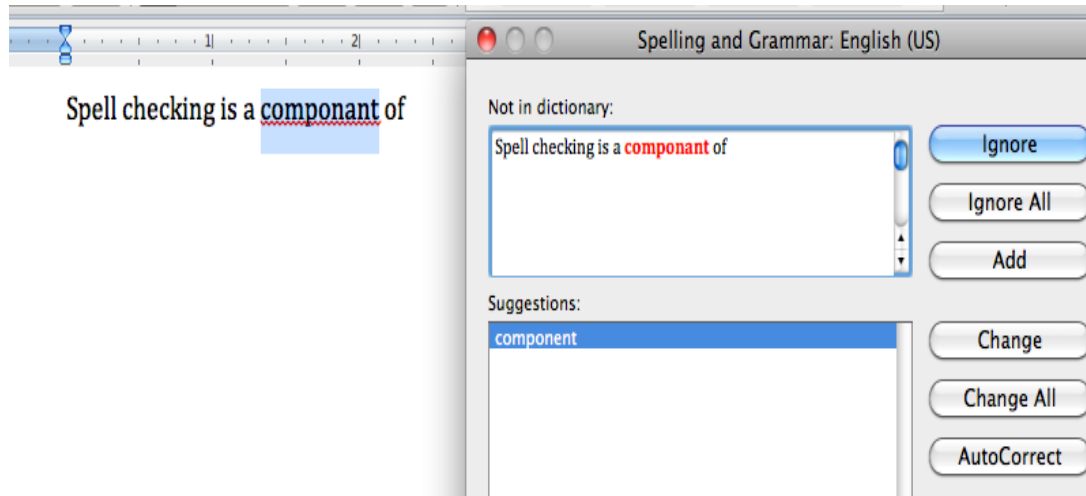
Search

Type your search terms, use '*' if you need to.
E.g., Alex* will match Alexander.

Spelling Correction

Apps For Spelling Correction

Word Processing



Phones



Web Search



Showing results for natural language processing
Search instead for natural langage processing

Rates of spelling errors

Depends on the Appln: ~1–20% error rates

26%: **Web Queries** Wang *et al.* 2003

13%: Retyping, no backspace: Whitelaw *et al.* English & German

7%: Words corrected retyping on phone-sized organizer

2%: Words uncorrected on organizer Soukoreff & MacKenzie 2003

1-2%: Retyping: Kane and Wobbrock 2007, Gruden *et al.* 1983

Spelling Tasks

- Spelling Error **Detection**
- Spelling Error **Correction**
 - Autocorrect
hte → the
- Spelling **Suggestion**
 - Correct Spelling Suggestion
 - List of Possible suggestions

Types of spelling errors

- **Non-word Errors**
 - graffe → giraffe
- **Real-word Errors**
 - Typographical errors
 - three → there
 - Cognitive Errors (homophones)
 - piece → peace,
 - too → two
 - your → you're
- Non-word correction → context insensitive
- Real-word correction almost needs to be context sensitive

Non-word spelling errors

- **Non-word spelling error detection:**
 - Any word not in a dictionary is an error
 - The larger the dictionary the better ... up to a point
 - (The Web is full of mis-spellings, so the Web isn't necessarily a great dictionary ...)
- **Non-word spelling error correction:**
 - Generate candidates: real words that are similar to error
 - Choose the one which is best:
 - Shortest weighted edit distance
 - Highest noisy channel probability

Real word & non-word spelling errors

- **For each word w , generate candidate set:**
 - Find candidate words with similar **pronunciations**
 - Find candidate words with similar **spellings**
 - Include w in candidate set
- **Choose best candidate**
 - Noisy Channel view of spell errors
 - Context-sensitive – so have to consider whether the surrounding words “make sense”
 - Flying form Heathrow to LAX → Flying from Heathrow to LAX

Character k-grams

- We just discussed **character bigrams and k-grams**:
 - st, pr, an ...
- We can also have **word bigrams and n-grams**:
 - palo alto, flying from, road repairs

Summary

In this class, we focused on:

(a) Recap: Positional Indexes

- Positional Index Size

(b) Wild card Queries

(c) Permuterm index

(d) Spell Correction

- Character k-grams

Acknowledgements

Thanks to ALL RESEARCHERS:

- Modern Information Retrieval Baeza-Yates and Ribeiro-Neto, Addison Wesley, 1999.
- **Introduction to Information Retrieval Manning, Raghavan and Schutze, Cambridge University Press, 2008.**
- Search Engines Information Retrieval in Practice W. Bruce Croft, D. Metzler, T. Strohman, Pearson, 2009.
- Information Retrieval Implementing and Evaluating Search Engines Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, MIT Press, 2010.
- Many Authors who contributed to SIGIR / WWW / KDD / ECIR / CIKM / WSDM and other top tier conferences
- **Prof. Mandar Mitra, Indian Statistical Institute, Kolkatata (<https://www.isical.ac.in/~mandar/>)**

Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- You may ask for one-to-one meeting

Best Approach

- You may leave me an email any time
(email is the best way to reach me faster)



Questions It's Your Time

