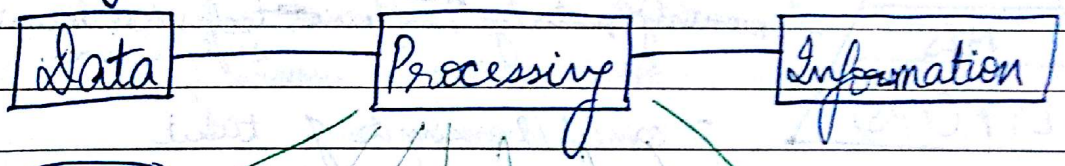
Camlin Page
Date 01/08/1998

Information Retrival



STEPS

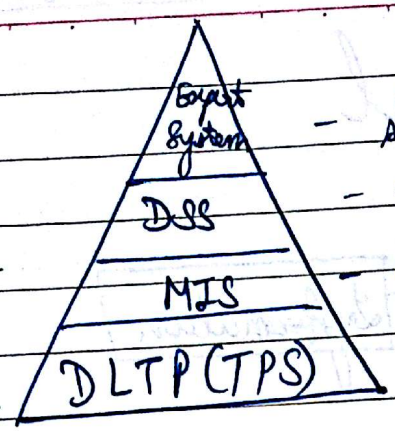
→ Collection
→ Classification - ① structured data (well defined attributes) ② semi-structured ③ unstructured.
→ Coding - item code, bar code.
→ Sorting
→ Solving
→ Validation - not entering incorrect data. - restrictions if pas.
→ Verification - password for login
→ Calculate
→ Storage - 2nd ary storage
→ Retrival

IR • query suggestion - suggestion for 2nd word on typing one word
- unstructured data
- not well defined features
- 'document' fetched from millions of sources
• maintain in-cash & tries to give u those terms.
• document ranking (Page rank)

User < Technical / statistical
Non-technical.

OLT - Online Transaction System.

Date / /



Top Level Management

Middle

- suggestion in decision making & user relies/has faith in it
- assisting - some contri to decision taking - decision support sy
- monthly/quarterly/weekly decision - technical decision board info. recieved.
  forecasting, targets, summaries, graphics
- cancel/booking etc. of tickets

- launching of new product - stategic decision.
  - bulky info. req.
- weather forecasting - semi-structured data
  ↳ prev. yr. data + this yr data

Types of IR :

① Boolean Retrival - True / False - Binary decision making

② Vector Space IR model - frequency - basic unit of VSIR
e.g. Goal post / kick
Given term
↳ suggesting based on frequency

(• Web Crawling)

Camlin Page
Date 31 08 2018

IR's goal - deliver most relavent info. - yet to be achieved
(objective) (extract) - query is not structured.
↳ semi/unstructured.
- context dep. (Search "apple")

most sy are not 100% accurate : based on stats & probab
- have to anticipate many things like context, why user writing this query,
- for same info., users can be diff. - diff. query answers (personalized)

- Uni of Illinois - speech - summarization
- ~~word~~ WordNet
- Opinion Mining - analyse what ppl giving opinion want to say
- Sentiment Analysis
  - blog v/s Twitter
  - short v/s long text analysis

What is IR?
- process of actively seeking out relavent info.

Document: web page, img, video clip, text → called doc. in IR context

IR in practice
- research - driven theoretical & experimental discipline.
  theorem, proofs, formulae ↳ empirical research
  stats
focus : context - dependent.
- Complexity - sy must be fast
- utilize space to best.

(DOM- web?)

2) librarian - indexing by cataloguing
e.g. one type → alphabetical indexing based on keyword
• this where IR originated. - initially for books.

3) cognitive sci - sentimental analysis.

4) philosophy - relavance

computational linguistics
- grammar can be converted to CL
1. syntactical analysis
2. symentical analysis - 'not' has no significance.
- find relation b/w keywords

• Wordcloud - diff colour, diff. font, diff orientation
↑ Info. visualization
- user feels comfortable to use
• retrival is not boolean. → fuzzy
All IR are fuzzy in nature.
• Most relavent words are present.

