Classic Models

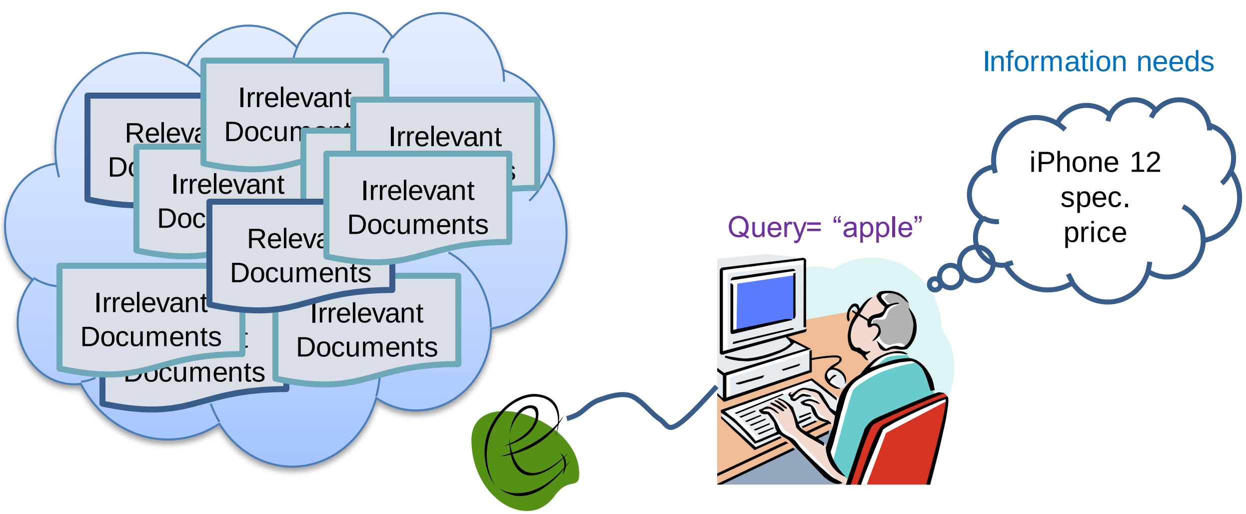
Kuan-Yu Chen (陳冠宇)

2020/10/16 @ TR-313, NTUST

Review

- Query & Information Need

- Relevance



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

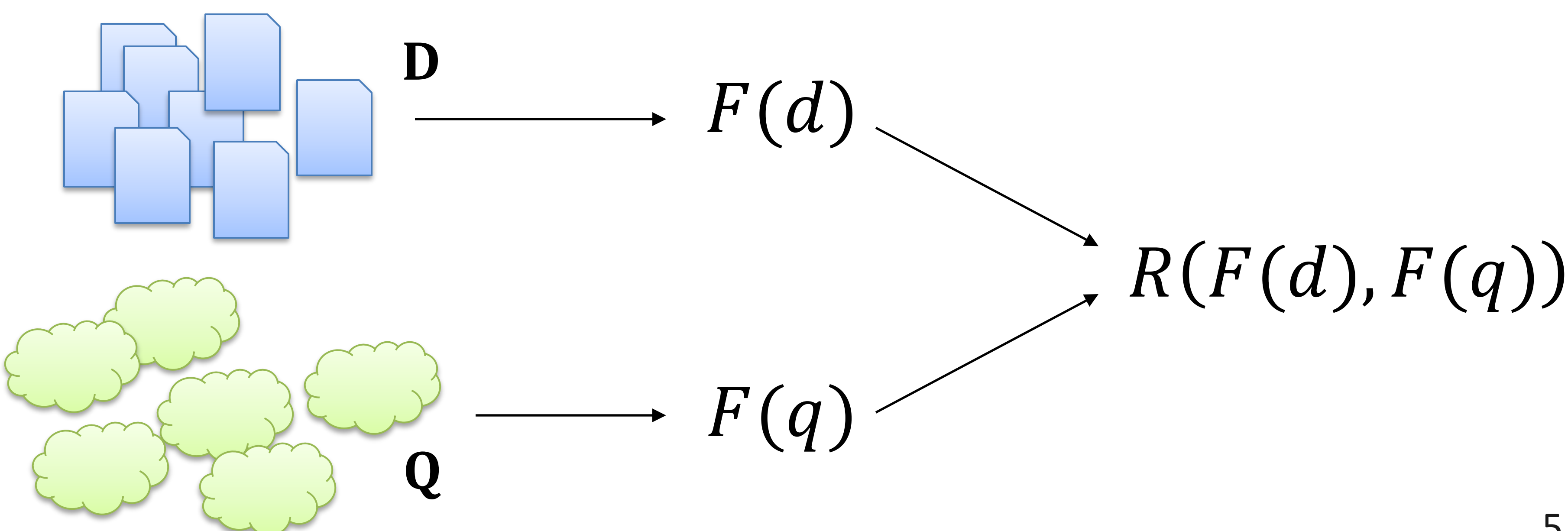
- A ranking is an ordering of the documents that reflects their relevance to a user query

