Data Mining and Machine Learning

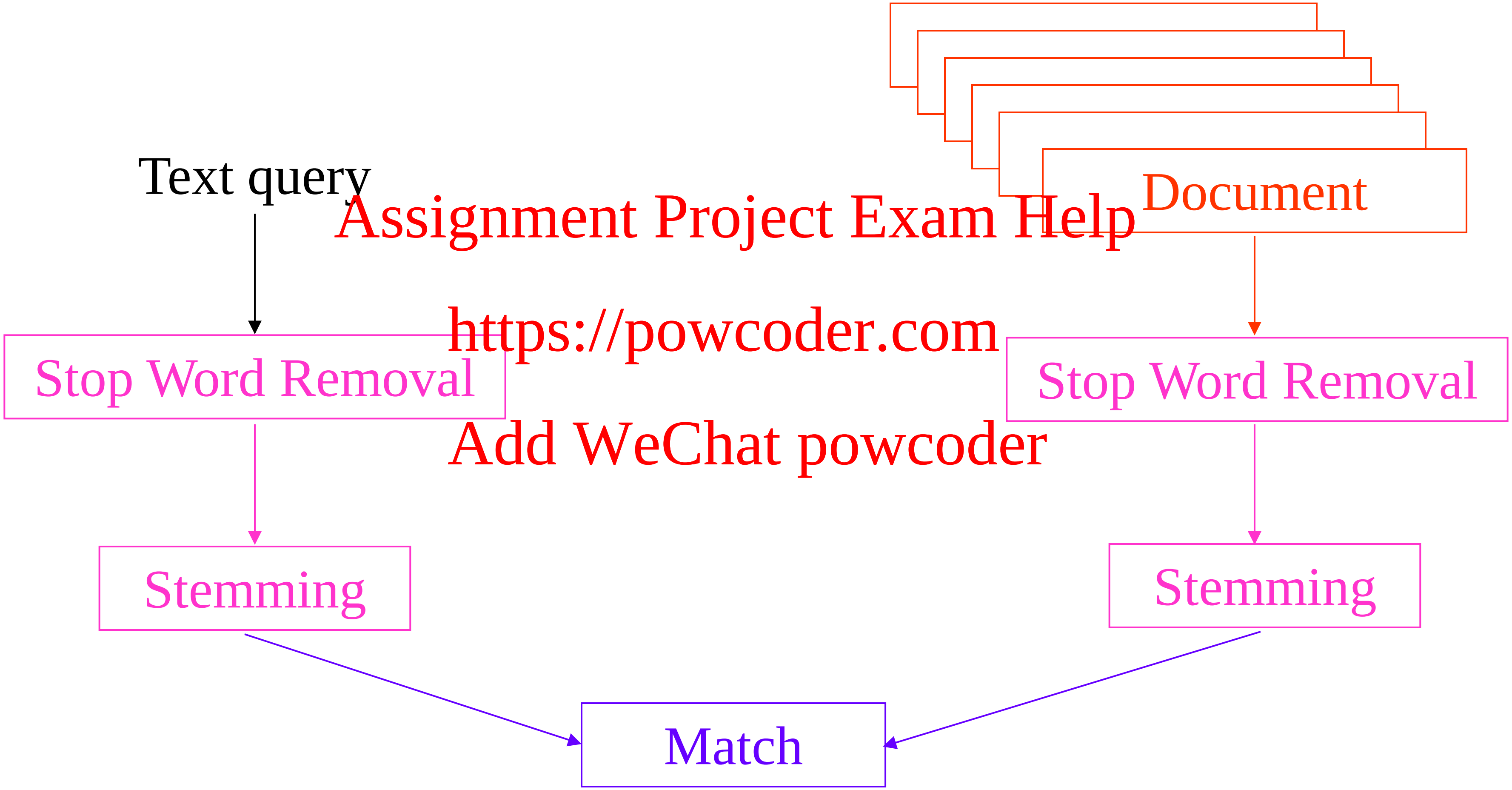
Lecture 4
TF-IDF Similarity, the Index and an Example

Peter Jančovič

Objectives

- Review IDF, TF-IDF weighting and TF-IDF similarity
- Practical considerations
- The word-document index
- Example calculation
- Assessing the retrieval

Summary of the IR process



IDF weighting

- One commonly used measure of the significance of a term for discriminating between documents is the Inverse Document Frequency (IDF)
- For a token ***t*** define:

$$IDF(t) = \log\left(\frac{ND}{ND_t}\right)$$

- *ND* is the total number of documents in the corpus
- ND_t is the number of those documents that include ***t***

TF-IDF weighting

- Let ***t*** be a term and ***d*** a document
- The weight $\boldsymbol{w_{td}}$ of term ***t*** for document ***d*** is:

$$w_{td} = f_{td} \cdot IDF(t)$$

where:

f_{td} = term frequency – the number of times ***t*** occurs in ***d***

- For w_{td} to be large:
  - f_{td} must be large, so ***t*** must occur often in ***d***
  - *IDF*(*t*) must be large, so ***t*** must only occur in relatively few documents

TF-IDF Similarity

- Define the similarity between query ***q*** and document ***d*** as:

$$Sim(q,d)=\frac{\boxed{\sum_{t\in q\cap d}} w_{td}\cdot w_{tq}}{\boxed{\|d\|}\cdot\boxed{\|q\|}}$$

Sum over all terms in both ***q*** and ***d***

‘Length’ of query ***q***

‘Length’ of document ***d***

Document length

- Suppose ***d*** is a document
- For each term ***t*** in ***d*** we can define the TF-IDF weight $\boldsymbol{w_{td}}$
- The length of document ***d*** is defined by:

$$Len(d) = \|d\| = \sqrt{\sum_{t \in d} w_{td}^2}$$

Practical Considerations

- Given a query $\boldsymbol{q}$:
  - Calculate $\|q\|$ and $\boldsymbol{w_{tq}}$ for each term $\boldsymbol{t}$ in $\boldsymbol{q}$
  - Not too much computation!

