Classic Models

Kuan-Yu Chen (陳冠宇)

2018/09/21 @ TR-514, NTUST

Review

- Query & Information Need
- Relevance
- Information Retrieval & Data Retrieval

IR Modeling

- Modeling in IR is a complex process aimed at producing a ranking function
  - Ranking function is a function that assigns scores to documents with regard to a given query

- This process consists of two main tasks
  - The conception of a logical framework for representing documents and queries
    - Representation
  - The definition of a ranking function that allows quantifying the similarities among documents and queries
    - Ranking

Ranking

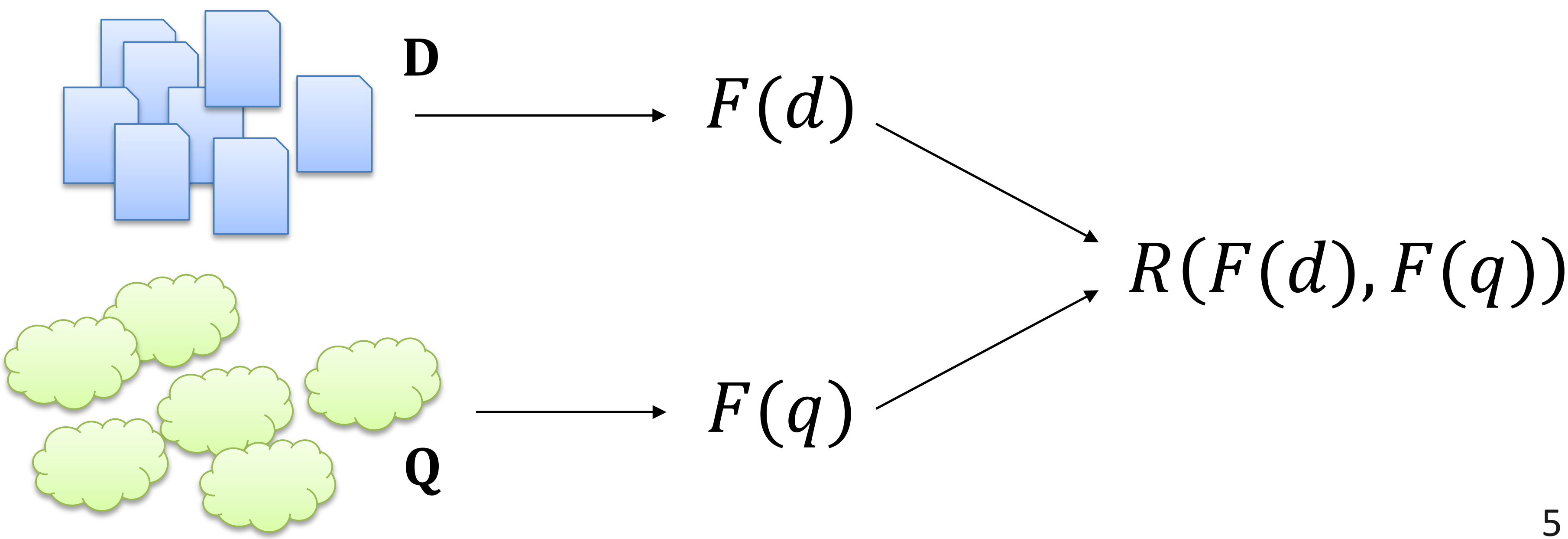
- A ranking is an ordering of the documents that reflects their relevance to a user query

- Any IR system has to deal with the problem of predicting which documents the users will find relevant

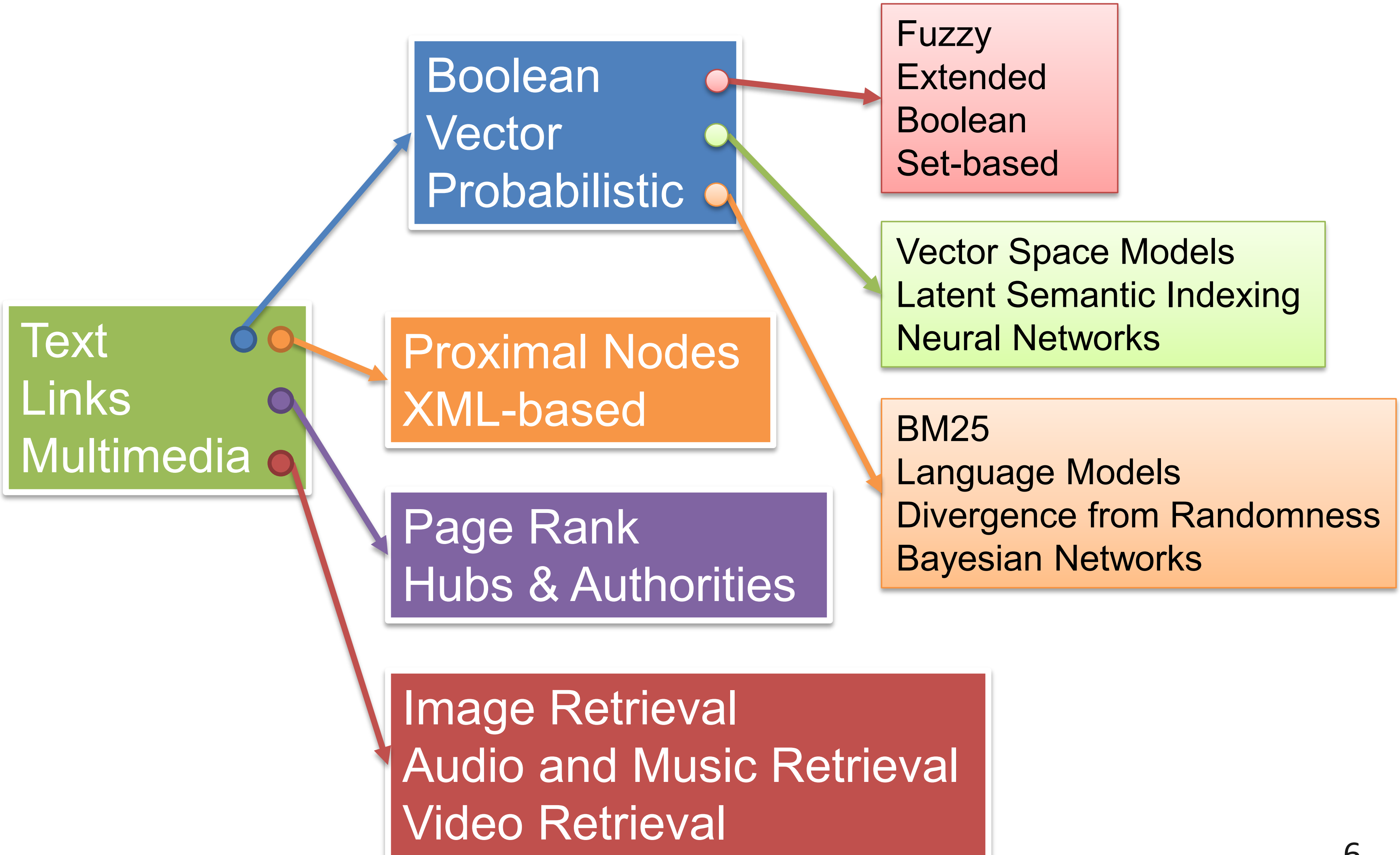
- This problem naturally embodies a degree of uncertainty, or vagueness
  - **Relevance!**