Camlin Page
Date 6 / 08 / 2018

How does it do this?

query - collection of words.
- even sequence of words (syntax & grammar) is not v. imp. (query suggestion)
- semantically imp keywords

Expanding a query → some related terms can be found internally and added in query.

1) • can generate a topic based thesaurus.
e.g. bomb entered by user., pick one/more of the

2) • generally query length ≯ 25-30
• lexical/stat. analysis/content & concept - reads 100s of docs.
i.e only good if after reading 100s of docs. on football - how good ~~post~~. ✓

3) relevance & feedback - in the beginning of Google, there was no text on search page, but now by seeing it/reading it, we know if it is relevant to us.
• Page ranking - dynamically changes.

Indexing and matching model - all of docs on thread, so how to make a mechanism to extract some documents not all.

Q. How to create index/sort as these are text docs?
A - Keywords based hashing. (rest are common words)
• extract keywords from docs.

Role of UI in IR

We use google search engine even though there are other search engines. Q Why? A Google provides excellent UI.

- not always able to define query well — prob. def.

Faceted organization

for some datasets we can define identify some features
- One of IR tasks is extracting features
medical database
- On what features, the doc. should be extracted.
- Doc. should be organized on the basis of some features
- The doc. is composed of .... then ... features
  - ↳ composition (design term)

Structuring a doc. collection

1) Supervised → identify features & classify docs. on features
2) Unsupervised (clustering) - features are not known.
   (keywords work as good features here)

Basic IR Models

1) **Boolean models**
   (In DBMS - using AND, OR)
   - either there is a result or nothing.
2) **Vector space model**
   - each doc. is expressed in terms of keywords

$d_1 = \{dt_1, dt_2, dt_3 \ldots dtm\}$
$d_2 = \{dt_5, dt_4, \ldots dtm\}$
$d_3 = \{ \_ \_ \_ \_ \_ \_ \_ \_ \_ \}$

attribute value

- **Binary**
  present - 1
  not - 0
- **Frequency**
  no. of times it is present
- **Weights**
  weight (imp) per word
  (reward one word & penalize another)

3) Cosine similarity

1 or 0 ⟶ docum & Lulong.

↳ belongs to content

- semantically related.

4) Text sami larity
- based on score, arranged/displayed decreasing order. (most to least rel.)
- set a threshold. (only above score of '0.5')

Tokenization & Indexing

Quality measures

Q- How to measure relavence?

A: Recall & Precision

eg: no. of doc = 1000

" " relavent doc = 350

" " retrieved " = 500

" " rel " " = 200

$$\text{Precision} = \frac{\text{no. of rel. ret. doc}}{\text{" ret. doc.}} = \frac{200}{500}$$

$$\text{Recall} = \frac{\text{no. of rel. ret doc.}}{\text{no. of rel. doc.}} = \frac{200}{350}$$

Goal High precision + High recall

$$\text{F-Measure (HM of P \& R)} = \frac{2PR}{P+R}$$

P ↑ R ↓ ↓ R ↑ P ↓

⇓

Need for tuning of para.

Every doc is set of tokens.

convert words to their word level

Camlin Page
Date / /

convert words to their root words

- Lemmitization
  eg good, better, best → good

- Stemming (≈ Lemmitization)
  eg play, plays, played, playing → play
  - reduce terms to their roots before indexing.

- Example from Book — Shakespeare.

Tokenization

- seperate out the words
- Not all words are taken as ~~token~~ keywords

Normalization

Match U.S.A. & USA
- Normalize terms in indexed text

Case Folding

- reduce all letters to lowercase.

Stop words

- Most common words.
  latest ~~word~~ Eng. stopwords → 720

TREC Conference

- if not performing syntactical analysis can remove them
- Obj — to ~~remove~~ reduce dimension.
- e.g. 'King of Denmark'
  - ↳ Needed

Indexing

- Inverted index ≡ word → doc containing 'word'.
- flexible & fast.