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

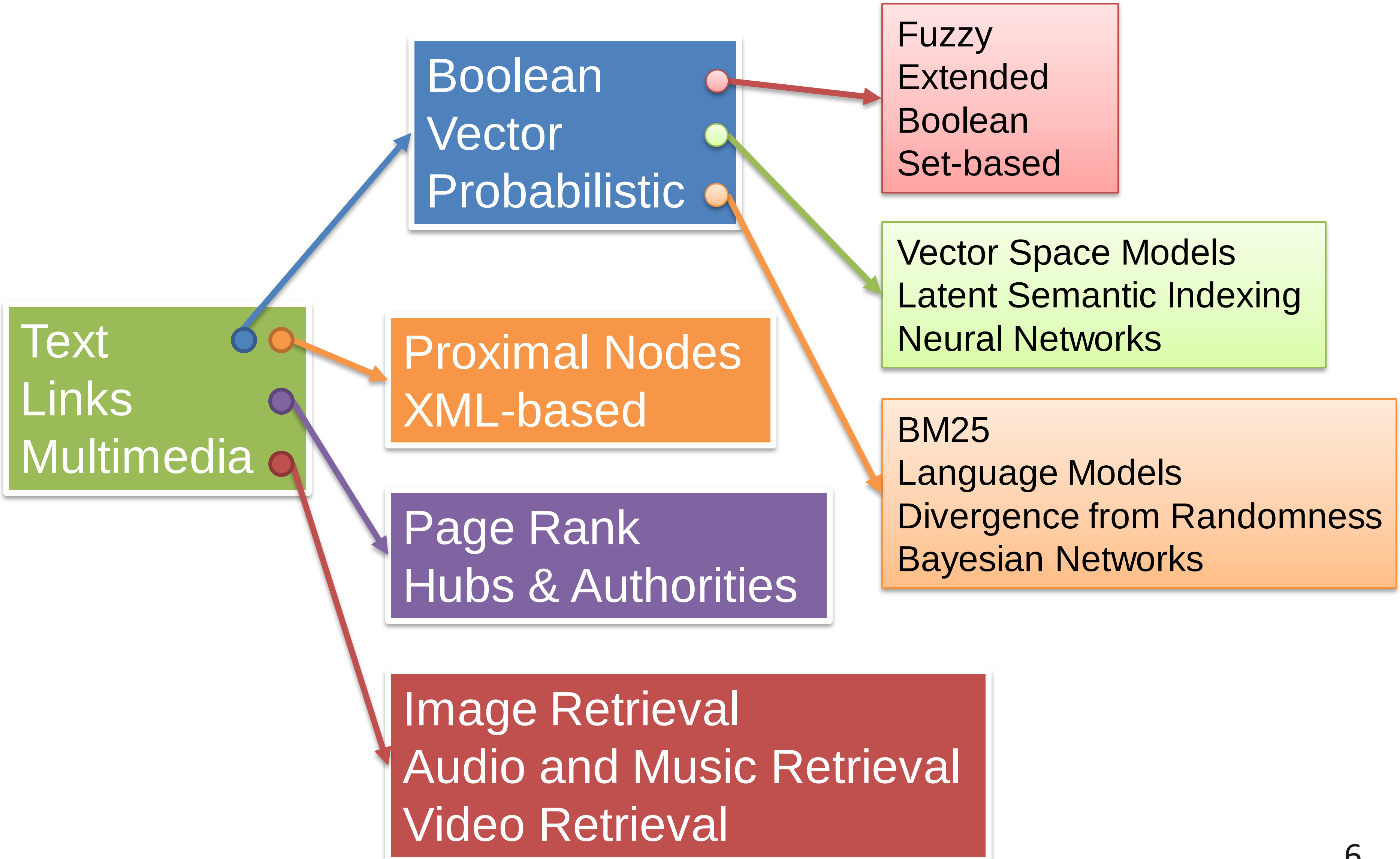
- This problem naturally embodies a degree of uncertainty, or vagueness
  - **Relevance!**