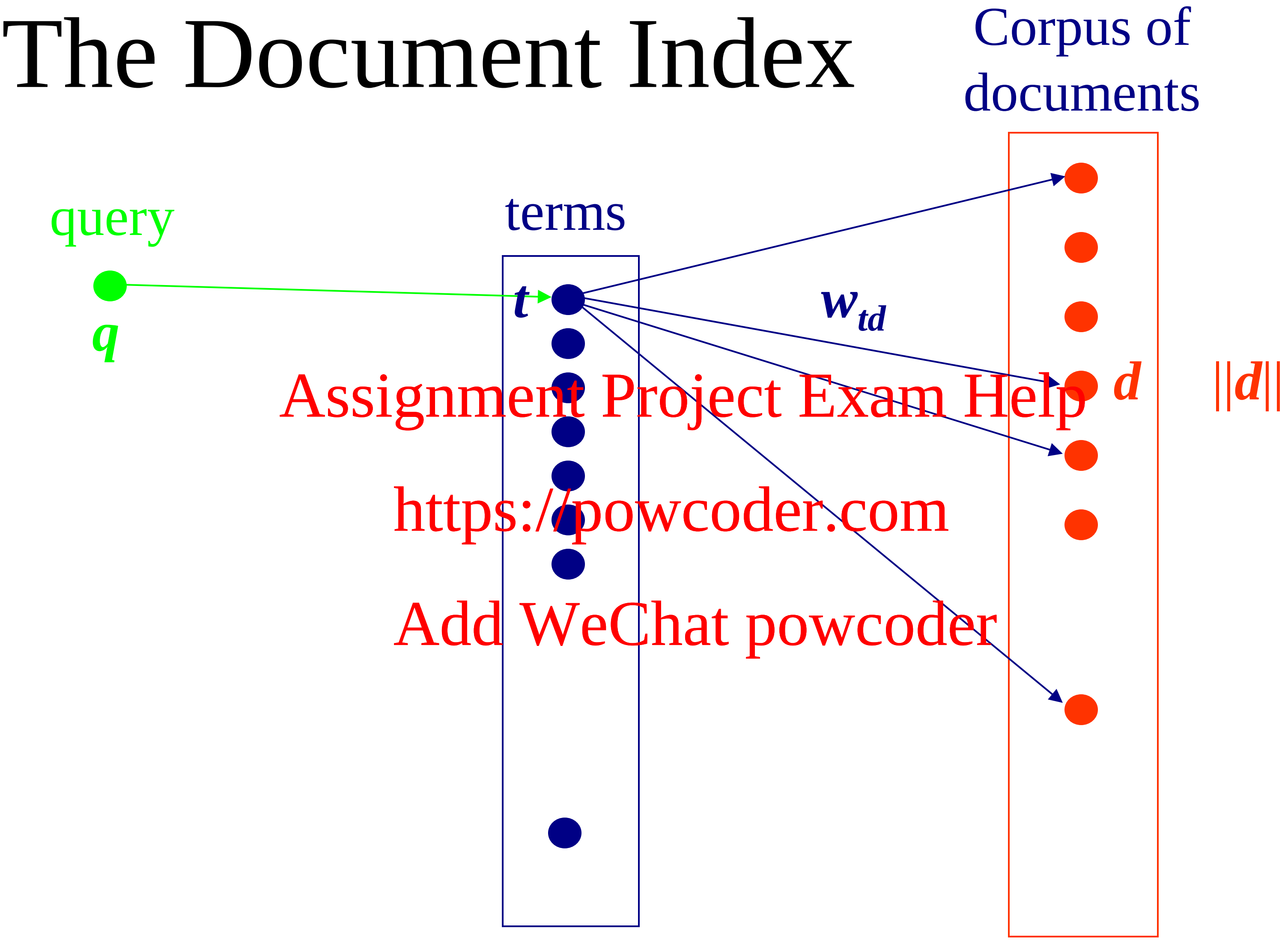
- For each document $\boldsymbol{d}$
  - $\|d\|$ can be computed in advance
  - $\boldsymbol{w_{td}}$ can be computed in advance for each term $\boldsymbol{t}$ in $\boldsymbol{d}$

- Potential number of documents is <u>huge</u>
- Potential time to compute all values $Sim(\boldsymbol{q},\boldsymbol{d})$ is huge!