Formal Expression

- An IR model is a **quadruple** $[\mathbf{D}, \mathbf{Q}, F, R]$
  - $\mathbf{D}$ is a set of documents in the collection $\mathbf{D} = \{d_1, \dots, d_{|\mathbf{D}|}\}$
  - $\mathbf{Q}$ is a set of user queries $\mathbf{Q} = \{q_1, \dots, q_{|\mathbf{Q}|}\}$
  - F is a function that translates the queries and documents into a sort of representations
  - R is a ranking function



Taxonomy of Classic IR Models



Index Term

- Each document is represented by a set of representative keywords or index terms
  - An index term is **a word** or **group of consecutive words** in a document
- A pre-selected set of index terms can be used to summarize the document contents
  - Lexicon
- However, it might be interesting to assume that **all words are index terms** (full text representation)

Boolean Model

Boolean Model – 1

- Boolean model is a simple model, which based on **set theory** (集合論) and **Boolean algebra** (邏輯代數)

- Documents are represented by a term-document incidence matrix
  - Terms are units

- Queries specified as Boolean expressions
  - quite intuitive and precise semantics
  - neat formalism

Boolean Model – 2

- For documents
  - d_1 = The way to avoid linearly scanning is to index the documents in advance
  - d_2 = The model views each document as just a set of words
  - d_3 = We will discuss and model these size assumption

Vocabulary / Lexicon

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 3

- For term-document matrix
  - Each row associates with a term, which shows the documents it appears in
  - Each column associates with a document, which reveals the terms that occur in it

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 4

- Let's query "way"

$$way = [1\ 0\ 0]$$

$$\therefore answer = d_1$$

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 5

- Let's query non-"way"

$$\neg way = \neg[1\ 0\ 0] = [0\ 1\ 1]$$

$$\therefore answer = d_2\ \&\ d_3$$

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 6

- Let's query "document" and "model"

$$document \wedge model = [1\ 1\ 0] \wedge [0\ 1\ 1] = [0\ 1\ 0]$$

$$\therefore answer = d_2$$

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 7

- Let's query "avoid" or "view"

$$avoid \lor view = [1\ 0\ 0] \lor [0\ 1\ 0] = [1\ 1\ 0]$$

$$\therefore answer = d_1 \& d_2$$

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – 8

- Let's query "avoid" and ("view" or non-"model")

$$avoid \wedge (view \vee \neg model) = [1\ 0\ 0] \wedge ([0\ 1\ 0] \vee \neg[0\ 1\ 1])$$

$$[1\ 0\ 0] \wedge ([0\ 1\ 0] \vee [1\ 0\ 0])$$

$$[1\ 0\ 0] \wedge [1\ 1\ 0]$$

$$[1\ 0\ 0]$$

$$\therefore answer = d_1$$

	d_1	d_2	d_3
⋮			
way	1	0	0
document	1	1	0
model	0	1	1
avoid	1	0	0
view	0	1	0
discuss	0	0	1
advance	1	0	0
⋮			

Boolean Model – Drawbacks

- Retrieval based on binary decision criteria with **no notion of partial matching**
  - Data retrieval?