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

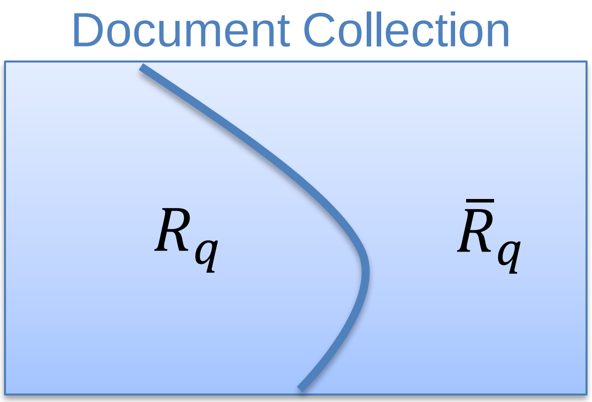
- The probabilistic model captures the IR problem using a **probabilistic framework**
  - Tries to estimate the **probability** that a document will be relevant to a user query
    - $P(R_q|d_j)$
  - Assumes that this probability depends on the query and document representations only
    - Hyper-links 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

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



Derivation

- By using Bayes' rule

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

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

Constant for the given query *q*

Probabilistic Model – 1

- The probabilistic model can be computed by

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

  - $P(d_j|R_q)$ 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

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

  - $P(w_i|R_q)$ is the probability that the term w_i is present in a document randomly selected from R_q
  - $P(\bar{w}_i|R_q)$ 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

Probabilistic Model – 3

- Since we assume index terms follow the Bernoulli distributions

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

- The probabilistic model can be translated to:

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

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

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

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

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

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

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

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

Probabilistic Model – 6

- Consequently, we can obtain

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

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

Constant for any 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
  - Here, we derive the **Binary Independence Model**

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

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
  - Here, the **Robertson-Sparck Jones Equation** is derived

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

	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**!

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

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

	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

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

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

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

- Advantages:
  - Documents can be ranked in decreasing order of probability of relevance

- Disadvantages:
  - need to estimate $P(w_i|R_q)$
    - Only approximation can be calculated
  - method does not take "term 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**
  - Stop words!

- 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

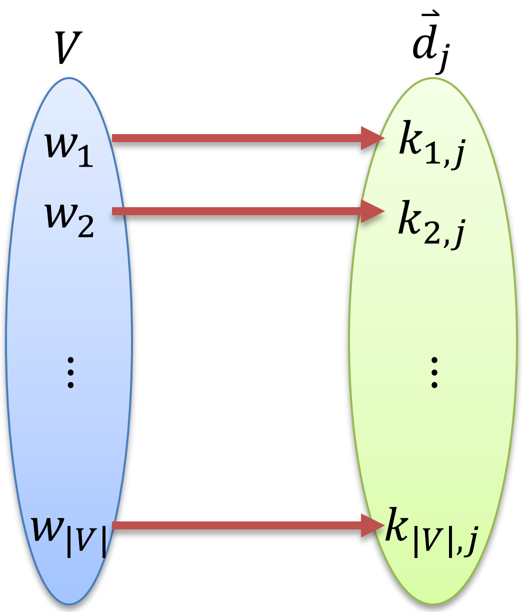
- To characterize term importance, we associate a weight $k_{i,j} > 0$ with each term w_i that occurs in the document d_j
  - If w_i that does not appear in the document d_j, then $k_{i,j} = 0$