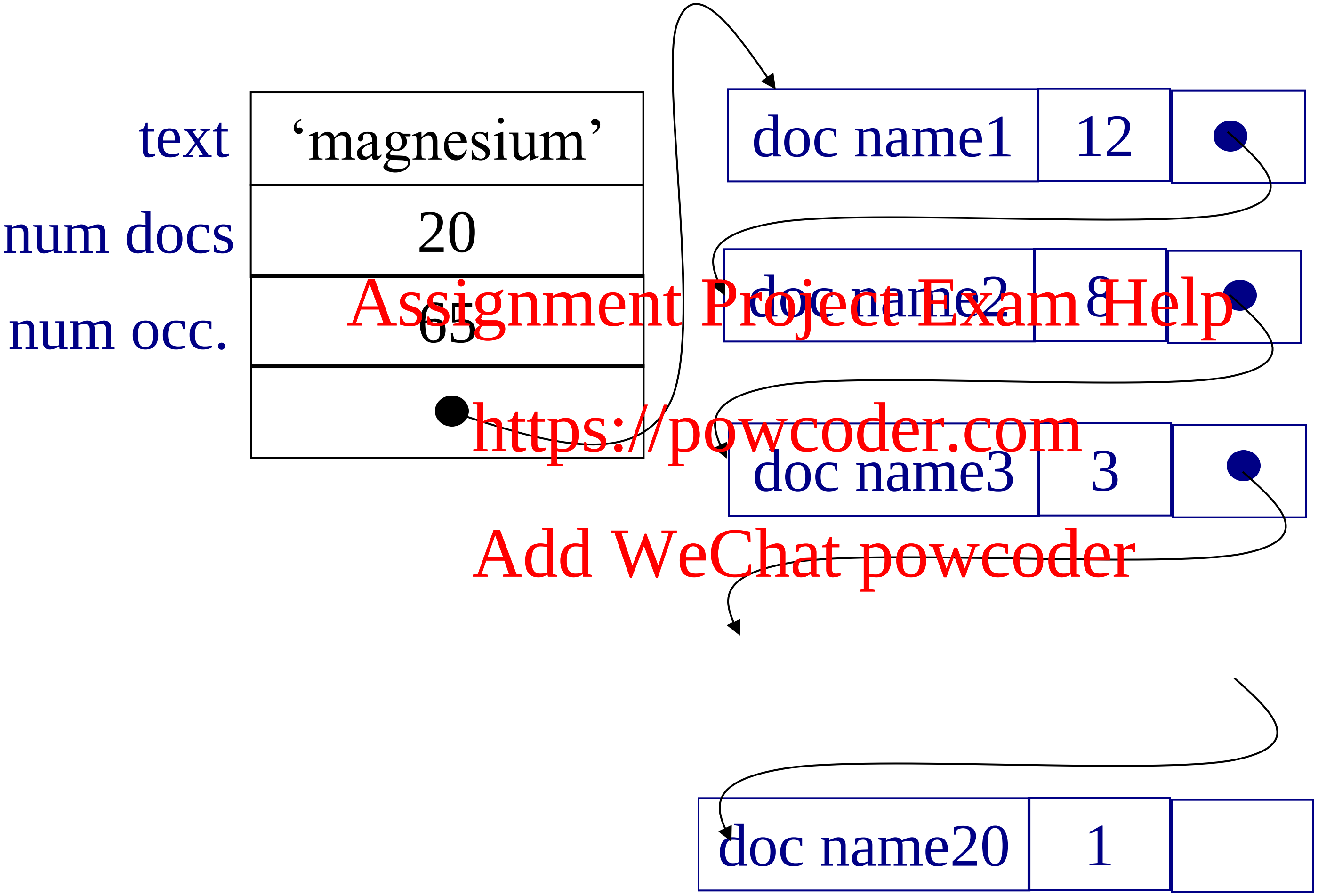
Practical Considerations Continued

- Suppose the query ***q*** contains a term ***t***
- If ***t*** didn’t already occur in the corpus it’s of no use
- Need to identify all documents ***d*** which include ***t*** (so that we can calculate *Sim*(***q,d***) for these ***d***)
- This will take too long if the number of documents is very large (as it will be in real applications)
- To speed up this computation, we compute a data structure, called the Document Index, in advance

The Document Index



The Document Index



text | ‘magnesium’