- **No ranking** of the documents is provided (absence of a grading scale)

- Information need has to be translated into a Boolean expression, which most users find awkward
  - The Boolean queries formulated by the users are most often too simplistic

- The model frequently returns either too few or too many documents in response to a user query

Probabilistic Model

The Probabilistic Model

- The probabilistic model captures the IR problem using a **probabilistic framework**
  - Tries to estimate the **probability** that a document will be relevant to a user query
    - $P(R_q|d_j)$
  - Assumes that this probability depends on the query and document representations only
    - Hyper-links and other information
  - The **ideal answer set**, referred to as R_q, should maximize the probability of relevance

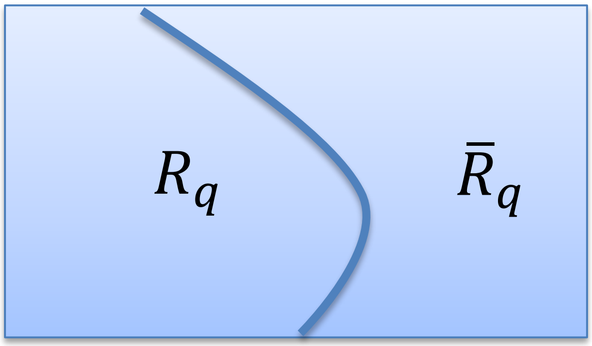
Formal Expression

- R_q be the set of relevant documents to a given query q
- $\bar{R}_q$ be the set of non-relevant documents to query q
- $P(R_q|d_j)$ be the probability that d_j is relevant to the query q
- $P(\bar{R}_q|d_j)$ be the probability that d_j is non-relevant to q

- The relevance degree can be defined as

$$sim(d_j, q) = \frac{P(R_q|d_j)}{P(\bar{R}_q|d_j)}$$

Document Collection



Derivation

- By using Bayes' rule

$$sim(d_j, q) = \frac{P(R_q|d_j)}{P(\bar{R}_q|d_j)} = \frac{\frac{P(R_q, d_j)}{P(d_j)}}{\frac{P(\bar{R}_q, d_j)}{P(d_j)}} = \frac{P(R_q, d_j)}{P(\bar{R}_q, d_j)}$$

$$= \frac{\frac{P(R_q, d_j)}{P(R_q)} P(R_q)}{\frac{P(\bar{R}_q, d_j)}{P(\bar{R}_q)} P(\bar{R}_q)} = \frac{P(d_j|R_q)\boxed{P(R_q)}}{P(d_j|\bar{R}_q)\boxed{P(\bar{R}_q)}} \propto \frac{P(d_j|R_q)}{P(d_j|\bar{R}_q)}$$

Constant for the given query *q*

Probabilistic Model – 1

- The probabilistic model can be computed by

$$sim(d_j, q) = \frac{P(d_j|R_q)P(R_q)}{P(d_j|\bar{R}_q)P(\bar{R}_q)} \propto \frac{P(d_j|R_q)}{P(d_j|\bar{R}_q)}$$

  - $P(d_j|R_q)$ probability of randomly selecting the document d_j from the set R_q
  - $P(R_q)$ probability that a document randomly selected from the entire collection is relevant to query
  - $P(d_j|\bar{R}_q)$ and $P(\bar{R}_q)$ are analogous and complementary

Probabilistic Model – 2

- We make the Naive Bayes conditional independence assumption that the presence or absence of a word in a document is independent of the presence or absence of any other word

$$sim(d_j, q) \propto \frac{P(d_j|R_q)}{P(d_j|\bar{R}_q)} = \frac{\left(\prod_{w_i \in d_j} P(w_i|R_q)\right)\left(\prod_{w_i \notin d_j} P(\overline{w}_i|R_q)\right)}{\left(\prod_{w_i \in d_j} P(w_i|\bar{R}_q)\right)\left(\prod_{w_i \notin d_j} P(\overline{w}_i|\bar{R}_q)\right)}$$

  - $P(w_i|R_q)$ is the probability that the term w_i is present in a document randomly selected from R_q
  - $P(\overline{w}_i|R_q)$ is the probability that w_i is not present in a document randomly selected from the set R_q
  - probabilities with $\bar{R}_q$: analogous to the ones just described

$$P(w_i|R_q) + P(\overline{w}_i|R_q) = 1$$
$$P(w_i|\bar{R}_q) + P(\overline{w}_i|\bar{R}_q) = 1$$

Probabilistic Model – 3

- Since we assume index terms follow the Bernoulli distributions

$$P(w_i|R_q) + P(\bar{w}_i|R_q) = 1$$
$$P(w_i|\bar{R}_q) + P(\bar{w}_i|\bar{R}_q) = 1$$

- The probabilistic model can be translated to:

$$sim(d_j, q) \propto \frac{\left(\prod_{w_i \in d_j} P(w_i|R_q)\right)\left(\prod_{w_i \notin d_j} P(\bar{w}_i|R_q)\right)}{\left(\prod_{w_i \in d_j} P(w_i|\bar{R}_q)\right)\left(\prod_{w_i \notin d_j} P(\bar{w}_i|\bar{R}_q)\right)}$$
$$= \frac{\left(\prod_{w_i \in d_j} P(w_i|R_q)\right)\left(\prod_{w_i \notin d_j} \left(1 - P(w_i|R_q)\right)\right)}{\left(\prod_{w_i \in d_j} P(w_i|\bar{R}_q)\right)\left(\prod_{w_i \notin d_j} \left(1 - P(w_i|\bar{R}_q)\right)\right)}$$

Probabilistic Model – 4

- Then, we take logarithms:

$$sim(d_j, q) \propto \frac{\left(\prod_{w_i \in d_j} P(w_i|R_q)\right)\left(\prod_{w_i \notin d_j}\left(1 - P(w_i|R_q)\right)\right)}{\left(\prod_{w_i \in d_j} P(w_i|\bar{R}_q)\right)\left(\prod_{w_i \notin d_j}\left(1 - P(w_i|\bar{R}_q)\right)\right)}$$

$$= log \prod_{w_i \in d_j} P(w_i|R_q) + log \prod_{w_i \notin d_j} \left(1 - P(w_i|R_q)\right)$$

$$-log \prod_{w_i \in d_j} P(w_i|\bar{R}_q) - log \prod_{w_i \notin d_j} \left(1 - P(w_i|\bar{R}_q)\right)$$