Best rep. of concept :-

1. word — coverage good, not precise
2. phrase — poor coverage, more precise
3. concept — set of rel. words | good coverage, precise

For each term T, we must store a list of all doc. that contain T. (also keep freq.)
- Organized in sorted order.

(2) • Post list - updated everytime

- Term & Doc. Freq.

Camlin Page
Date / /

Assignment - 1 → 17 Aug 2018 (Deadline)

docs = 500

↓

find unique terms in all docs

↓

~~index terms~~

freq. of each term (Hash Map)

↓

index terms

"Cryp's law"

+

Come up with ideas for indexing

(TRAIN)

Date 10/08/2018

Similarity Measures

(Metrices)

	t_1	t_2	t_3	----	t_m
d_1	$d_1 f t_1$	$d_1 f t_2$	---	----	$d_1 f t_m$
d_2	$d_2 f t_1$	$d_2 f t_2$	---	----	$d_2 f t_m$
d_3					
:					
d_n					

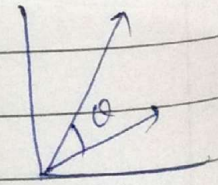
1) Euclidean Distance

K-means ⟶ Uses Euclidean distance

$$\sqrt{\sum_{i=1}^{m} (d_2 f t_i - d_1 f t_i)^2}$$

↳ Sim b/w doc1 & doc2

2)



Cosine coefficient

3) Jakkard Coefficient

indep. of freq. — if occurs !)
else : 0.

$$J(d_1, d_2) = \frac{|d_1 \cap d_2|}{|d_1| + |d_2| + -|d_1 \cap d_2|}$$

$\left(\frac{|d_1 \cap d_2|}{|d_1 \cup d_2|}\right)$

d_{m-1} d_m

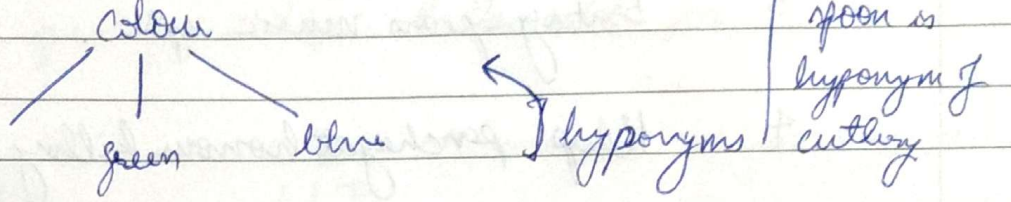
Camlin Page
Date / /

Semantic Relatedness

* Indicated degree to which words are associated via any type:-

① synonym
② antonym
③ hyponym

eg.



④ hypernym

eg. colour is hypernym of red.

⑤ meronym.

CE is meronym of eyg. instr.
part of

---

4) F-Score

$$F = 2\,\frac{P \cdot R}{P + R}$$

P = Precision
R = Recall.

5) Pearson Correlation Coeff.

-1 to 1. ← correlation

- Measure of linear dependence b/w 2 vars.

6) Con't

Assignment - 2

query 1 Annahazare Anti Land Acquisition Bill

2 Stock market mutual fund

3 Britny Spears music mp3

4 Khap panchayat honour killing

5 Sql server dbms database.

Q-1 For the above queries fetch the top 100 doc. retrieved from the Google.

Q-2 Calc. the various similarity measures and analyze the relevancy of a particular measure.

(all)

frag mat. for all except Jaccard find if there / not.

Masquerade User Detection

Masquerade – inside / outside the system
- disguise.
- attacker pretends to be auth. user of sys. in order to gain access to it / to gain greater privileges than auth. for.

13/8/18

C_1 ..... C_{635} → Commands in Linux: 635

	C_1		C_{635}
d_1	d_1f_1	d_1f_2	d_1f_{635}
d_2			
d_{50}	$d_{50}f_1$		$d_{50}f_{635}$

: find freq. of each command for each user

→ Sparse matrix

user – identification signature is the commands used everyday based on job.