num docs | 20

num occ. | 65

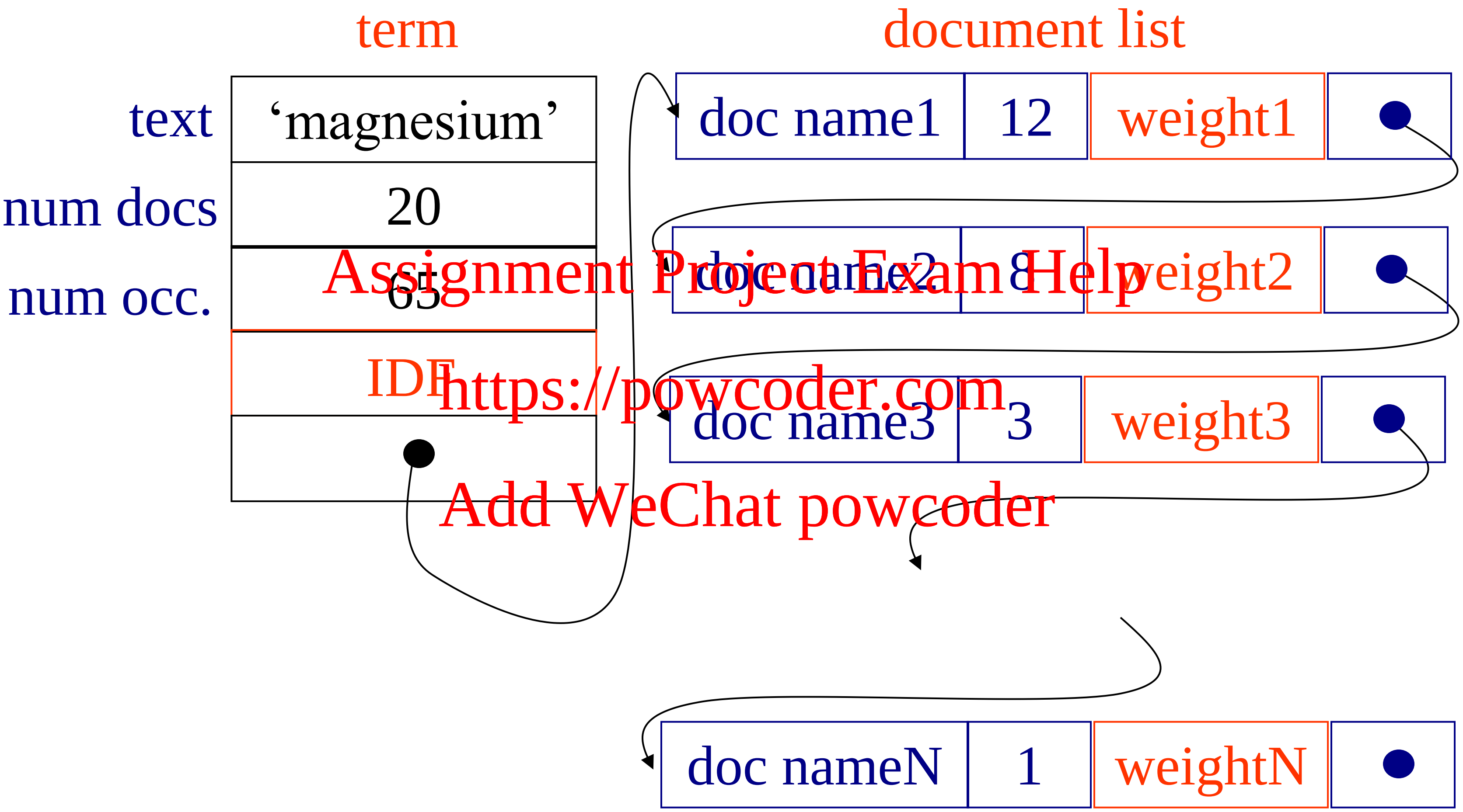
doc name1 | 12

doc name2 | 8

doc name3 | 3

doc name20 | 1

The Document Index



Practical considerations

- Order terms according to decreasing IDF
- For each term, order documents according to decreasing weight
- For each term in the query
  - Identify term in index
  - Increment similarity scores for documents in the list for this term
  - Stop when weight falls below some threshold

Building a simple text-IR system

(Preview of the IR lab)

- Example query: `communication and networks`
- Store query in `query.txt`

  - Remove stop words from query:

    ```
    stop stopList50 query.txt > query.stp
    ```

    ```
    communication networks
    ```

  - Run the stemmer on the query:

    ```
    porter-stemmer query.stp > query.stm
    ```

    ```
    comm network
    ```

- IDFs from index : `comm` – 1.422662, `network` – 1.583005

Building a simple text-IR system

(Preview of the IR lab)

- Run retrieval:
- Compile `retrieve.c`
  - `retrieve index query.stm`

```
Results (documents with similarity > 0)
==========================================

document=AbassiM.stm sim=0.176467
document=AgricoleW.stm sim=0.020104
document=AngCX.stm sim=0.051134
document=AngeloZ.stm sim=0.015214
document=AppadooD.stm sim=0.026804
...
document=YeapKS.stm sim=0.023740
document=YiuMLM.stm sim=0.265370

 Best document is YiuMLM.stm (0.265370)
```

Analysis of original document

Networking, **network** security and traffic based sampling

Project Specification:
Background. (Please include a general scene-setting overview of the project - targeted at the non-specialist)
A general view of **networking**, its flaws, and ways to combat security problems. The growing popularity of wireless **networking** means that the technology is suspect to attacks. A coverage of current technologies and further investigation into this area provides the background to this project. This will focus the project on **Network** security. The area of **network** security included **network** sampling methods. This allows for traffic monitoring along with random based sampling of files sent across a LAN. Further observations on applying this monitoring process can be applied to the internet.

Expected Outcomes. (Please include a specification for the expected outcomes of this project when undertaken by an average student. e.g. 'The aim of this project is to design and ....') The aim of this project is to design a **network** sampling tool, which monitors **network** traffic. This should monitor inbound and outbound traffic, directly observing port activity and include basic monitoring of IP protocols, such as TCP and UDP traffic. Background theory and knowledge based on **networking** is researched into, such as broadband **communication** technologies, and applications of such security tools concerning security.

Fallback and Rebuild Position. (Students sometimes have difficulty in delivering the stated outcomes. Using bullet points, please list a suitable set of minimal target objectives.) * The basic understanding of the sampling methods will allow a demonstration of the mathematical theory and practical programming examples to be identified. This will allow a simpler system using purely text files as the incoming source for sampling. * Having identified basic sampling elements of say of one character, blocks of elements can then be sample such as simple message, images and possibly sound.

Enhancement Position. (It is anticipated that many students will achieve the expected outcomes stated above. Using bullet points, please list a suitable set of achievable enhancement objectives.) * Peer 2 peer program detection - detection of peer to peer traffic activity from **network** traffic. * Detection of messaging programs such as MSN or ICQ * Identification of files being sent from sampled **network** traffic

Analysis of stopped and stemmed document

third year beng final year design project 2003/2004 project titl **network** **network** secur traffic base sampl student name mlm yiu supervisor ajg project specif background pleas includ gener scene-set overview project target non-specialist gener view **network** it flaw wai combat secur problem grow popular wireless **network** mean technolog suspect attack coverag current technolog further investig into area provid background project focu project **network** secur area **network** secur includ **network** sampl method allow traffic monitor along random base sampl file sent across lan further observ appli monitor process can appli internet expect outcom pleas includ specif expect outcom project undertaken averag student e.g aim project design aim project design **network** sampl tool monitor **network** traffic should monitor inbound outbound traffic directli observ port activ includ basic monitor ip protocol such tcp udp traffic background theori knowledg base **network** research into such broadband **commun** technolog applic such secur tool concern secur fallback rebuild posit student sometim difficulti deliv state outcom us bullet point pleas list suitabl set minim target object basic understand sampl method allow demonstr athemat theori practic program exampl identifi allow simpler system us pure text file incom sourc sampl have identifi basic sampl element sai on charact block element can then sampl such simpl messag imag possibl sound enhanc posit anticip mani student achiev expect outcom state abov us bullet point pleas list suitabl set achiev enhanc object peer 2 peer program detect detect peer peer traffic activ **network** traffic detect messag program such msn icq identif file be sent sampl **network** traffic project uniqu expect project should essenti uniqu least 80 project content thu student should abl meet project outcom reproduc materi previou project report pleas confirm uniqu project place tick adjac box