Probabilistic Model – 5

- By using a trick

$$
\begin{aligned}
sim(d_j, q) \propto\ & log \prod_{w_i \in d_j} P(w_i|R_q) + log \prod_{w_i \notin d_j} \left(1 - P(w_i|R_q)\right) \\
& -log \prod_{w_i \in d_j} P(w_i|\bar{R}_q) - log \prod_{w_i \notin d_j} \left(1 - P(w_i|\bar{R}_q)\right) \\
=\ & log \prod_{w_i \in d_j} P(w_i|R_q) + log \prod_{w_i \notin d_j} \left(1 - P(w_i|R_q)\right) \\
& -log \prod_{w_i \in d_j} P(w_i|\bar{R}_q) - log \prod_{w_i \notin d_j} \left(1 - P(w_i|\bar{R}_q)\right) \\
& +log \prod_{w_i \in d_j} \left(1 - P(w_i|R_q)\right) - log \prod_{w_i \in d_j} \left(1 - P(w_i|R_q)\right) \\
& +log \prod_{w_i \in d_j} \left(1 - P(w_i|\bar{R}_q)\right) - log \prod_{w_i \in d_j} \left(1 - P(w_i|\bar{R}_q)\right)
\end{aligned}
$$

Probabilistic Model – 6

- Consequently, we can obtain

$$
\begin{aligned}
sim(d_j, q) \propto\; & \log \prod_{w_i \in d_j} P(w_i|R_q) + \log \prod_{w_i \notin d_j} \left(1 - P(w_i|R_q)\right) \\
& -\log \prod_{w_i \in d_j} P(w_i|\bar{R}_q) - \log \prod_{w_i \notin d_j} \left(1 - P(w_i|\bar{R}_q)\right) \\
& +\log \prod_{w_i \in d_j} \left(1 - P(w_i|R_q)\right) - \log \prod_{w_i \in d_j} \left(1 - P(w_i|R_q)\right) \\
& +\log \prod_{w_i \in d_j} \left(1 - P(w_i|\bar{R}_q)\right) - \log \prod_{w_i \in d_j} \left(1 - P(w_i|\bar{R}_q)\right)
\end{aligned}
$$

$$
\begin{aligned}
= \; & \log \prod_{w_i \in d_j} \frac{P(w_i|R_q)}{1 - P(w_i|R_q)} + \underline{\log \prod_{w_i} \left(1 - P(w_i|R_q)\right)} \\
& +\log \prod_{w_i \in d_j} \frac{1 - P(w_i|\bar{R}_q)}{P(w_i|\bar{R}_q)} - \underline{\log \prod_{w_i} \left(1 - P(w_i|\bar{R}_q)\right)}
\end{aligned}
$$

Constant for the given query q and document d_j

Probabilistic Model – 7

- So, we have

$$sim(d_j, q) \propto log \prod_{w_i \in d_j} \frac{P(w_i|R_q)}{1 - P(w_i|R_q)} + log \prod_{w_i \in d_j} \frac{1 - P(w_i|\bar{R}_q)}{P(w_i|\bar{R}_q)}$$

- Further, lets make an additional simplifying assumption that we **only consider terms that occurring in the query**
  - This is a key expression for ranking computation in the probabilistic model

$$sim(d_j, q) \propto \sum_{w_i \in d_j \& w_i \in q} log \frac{P(w_i|R_q)}{1 - P(w_i|R_q)} + log \frac{1 - P(w_i|\bar{R}_q)}{P(w_i|\bar{R}_q)}$$

Binary Independence Model

How to Estimate? – 1

$$sim(d_j, q) \propto \sum_{w_i \in d_j \& w_i \in q} log \frac{P(w_i|R_q)}{1 - P(w_i|R_q)} + log \frac{1 - P(w_i|\bar{R}_q)}{P(w_i|\bar{R}_q)}$$

- For a given query, if we have
  - N be the number of documents in the collection
  - n_i be the number of documents that contain term w_i
  - R_q be the total number of relevant documents to query q
  - r_i be the number of relevant documents that contain term w_i

	Relevant	Non-relevant	All Documents
Documents that contain w_i	r_i	$n_i - r_i$	n_i
Documents that do not contain w_i	$R_q - r_i$	$N - n_i - (R_q - r_i)$	$N - n_i$
All documents	R_q	$N - R_q$	N

How to Estimate? – 2

- The probabilities can be estimated by:

$$P(w_i|R_q) = \frac{r_i}{R_q}$$

$$P(w_i|\bar{R}_q) = \frac{n_i - r_i}{N - R_q}$$

	Relevant	Non-relevant	All Documents
Documents that contain w_i	r_i	$n_i - r_i$	n_i
Documents that do not contain w_i	$R_q - r_i$	$N - n_i - (R_q - r_i)$	$N - n_i$
All documents	R_q	$N - R_q$	N

- Then, the equation for ranking computation in the probabilistic model could be rewritten as

$$sim(d_j, q) \propto \sum_{w_i \in d_j \& w_i \in q} log \frac{P(w_i|R_q)}{1 - P(w_i|R_q)} + log \frac{1 - P(w_i|\bar{R}_q)}{P(w_i|\bar{R}_q)}$$

$$= \sum_{w_i \in d_j \& w_i \in q} log \frac{\frac{r_i}{R_q}}{1 - \frac{r_i}{R_q}} + log \frac{1 - \frac{n_i - r_i}{N - R_q}}{\frac{n_i - r_i}{N - R_q}}$$

$$= \sum_{w_i \in d_j \& w_i \in q} log \left(\frac{r_i}{R_q - r_i} \cdot \frac{N - R_q - n_i + r_i}{n_i - r_i} \right)$$