- 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
  - For dictionary terms that do not occur in the document, this weight is zero



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	$1 + log_2(tf_{i,j})$
Double Normalization 0.5	$0.5 + 0.5 \frac{tf_{i,j}}{max_j tf_{i,j}}$
Double Normalization σ	$\sigma + (1 - \sigma) \frac{tf_{i,j}}{max_j tf_{i,j}}$

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

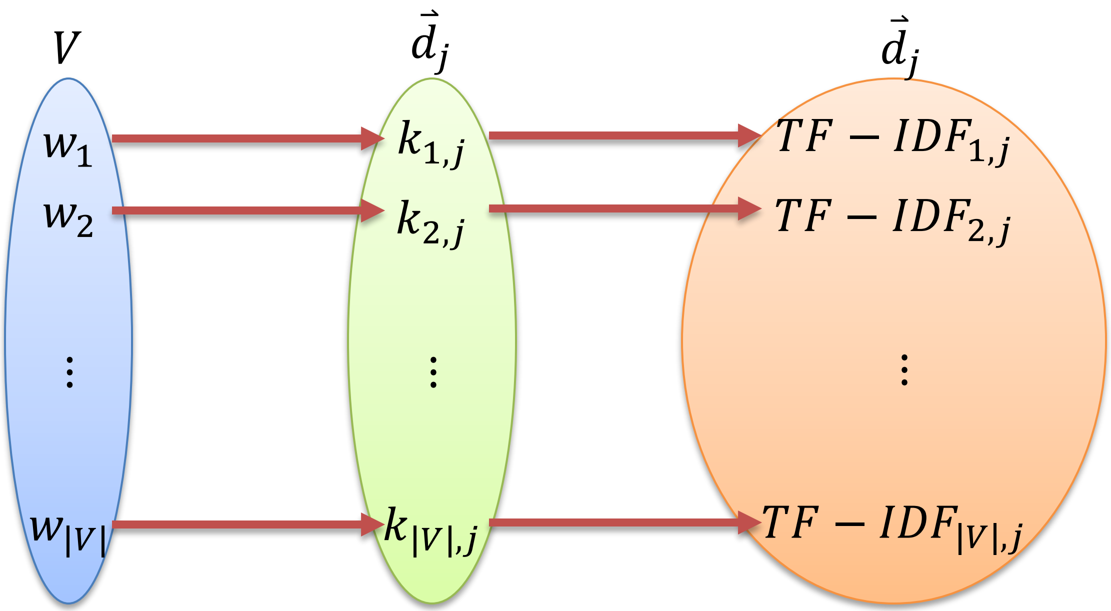
- We now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document

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

  - $TF - IDF_{i,j}$ 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 – 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**