Example 2 – calculating *sim*(*q*,*d*)

- Text (*d*):
  - *The data mining course describes a set of methods for data mining and information retrieval*

- Text with stop-words removed (stopList50):

  - *data mining course describes set methods data mining information retrieval*

- Stemmed text (Porter Stemmer):
  - *data mine cours describ set method data mine inform retriev*

Example - query

- Question *(q):*
  - *Is there a module on data mining or information retrieval?*

- Question – stop words removed:

  - *module data mining text retrieval*

- Question – stemmed:
  - *modul data mine text retriev*

Example - terms

Text	*f*	*IDF*
data	2	1.5
mine	2	2.5
cours	1	1.2
describ	1	0.8
set	1	0.6
method	1	0.8
inform	1	1.1
retriev	1	2.6

Query	*f*	*IDF*
modul	1	1.6
data	1	1.5
mine	1	2.5
text	1	1.2
retriev	1	2.6

Note that these values are given – they cannot be calculated from the information that is available

Weight calculation - document

Text	*f*	*IDF*	weight = *f* * *IDF*
data	2	1.5	3.0
mine	2	2.5	5.0
cours	1	1.2	1.2
describ	1	0.8	0.8
set	1	0.6	0.6
method	1	0.8	0.8
inform	1	1.1	1.1
retriev	1	2.6	2.6

Weight calculation - query

Query	*f*	*IDF*	weight = *f* * *IDF*
modul	1	1.6	1.6
data	1	1.5	1.5
mine	1	2.5	2.5
text	1	1.2	1.2
retriev	1	2.6	2.6

Document length

- Suppose ***d*** is a document
- For each term ***t*** in ***d*** we can define the TF-IDF weight $\boldsymbol{w_{td}}$
- The length of document ***d*** is defined by:

$$Len(d) = \|d\| = \sqrt{\sum_{t \in d} w_{td}^2}$$

Length calculation - document

Text	*f*	*IDF*	*weight*	*weight*^{2}
data	2	1.5	3.0	9.0
mine	2	2.5	5.0	25.0
cours	1	1.2	1.2	1.44
describ	1	0.8	0.8	0.64
set	1	0.6	0.6	0.36
method	1	0.8	0.8	0.64
inform	1	1.1	1.1	1.21
retriev	1	2.6	2.6	6.76
			SUM	45.05
			Document Length	6.71

Length calculation - query

Query	*f*	*IDF*	*weight*	*weight*^{2}
modul	1	1.6	1.6	2.56
data	1	1.5	1.5	2.25
mine	1	2.5	2.5	6.25
text	1	1.2	1.2	1.44
retriev	1	2.6	2.6	6.76
			SUM	19.26
			Query length	4.39

TF-IDF Similarity

- Define the similarity between query ***q*** and document ***d*** as:

$$Sim(q,d) = \frac{\sum_{t \in q \cap d} w_{td} \cdot w_{tq}}{\boxed{\|d\|} \cdot \boxed{\|q\|}}$$

‘Length’ of ***q*** = ***4.39***

‘Length’ of ***d*** = ***6.71***

Example – common terms

- Terms which occur in both the document and the query
- Query
  - *modul data mine text retriev*
- Document
  - *data mine cours describ set method data mine inform retriev*
- Common terms
  - *data, mine, retrieve*

Example – common terms

Term	$w_{t,d} * w_{t,q}$
data	3.0*1.5 = 4.5
mine	5.0*2.5 = 12.5
retrieve	2.6*2.6 = 6.76
	SUM = 23.76

TF-IDF Similarity

- Define the similarity between query ***q*** and document ***d*** as:

$$Sim(q,d) = \frac{\boxed{\sum_{t \in q \cap d}} w_{td} \cdot w_{tq}}{\boxed{\|d\|} \cdot \boxed{\|q\|}}$$