In Practice – 1

- For handling the zero problem in the denominator, we add 0.5 to each of the terms in the formula

$$sim(d_j, q) \propto \sum_{w_i \in d_j \& w_i \in q} log\left(\frac{r_i + 0.5}{R_q - r_i + 0.5} \cdot \frac{N - R_q - n_i + r_i + 0.5}{n_i - r_i + 0.5}\right)$$

Robertson-Sparck Jones Equation

	Relevant	Non-relevant	All Documents
Documents that contain w_i	r_i	$n_i - r_i$	n_i
Documents that do not contain w_i	$R_q - r_i$	$N - n_i - (R_q - r_i)$	$N - n_i$
All documents	R_q	$N - R_q$	N

In Practice – 2

- In real case, it is hard to obtain the statistics of R_q and r_i
  - Ground truth?
  - A simplest way is to assume they are zero!

$$sim(d_j, q) \propto \sum_{w_i \in d_j \& w_i \in q} log\left(\frac{r_i + 0.5}{R_q - r_i + 0.5} \cdot \frac{N - R_q - n_i + r_i + 0.5}{n_i - r_i + 0.5}\right)$$

$$\approx \sum_{w_i \in d_j \& w_i \in q} log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

	Relevant	Non-relevant	All Documents
Documents that contain w_i	r_i	$n_i - r_i$	n_i
Documents that do not contain w_i	$R_q - r_i$	$N - n_i - (R_q - r_i)$	$N - n_i$
All documents	R_q	$N - R_q$	N

Pros and Cons

- Advantages:
  - Docs ranked in decreasing order of probability of relevance

- Disadvantages:
  - need to estimate $P\left(w_i \middle| R_q\right)$
  - method does not take "frequency" into account
  - the lack of document length normalization
    - The longer the document, the larger the score?

Overlap Score Model

Term Weighting – 1

- The terms of a document are not equally useful for describing the document contents
  - There are index terms which are **vaguer**

- There are (occurrence) properties of an index term which are useful for evaluating the importance of the term in a document
  - For instance, a word which appears in all documents of a collection is completely useless for retrieval tasks
  - However, deciding on the importance of a term for summarizing the contents of a document is not a trivial issue

Term Weighting – 2

- To characterize term importance, we associate a weight $k_{i,j} > 0$ with each term w_i that occurs in the document d_j
  - If w_i that does not appear in the document d_j, then $k_{i,j} = 0$

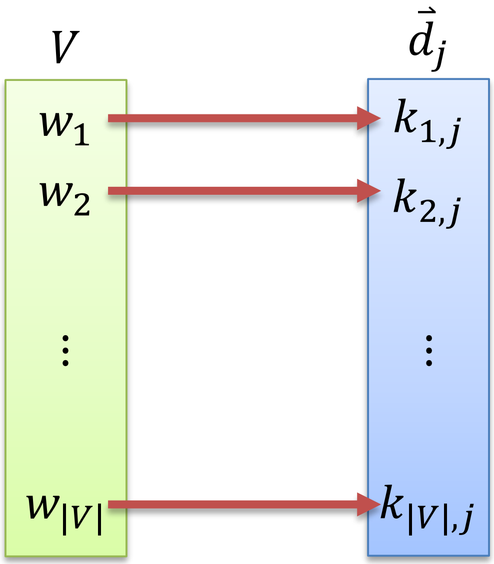
- The weight $k_{i,j}$ **quantifies the importance** of the index term w_i for describing the contents of document d_j

- These weights are useful to **compute a rank** for each document in the collection with regard to a given query

Formal Expression

- w_i be an index term and d_j be a document
- $V = \{w_1, \dots, w_{|V|}\}$ be the set of all index terms
- $k_{i,j} > 0$ be the weight associated with w_i and d_j

- We can define a $|V|$-dimensional weighted vector $\vec{d}_j$ that contains the weight of each index term $w_i \in V$ in the document d_j



Term Frequency – 1

- The value of $k_{i,j}$ is proportional to the term frequency
  - **Luhn Assumption**
  - The weights $k_{i,j}$ can be computed using the **frequencies of occurrence** of the term within the document

$$k_{i,j} = tf_{i,j}$$

- This is based on the observation that high frequency terms are important for describing documents
  - The more often a term occurs in the text of the document, the higher its weight

Term Frequency – 2

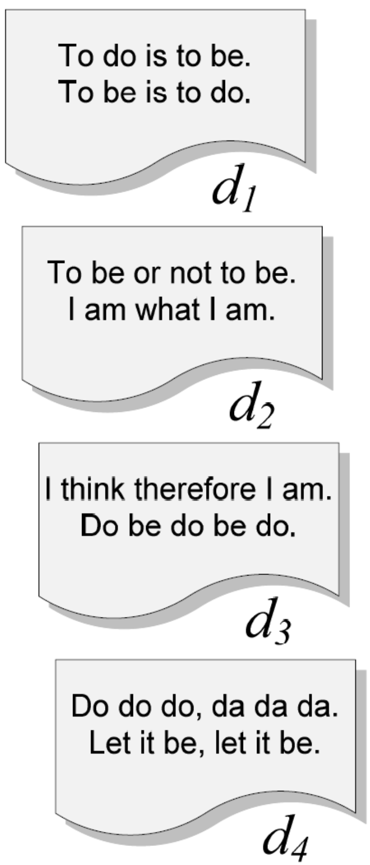
- Several variants of tf weight have been proposed