Hypothesis: deviating from signature commands then that user is masquerading.

Objective: Find similarity b/w commands using some similarity matrix – ∵ Sparse + each user uses set of ~~related~~ and everyday ⟹ related (order not imp.) (based on use)

	C_1	C_2	C_3	C_4	.....	C_{635}
C_1		C_1C_2	C_1C_3	C_1C_4		
C_2			C_2C_3	C_2C_4		
C_3				C_3C_4		
C_{635}						

Similarity b/w C_1 & C_2 based on Jaccard matrix

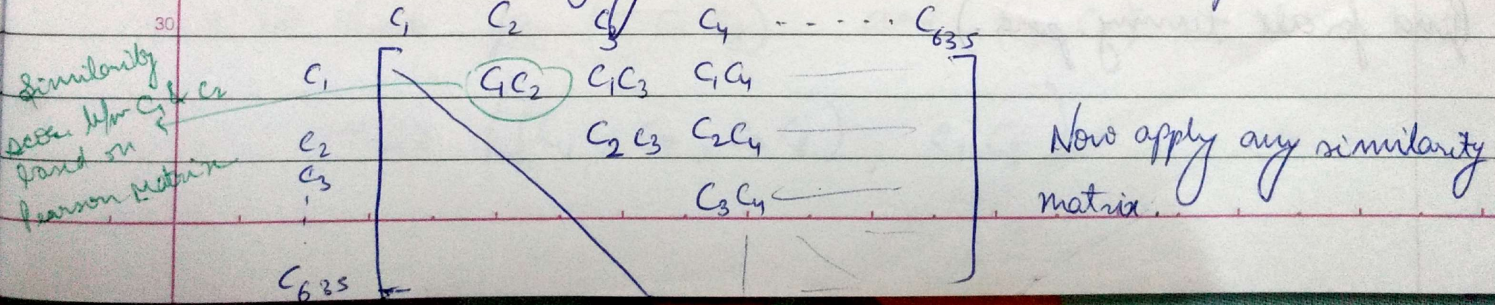
Now apply any similarity matrix.

d_{50} d_{50f_1} $d_{50f_{635}}$

user - identification signature is the commands used everyday based on job.

Hypothesis: deviating from signature commands then that user is masquerading.

Objective: Find similarity b/w commands using some similarity matrix - ∵ Sparse + each user uses set of ~~related~~ and everyday ⟹ related (order not imp.) (based on use)



Now, we want to see cmd pattern which is signature of a user.

Q- How many cmd are to be ~~added to~~ considered as signature?

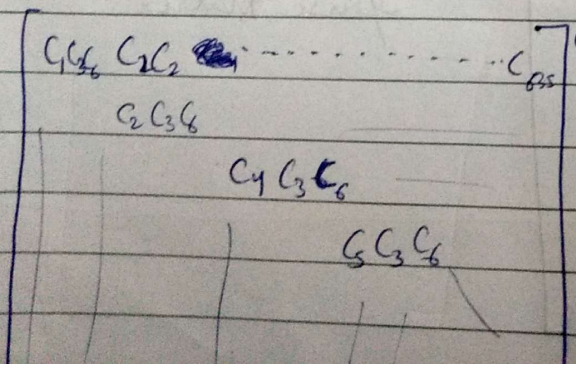
A- Experiments

Max. simi. → Related pair of cmd.

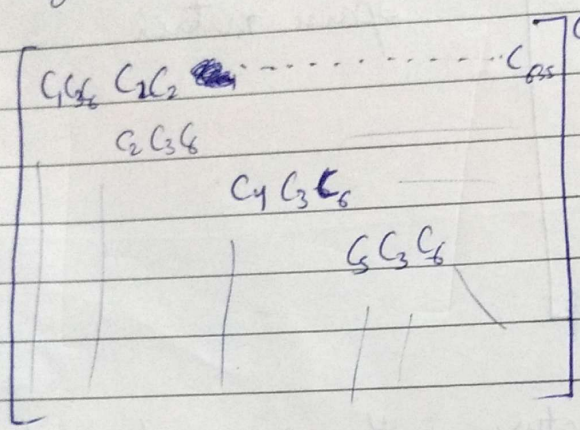
e.g. $C_3 C_6$. — max. simi.