Sum over all terms in both ***q*** and ***d***

= 23.76

'Length' ***q***

= 4.39

'Length' ***d***

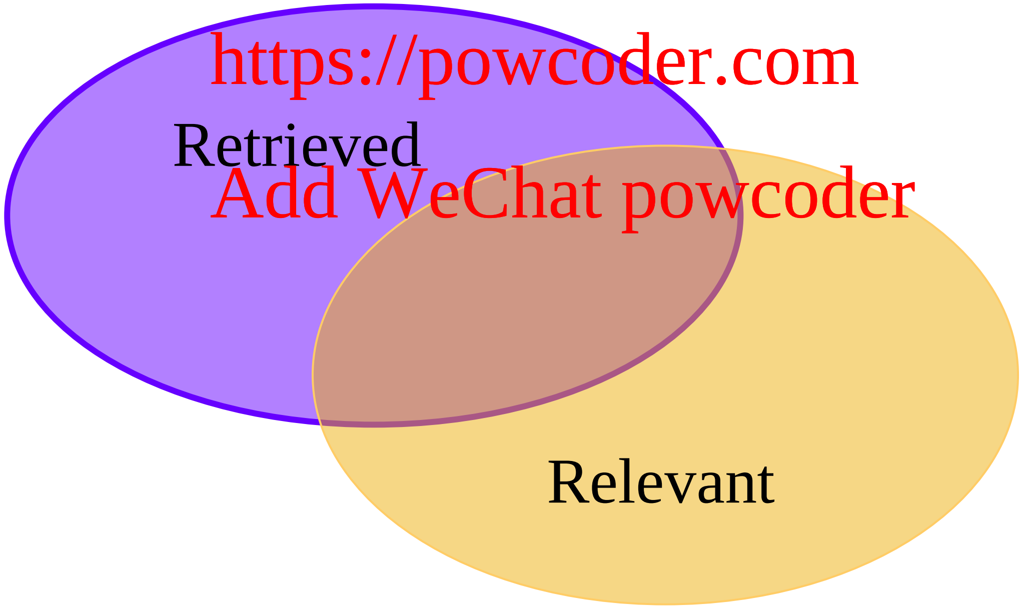
= 6.71

Example – final calculation

$$sim(q,d) = \frac{23.76}{6.71 * 4.39} = 0.81$$

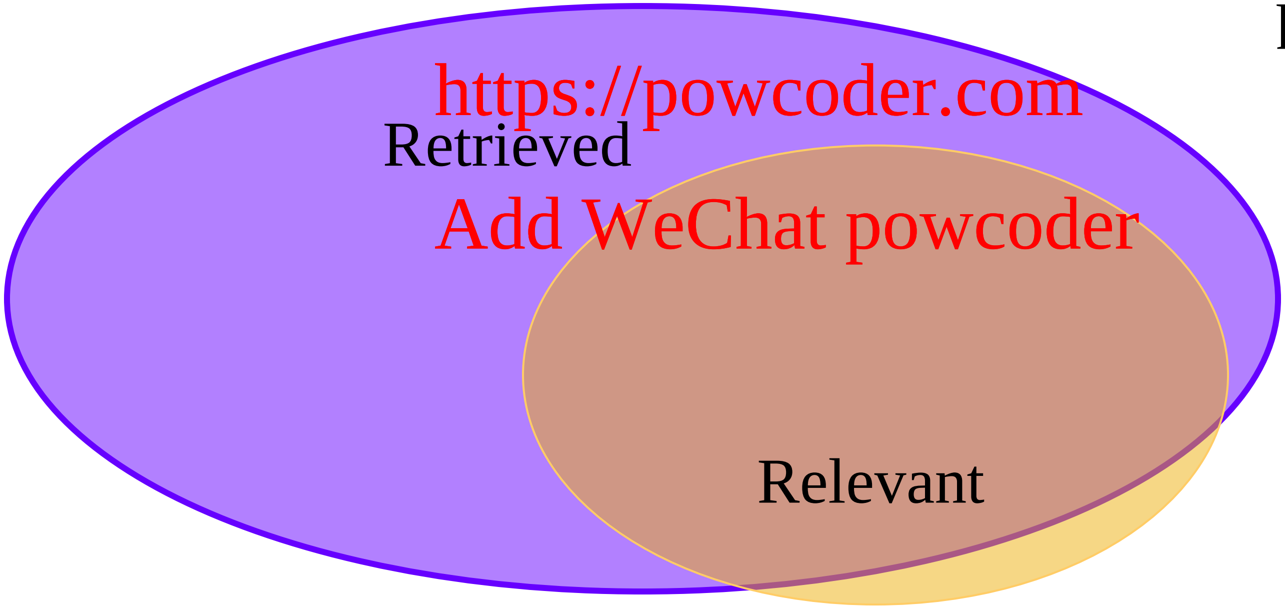
Assessing the Retrieval

- Two measures typically used:
  - Recall
  - Precision



Recall

$$\text{Recall} \equiv \frac{|\text{Retrieved} \cap \text{Relevant}|}{|\text{Relevant}|}$$

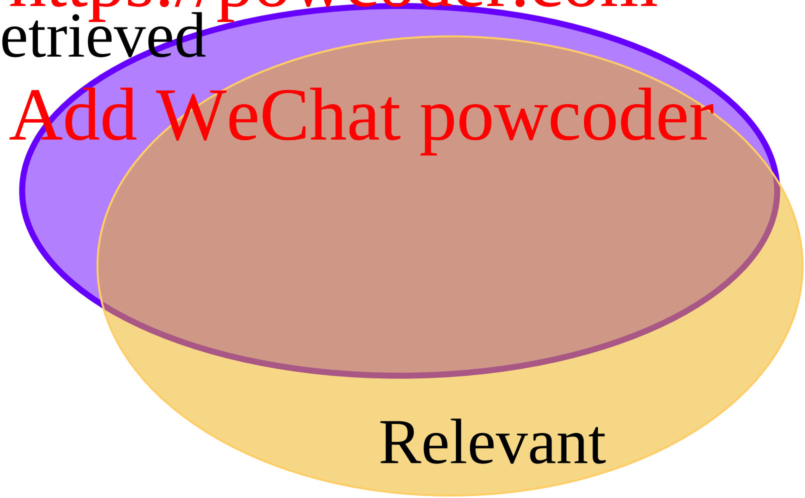


high recall retrieval

Precision

$$\text{Precision} \equiv \frac{|\text{Retrieved} \cap \text{Relevant}|}{|\text{Retrieved}|}$$