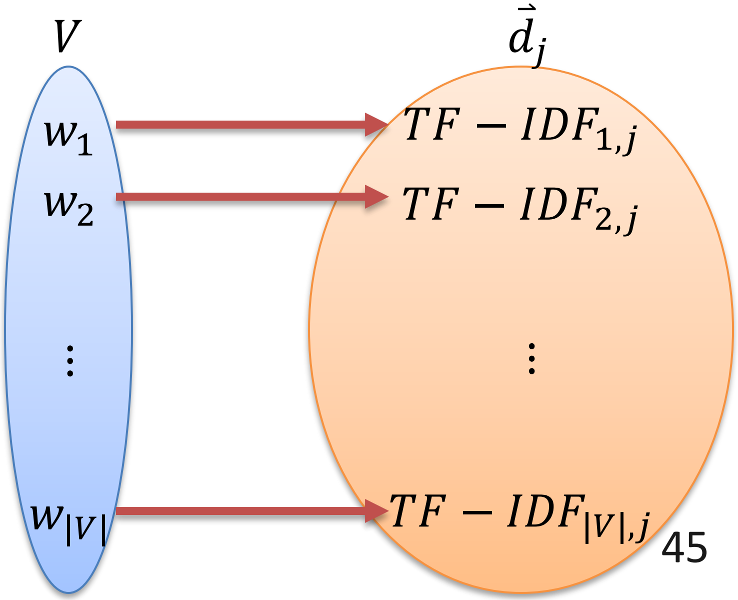
Overlap Score Model – 2

- 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!**

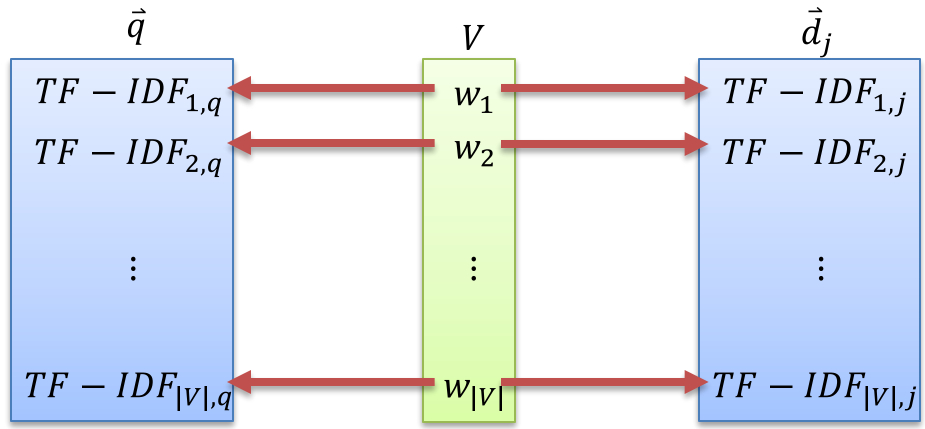
$$sim(d_j, q) \approx \sum_{w_i \in d_j \& w_i \in q} log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$



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 $TF - IDF$



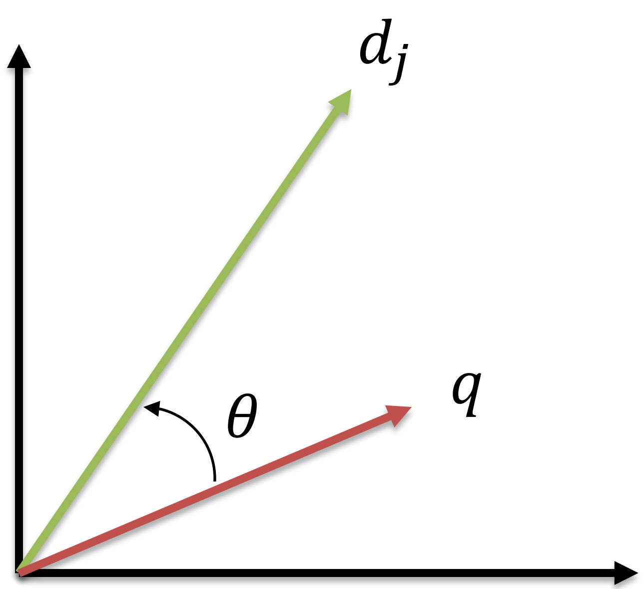
- The relevance degree between a given query and a document can be computed by referring to the cosine similarity measure