Binary	$\{0, 1\}$
Raw Frequency	$tf_{i,j}$
Log Normalization	$1 + log_2\left(tf_{i,j}\right)$
Double Normalization 0.5	$0.5 + 0.5 \dfrac{tf_{i,j}}{max_j tf_{i,j}}$
Double Normalization σ	$\sigma + (1 - \sigma) \dfrac{tf_{i,j}}{max_j tf_{i,j}}$

Term Frequency – 3

- Take "Log Normalization" for example

$$\bar{tf}_{i,j} = \begin{cases} 1 + log_2(tf_{i,j}) & \text{if } tf_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$



Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

Inverse Document Frequency – 1

- Raw term frequency as above suffers from a critical problem
  - All terms are considered equally important when it comes to assessing relevancy on a query
  - In fact certain terms have little or no discriminating power in determining relevance

- An immediate idea is to scale down the term weights by leveraging the document frequency of each term
  - Document Frequency df_i: the number of documents in the collection that contain the term w_i

Inverse Document Frequency – 2

- Denoting as usual the total number of documents in a collection by N, we define the ***inverse document frequency*** of a term w_i as follows

$$idf_i = log \frac{N}{df_i}$$

  - The idf of a rare term is high, whereas the idf of a frequent term is likely to be low
  - idf is used to reveal the **term specificity**

Inverse Document Frequency – 3

- Five distinct variants of idf weight

Unary	1
Inverse Frequency	$log\,\frac{N}{n_i}$
Inverse Frequency Smooth	$log\left(1+\frac{N}{n_i}\right)$
Inverse Frequency Max	$log\left(1+\frac{max_i(n_i)}{n_i}\right)$
Probabilistic Inverse Frequency	$log\,\frac{N-n_i}{n_i}$

TF-IDF

- We now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document

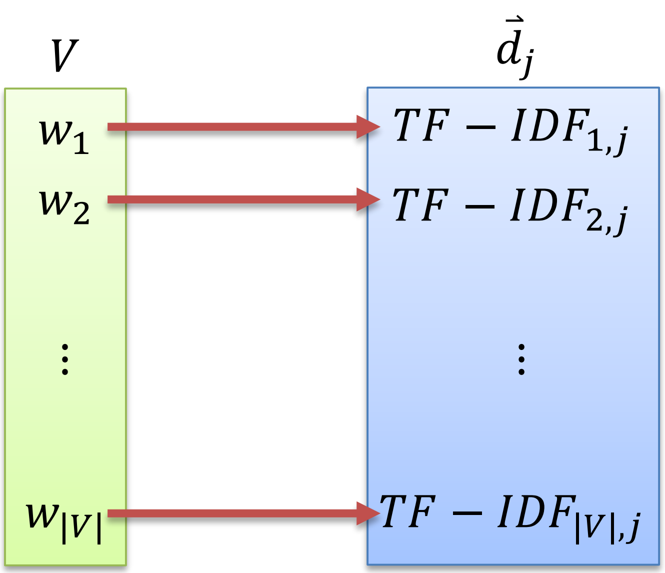
$$TF - IDF_{i,j} = tf_{i,j} \times idf_i$$

  – $TF - IDF_{i,j}$ assigns to term w_i a weight in document d_j
    - $TF - IDF_{i,j}$ will be higher when w_i occurs many times within a small number of documents
    - It will be lower when the term occurs fewer times in a document, or occurs in many documents
    - It will be the lowest when the term occurs in virtually all documents ($idf_i = 0$)

Overlap Score Model

Overlap Score Model – 1

- At this point, we may view each document as a vector with one component corresponding to each term in the dictionary
  - The weight for each component is determined by its $TF-IDF_{i,j}$
  - For dictionary terms that do not occur in the document, this weight is zero



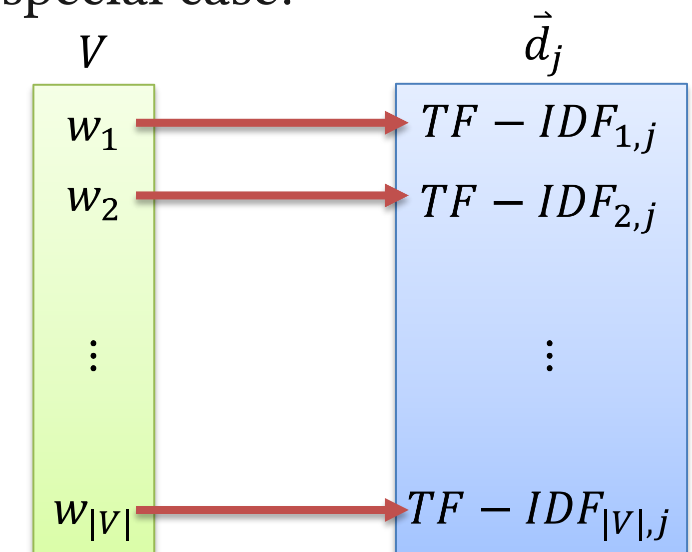
Overlap Score Model – 2

- The score of a document d_j is the sum over all query terms of the $TF-IDF_{i,j}$ weight of the query terms occurs in d_j

$$sim(q, d_j) = \sum_{w_i \in q} TF - IDF_{i,j}$$

  - Robertson-Sparck Jones Equation is a special case!

$$sim(d_j, q) \approx \sum_{w_i \in d_j \& w_i \in q} log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$



Vector Space Model

The Vector Space Model – 1

- Opposite to the overlap score model, we now present queries as vectors in the same vector space as the document collection
  - In other word, documents and queries are all vectors, and the weight for each component is determined by its $TF - IDF$