Now treat $C_3 C_6$ as one new cmd in place of C_3, C_6 indi.

- 1st pair that is most simi is removed from mat. & new col. is added as a pair.
- Next find NEW simi. ~~toto~~ matrix. with a threshold (say 500).
- Again a max. value will come.



- Next find new sim. ~~to/vo~~ matrix.
  (say 500).
- Again a max. value will come.



↑ Hitrate : ~~Abito~~ declare masquerade user as masquerade user

↓ Missrate : " " " " " " auth. user

↓ False alarm rate : Declare ~~masquerade~~ auth. user as masquerade. ~~auth~~ user.

Goal
find for all tuning para.)

Movies on dataset ⟶ Slide 8 – Similarity Matrix

Camlin | Page | Date / /

Tuning para. – ① Threshold of no. of simi. cand.
② frequency of generated cand.
③ threshold of simi.

Use pattern over test data.

Say user : ⟶ out of 100 – 80 in pattern
– 20 out of "

↓

Set threshold (say 70 in pattern – OK)
(Based on assumption that users get bored and can try thro cand not to harm sys./access maliciously) (Human Nature has to be taken into acc)

17/08/18

C_1 C_2 C_3 ---- C_{635}

C_1
C_2
C_3
:
C_{635}

$C_{321} \cdot C_{540}$

max (max ($C_{30} \cdot C_{321}$, $C_{30} \cdot C_{540}$), ($C_{47} \cdot C_{90}$))

• To check b/w C_7 and $(C_1 C_8)$
max $(C_1 C_7 , C_7 C_8)$

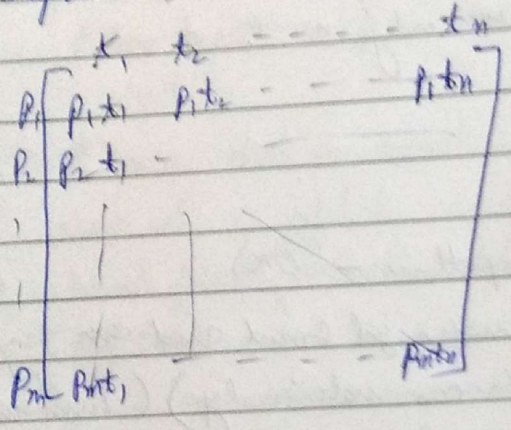
• To check for $(C_1 C_7 C_8)$ and $(C_2 C_3)$

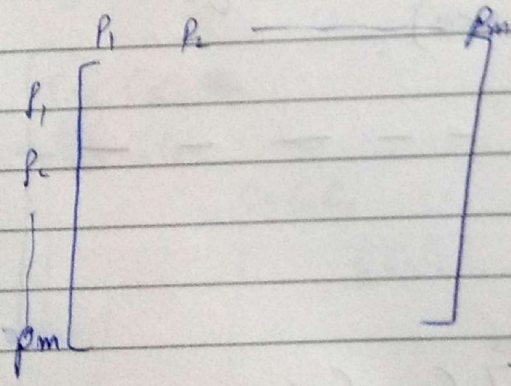
max (max $(C_1 , C_7 , C_7 C_8)$, $C_2 C_3$)

Text Summerization

- comprehencive view of doc.
- finding key-words

stopwords, lemmatization, stemming etc.





30% summary — extract top 30% ~~entire~~ paras. that are imime.

- P2 & P5 : simi
  → Index P2 & P5 together.
- We are not dropping any paras.
- If 2 paras are more simi — consider them as 1.
  If dissimilar — maybe not v. related to theme of doc.

Camlin Page
Date / /

semantic simi. on basis of para — not good idea

para can be converted to sentences.
- Better to find simi b/w sentences

Arrange sentences in descending order of simi.
→ Displays more relevant