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

The Vector Space Model – 2

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

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



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

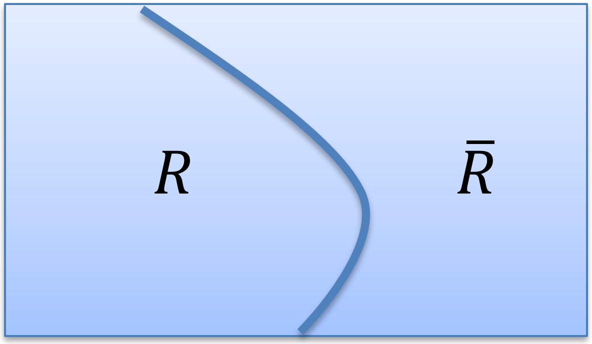
- Disadvantages
  - It assumes independence of index terms

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

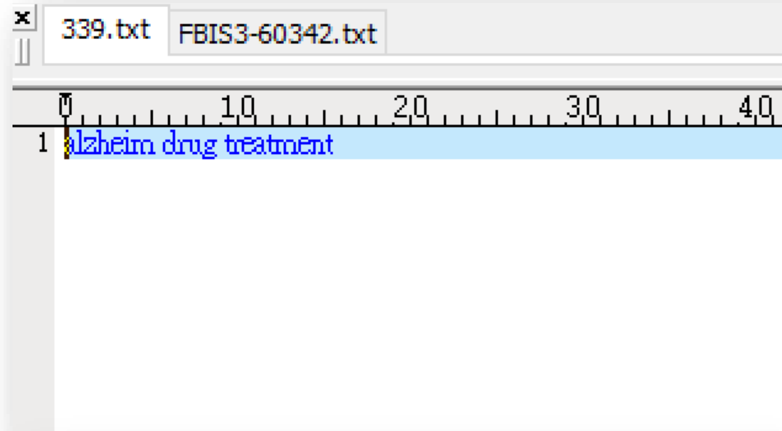
- Boolean model does not provide for partial matches and is considered to be the weakest classic model
- There is some controversy as to whether the **probabilistic model outperforms the vector space model**
  - Bruce Croft suggested that the probabilistic model provides a better retrieval performance
  - Salton *et al.* showed that the **vector space model outperforms probabilistic model** with general collections

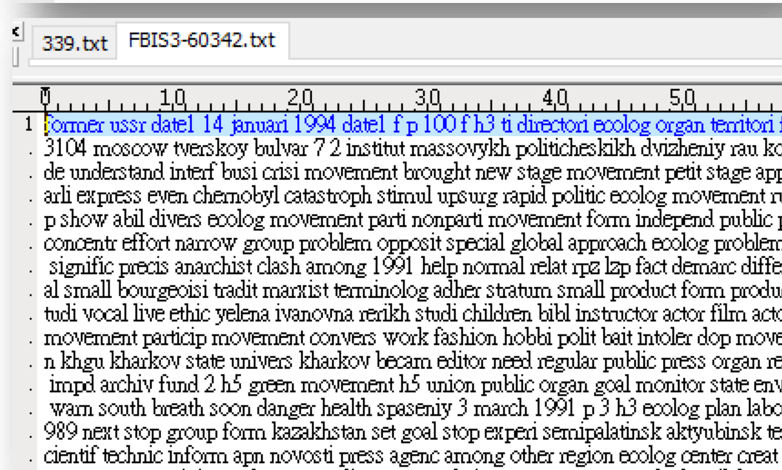
Homework 1 – Vector Space Model

Homework 1 – Description.

- In this project, you will have
  - 50 Queries
  - 4191 Documents

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





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

Homework 1 – Description..

- The evaluation measure is MAP
  - The **hard** deadline is 10/29 23:59
  - Please submit a **report** and your **source codes** to the Moodle system, otherwise you will get 0 point
  - You can get 13 points if you **outperform the baseline**
  - The report will be judge by TA, and you will get 1~2 points

- You should
  - Upload your answer file to kaggle
    - https://www.kaggle.com/t/7f84706b7b074267ae314582825fb725
    - The maximum number of daily submissions is 20
    - **Your team name is ID_Name**

      M123456_陳冠宇

Homework 1 – Submission Format

Questions?

kychen@mail.ntust.edu.tw