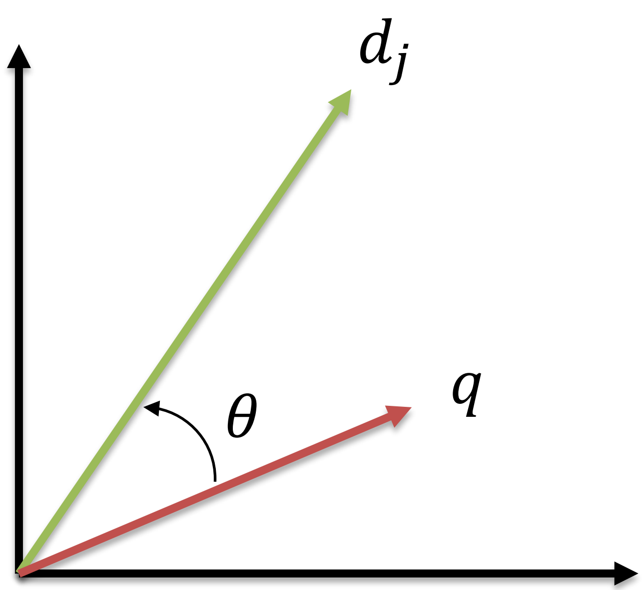
- The relevance degree between a given query and a document can be computed by referring to the cosine similarity measure

$$sim(q, d_j) = \frac{\vec{q} \cdot \vec{d_j}}{|\vec{q}||\vec{d_j}|}$$

The Vector Space Model – 2

- Similarity between a document d_j and a query q
  - If $w_{i,q} > 0$ and $w_{i,j} > 0$, we have $0 \leq sim(q, d_j) \leq 1$

$$sim(q, d_j) = cos(\theta) = \frac{\vec{q} \cdot \vec{d_j}}{|\vec{q}||\vec{d_j}|} = \frac{\sum_{w_i \in V} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{w_i \in V} w_{i,q}^2} \times \sqrt{\sum_{w_i \in V} w_{i,j}^2}}$$



Why cosine similarity measure?
Why not Euclidean distance?

The Vector Space Model – 3

- Recommended TF-IDF weighting schemes

Scheme	Document Term Weight	Query Term Weight
1	$tf_{i,j} \times log\frac{N}{n_i}$	$\left(0.5 + 0.5\frac{tf_{i,q}}{max_i(tf_{i,q})}\right) \times log\frac{N}{n_i}$
2	$1 + tf_{i,j}$	$log\left(1 + \frac{N}{n_i}\right)$
3	$(1 + tf_{i,j}) \times log\frac{N}{n_i}$	$(1 + tf_{i,q}) \times log\frac{N}{n_i}$

Pros & Cons

- Advantages
  - Term-weighting improves quality of the answer set
  - Partial matching is somewhat allowed
  - Cosine ranking formula sorts documents according to a degree of similarity to the query
  - Document length normalization is naturally built-in into the ranking

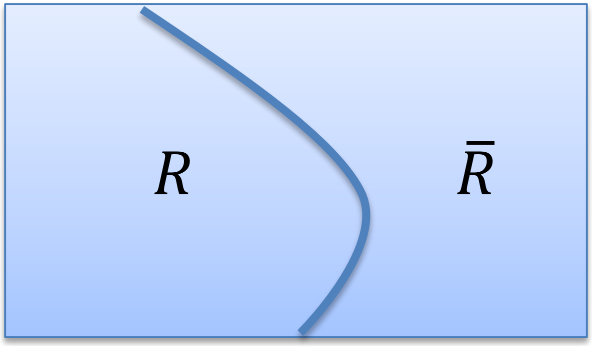
- Disadvantages
  - It assumes independence of index terms

Discussion & Comparison

TF vs. IDF

- The role of index terms

IR as a binary clustering
Relevance vs. Non-relevance



  - Which index terms (features) better describe the relevant class
    - Intra-cluster similarity (TF-factor)
    - Inter-cluster dissimilarity (IDF-factor)

Comparisons

- Boolean model does not provide for partial matches and is considered to be the weakest classic model

- There is some controversy as to whether the **probabilistic model outperforms the vector space model**
  - Croft suggested that the probabilistic model provides a better retrieval performance

- However, Salton [illegible] showed that the **vector space model outperforms probabilistic model** with general collections

Homework 1 – Retrieval Model

Homework 1 – Description.

- In this project, you will have
  - 16 Long Queries
  - 2265 Documents
  - Each words/term is represented as a number except that the number “-1” is a delimiter

20001.query

```
2280 3068 457 26763 27631 14645 732 4332 2690 2923 20646 38527 27352 -1
612 -1
1407 2986 -1
25607 432 2148 3140 -1
27414 31424 7680 2280 3068 457 24554 30066 855 1715 11629 -1
26763 18571 27631 14645 732 4332 -1
28226 23520 27646 21192 3015 24310 23142 -1
2280 3068 457 1200 -1
596 855 3015 732 1715 2033 -1
27414 2345 2471 699 24310 43624 2878 32518 13629 17554 -1
17030 18380 9561 -1
31765 19448 31359 3012 18828 -1
27646 2345 21192 11738 3015 23600 -1
596 23600 21189 341 12133 3015 1259 18571 27631 3015 14645 -1
7680 2572 8994 2345 21892 3123 2855 -1
8250 32818 699 24310 7971 30165 9608 3015 23142 -1
2280 3068 457 1200 -1
27414 3015 14177 469 1259 732 3237 24583 14177 -1
34043 9079 24310 2938 13109 34474 -1
20818 3015 13629 34445 13910 -1
24310 1408 34770 379 28672 29978 3015 19217 -1
21478 16574 10340 28940 -1
24054 3015 32474 1906 739 34445 2737 -1
9242 2938 1259 10355 3015 15128 -1
1155 34179 13023 -1
8769 17442 2717 24783 14645 19486 -1
3025 3379 39506 -1
27414 2345 1728 34975 662 27647 25995 25944 7570 23390 24310 3015 7971 -1
1027 -1
```