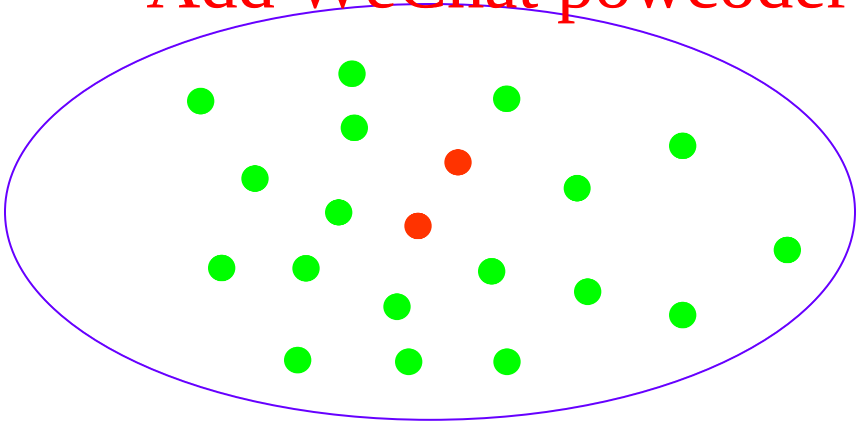
high precision retrieval



Example 1

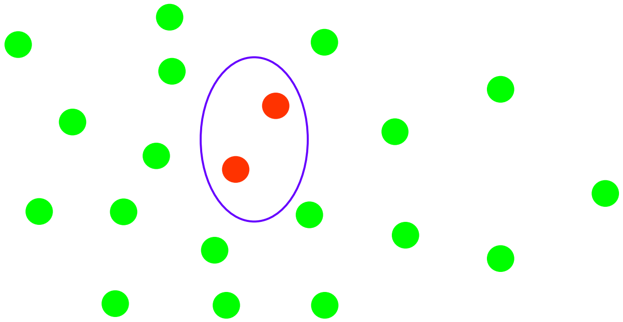
- 20 documents, 2 'about' Birmingham
- System 1 retrieves all 20 documents
  - Recall = 2/2 = 1
  - Precision = 2/20 = 0.1
  - System 1 has perfect recall, but low precision

Doc1
Doc2
Doc3
Doc4
Doc5
Doc6
Doc7
Doc8
Doc9
Doc10
Doc11
Doc12
Doc13
Doc14
Doc15
Doc16
Doc17
Doc18
Doc19
Doc20



Example 2

- System 2 retrieves Doc5 and Doc7
  - Recall = 2/2 = 1
  - Precision = 2/2 = 1
  - System 2 has perfect recall and precision

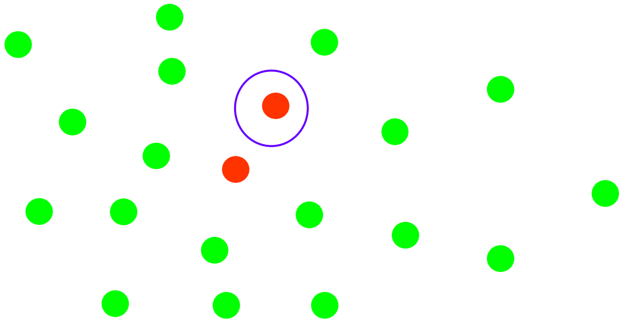


Doc1
Doc2
Doc3
Doc4
Doc5
Doc6
Doc7
Doc8
Doc9
Doc10
Doc11
Doc12
Doc13
Doc14
Doc15
Doc16
Doc17
Doc18
Doc19
Doc20

Example 3

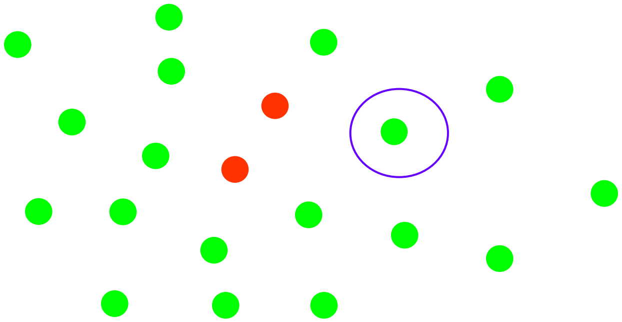
- System 3 retrieves Doc5
  - Recall = 1/2 = 0.5, Precision = 1/1 = 1
  - System 3 has poor recall but perfect precision

Doc1
Doc2
Doc3
Doc4
Doc5
Doc6
Doc7
Doc8
Doc9
Doc10
Doc11
Doc12
Doc13
Doc14
Doc15
Doc16
Doc17
Doc18
Doc19
Doc20



Example 4

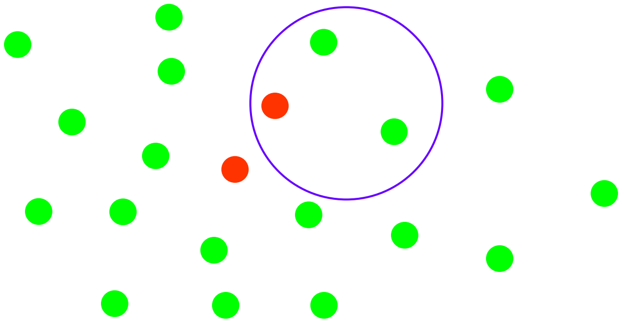
- System 4 retrieves Doc14
  - Recall = 0/2 = 0, Precision = 0/1 = 0
  - System 3 has poor recall and precision



Doc1
Doc2
Doc3
Doc4
Doc5
Doc6
Doc7
Doc8
Doc9
Doc10
Doc11
Doc12
Doc13
Doc14
Doc15
Doc16
Doc17
Doc18
Doc19
Doc20

Example 5

- System 5 retrieves Doc5, Doc8, Doc1
  - Recall = ½ = 0.5, Precision = 1/3 = 0.33



Doc1
Doc2
Doc3
Doc4
Doc5
Doc6
Doc7
Doc8
Doc9
Doc10
Doc11
Doc12
Doc13
Doc14
Doc15
Doc16
Doc17
Doc18
Doc19
Doc20