- Our goal is to implement a vector space model, and print out the ranking results for all of the queries

VOM19980220.0700.0166

```
19980220_0700_0800_VOA_MAN
02/20/1998 7:02:46.68
02/20/1998 7:03:41.13
40889 44022 10092 2471 9800 9561 38208 32528 3015 16920 16271 -1
10528 28193 16868 2572 11437 -1
33886 -1
38208 1200 -1
38208 13812 17231 3153 2710 2903 3015 16920 -1
40889 16972 2718 44022 43444 38478 2913 508 2718 32035 46265 -1
744 3455 9800 38208 1838 1944 2718 612 32528 3015 16920 9561 3025 13674 36506
28516 13247 40889 713 16920 26416 16868 3015 25634 -1
38208 1838 3429 1715 8775 32528 30081 709 21387 3015 16920 -1
24435 596 40889 13703 25312 18366 3015 13894 1330 18318 3082 596 30267 3015 2
3043 1259 40889 40526 35251 1200 -1
9561 38208 32528 16920 3015 16276 -1
44870 29696 38208 18412 24040 3015 30165 -1
33886 -1
38208 1200 -1
38208 3015 25958 17428 13812 596 1944 2718 612 1730 17231 3153 28654 1190 169

```

$$sim(q,d_j) = cos(\theta) = \frac{\vec{q}\cdot\vec{d_j}}{|\vec{q}||\vec{d_j}|} = \frac{\sum_{w_i\in V} w_{i,q}\times w_{i,j}}{\sqrt{\sum_{w_i\in V} w_{i,q}^2}\times\sqrt{\sum_{w_i\in V} w_{i,j}^2}}$$

Homework 1 – Description..

- The evaluation measure is MAP
  - The **hard** deadline is 10/5 11:00
  - You can get full points(10%) if you outperform the baseline, otherwise you will get 0

- You should
  - upload your answer file to kaggle
    - https://goo.gl/DXatTD
    - The maximum number of daily submissions is 20
    - **Your team name is ID_Name**

      M123456_陳冠宇
  - upload source codes and a report to moodle

Homework 1 – Submission Format

submission.txt

```
Query,RetrievedDocuments
20001.query,VOM19980619.0700.0347 VOM19980225.0700.0510 VOM19980317.0900.0192 VOM19980317.0900.0330 VOM19980
00.0173 VOM19980302.0700.0241 VOM19980303.0700.2287 VOM19980530.0730.0166 VOM19980404.0700.2088 VOM19980616.0
216 VOM19980614.0700.0357 VOM19980626.0700.0409 VOM19980403.0700.0489 VOM19980523.0730.0220 VOM19980524.0730.
VOM19980624.0900.0077 VOM19980625.0700.0363 VOM19980605.0730.0152 VOM19980602.0730.0102 VOM19980603.0730.0280
9980522.0730.0037 VOM19980228.0700.0327 VOM19980414.0900.0260 VOM19980223.0700.0765 VOM19980505.0700.0529 VOM
503.0730.0136 VOM19980319.0900.3416 VOM19980620.0730.0034 VOM19980302.0700.0209 VOM19980302.0900.2091 VOM1998
0900.0207 VOM19980305.0900.1926 VOM19980521.0730.0029 VOM19980504.0700.0376 VOM19980314.0700.0239 VOM19980619.
.0137 VOM19980611.0700.0150 VOM19980326.0700.2112 VOM19980522.0900.0269 VOM19980503.0700.0412 VOM19980428.090
4 VOM19980422.0900.0021 VOM19980605.0700.0194 VOM19980611.0700.0046 VOM19980223.0700.2728 VOM19980614.0730.02
M19980303.0700.0696 VOM19980326.0900.0149 VOM19980505.0700.0481 VOM19980614.0730.0034 VOM19980226.0900.1964 VO
80523.0730.0083 VOM19980316.0700.0356 VOM19980609.0900.0009 VOM19980314.0700.2300 VOM19980302.0700.2137 VOM19
4.0700.0458 VOM19980319.0900.2169 VOM19980305.0700.2126 VOM19980515.0700.0472 VOM19980403.0700.0129 VOM19980
30.0142 VOM19980618.0700.0234 VOM19980319.0900.0647 VOM19980527.0700.0528 VOM19980607.0730.0033 VOM19980305.0
20002.query,VOM19980530.0730.0101 VOM19980611.0900.0216 VOM19980506.0900.0089 VOM19980624.0700.0434 VOM19980
00.2021 VOM19980604.0900.0246 VOM19980606.0700.0562 VOM19980303.0900.2085 VOM19980225.0700.0999 VOM19980312.0
171 VOM19980220.0900.1979 VOM19980305.0700.0763 VOM19980627.0700.0360 VOM19980225.0700.0302 VOM19980529.0700.
VOM19980612.0730.0192 VOM19980319.0700.2737 VOM19980630.0700.0071 VOM19980526.0730.0131 VOM19980403.0700.0489
9980430.0900.0192 VOM19980502.0700.0307 VOM19980616.0700.0420 VOM19980319.0900.3468 VOM19980303.0900.1926 VOM
```

Questions?

kychen@mail.ntust.edu.tw