Assessing IR: Precision & Recall

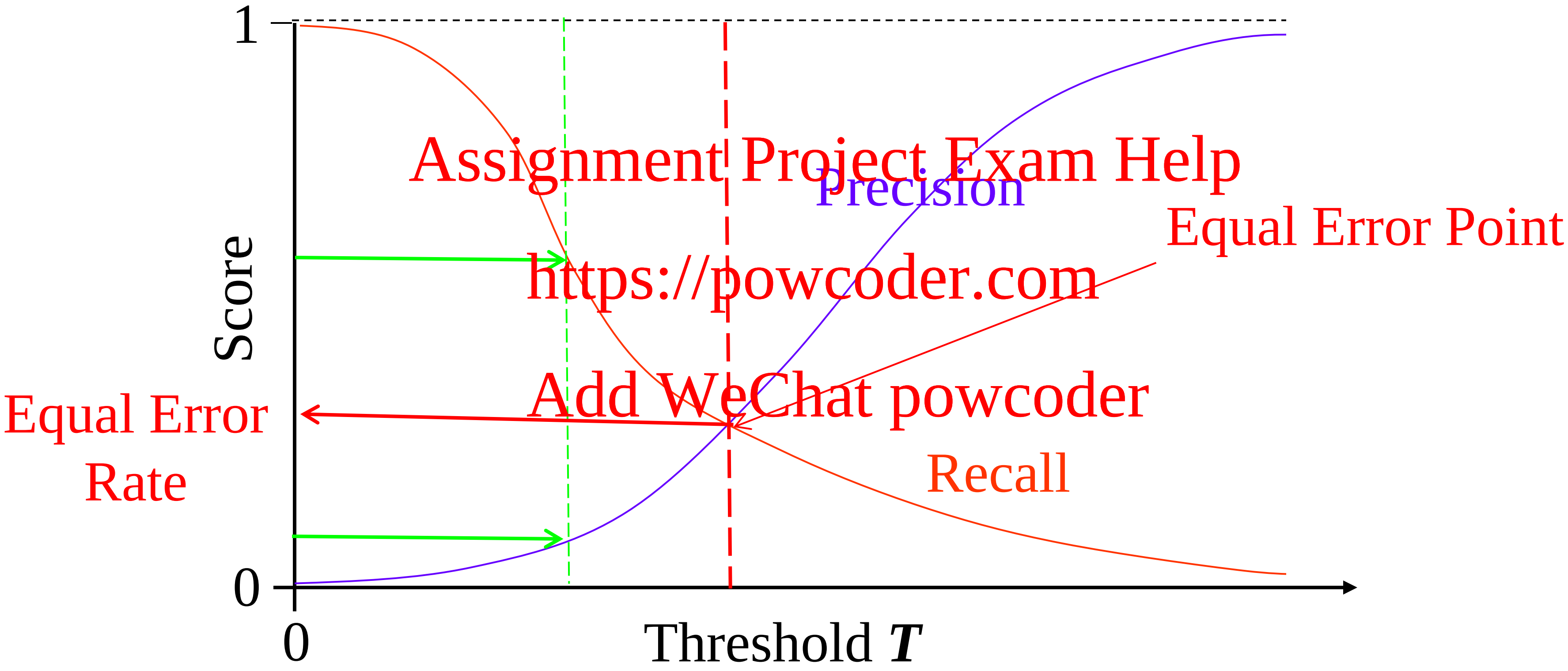
- In general, as number of documents retrieved increases:
  - Recall increases
  - Precision decreases
- In many systems:
  - Each query ***q*** and document ***d*** is assigned a similarity score *Sim*(***q***,***d***),
  - ***d*** is retrieved if *Sim*(***q***,***d***) is bigger that some threshold *T*
  - By changing *T* can trade Recall against Precision

Precision / Recall Tradeoff

- If the threshold is 0, all documents will be accepted:
  - High recall
  - Low precision
- As the threshold increases, system becomes more ‘discerning’
  - Fewer documents retrieved
  - Retrieved documents tend to be relevant - but lots missed
  - Low recall
  - High precision

ROC Curves

Receiver Operating Characteristic

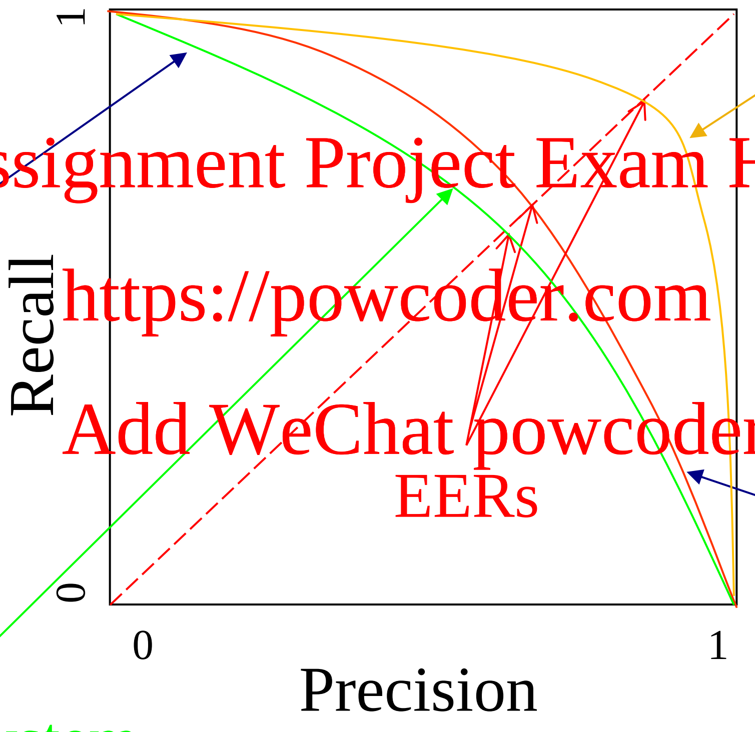


UNIVERSITY OF BIRMINGHAM

‘Precision – Recall’ graph

Also called a DET Curve

High recall, low precision. Many irrelevant documents retrieved



Better system

Worse system

High precision, low recall. ‘Selective’ system retrieves few, relevant documents

Query Processing

- Remember how we previously processed a query:
- Example:
  - "I need information on distance running"
- Stop word removal
  - information, distance, running
- Stemming
  - information, distance, run
- But what about:
  - "The London marathon will take place…"

Next lecture

- Vector representation of documents
- Cosine similarity
- Discovering “topics” in documents – Latent Semantic Analysis (LSA)