www.jntuworldupdates.org

Introduction to Automatic Indexing

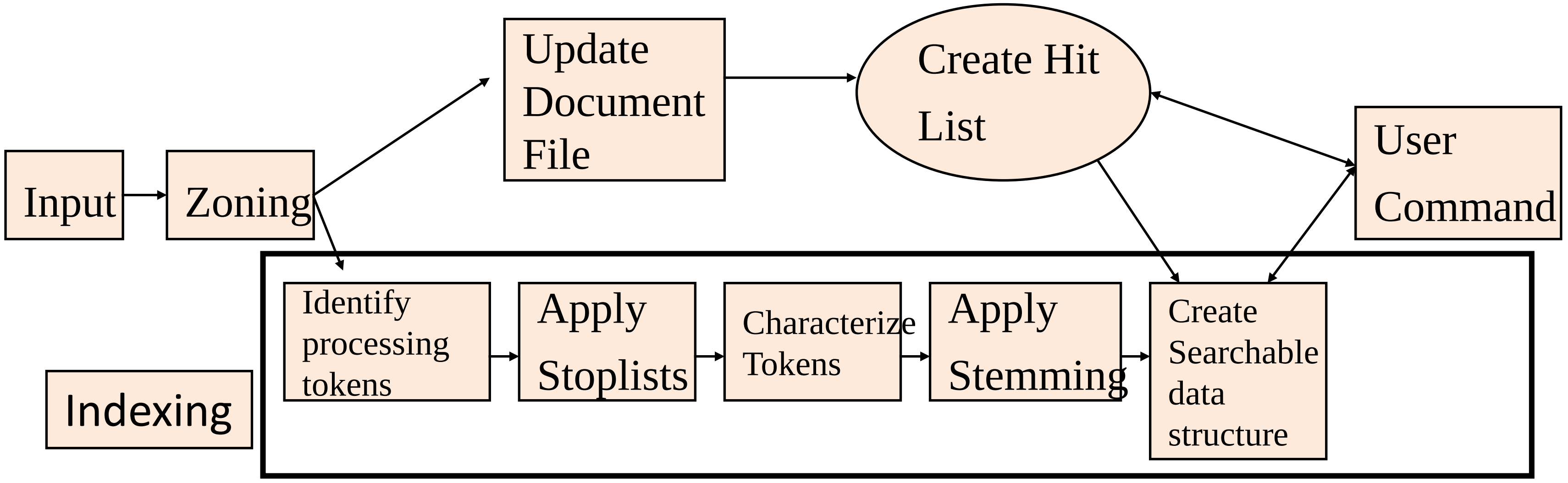
www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Overview

- The indexing process is a transformation of an item that extracts the semantics of the topics discussed in the item
- The extracted information is used to create the processing tokens and the searchable data structure.
- Automatic indexing is the process of analyzing an item to extract the information to be permanently kept in an index.
- Search Strategies are classified as statistical, natural language and concept.



Automatic Indexing Approaches

www.jntuworldupdates.org

- Statistical strategies
  - Most prevalent in commercial system
  - Cover broadest range of indexing technology
  - Approach
  - Use frequency of occurrence of events
    - » Events are related to occurrences of Processing Tokens(PT)s within documents and within the database
  - Store a single statistic, such as how often each word occurs in an item, that is used in generating relevance scores after a standard Boolean search
  - Statistics applied to the event data are probabilistic, Bayesian, vector space, neural net
    - » Static approach stores occurrence of each word in an item, that is used in generating relevance scores after a standard Boolean search.
    - » Probabilistic indexing stores the probability that a particular item satisfies a particular query
    - » Bayesian and Vector approaches store information used in generating a relative confidence level of an items relevance to a query.
    - » Neural Net are dynamic learning structures used to determine concept classes.

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Automatic Indexing Approaches (Cont.)

- Natural language
  - Additionally perform varying levels of natural language parsing of the item for disambiguating the context of the PTs and generalizes to more abstract concepts within an item (e.g., present, past, future actions)
    - This additional information is stored within the index to be used to enhance the search precision
- Concept indexing
  - Use the words within an item to correlate to concepts discussed in the item
  - A generalization of the specific words to values used to index the item
- Hypertext Linkages
  - Provide virtual threads of concepts between items versus directly defining the concept within an item.
- To maximize location of relevant items, applying several different algorithms to the same corpus provides the optimum results, but the storage and processing overhead is significant

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Statistical Indexing – Vector Weighting

Overview

www.jntuworldupdates.org

- The semantics of every item are represented as a vector
- A vector is a one-dimensional set of values, where the order/position of each value in the set is fixed and represents a particular domain
- In IR, each position in the vector typically represents a PT(Processing Token).
- Two approaches to the domain values
  - Binary: the domain contains the the value of one or zero
    - 1: represent the existence of the PT in the item
  - Weighted: the domain is the set of all real positive numbers
    - Relative importance of that PT in representing the semantics of the item (provide a basis for determining the rank of an item)
  - Ex: discuss petroleum refineries in Mexico

	Petroleum	Mexico	Oil	Taxes	Refineries
Binary	1	1	1	0	1
Weighted	2.8	1.6	3.5	.3	3.1

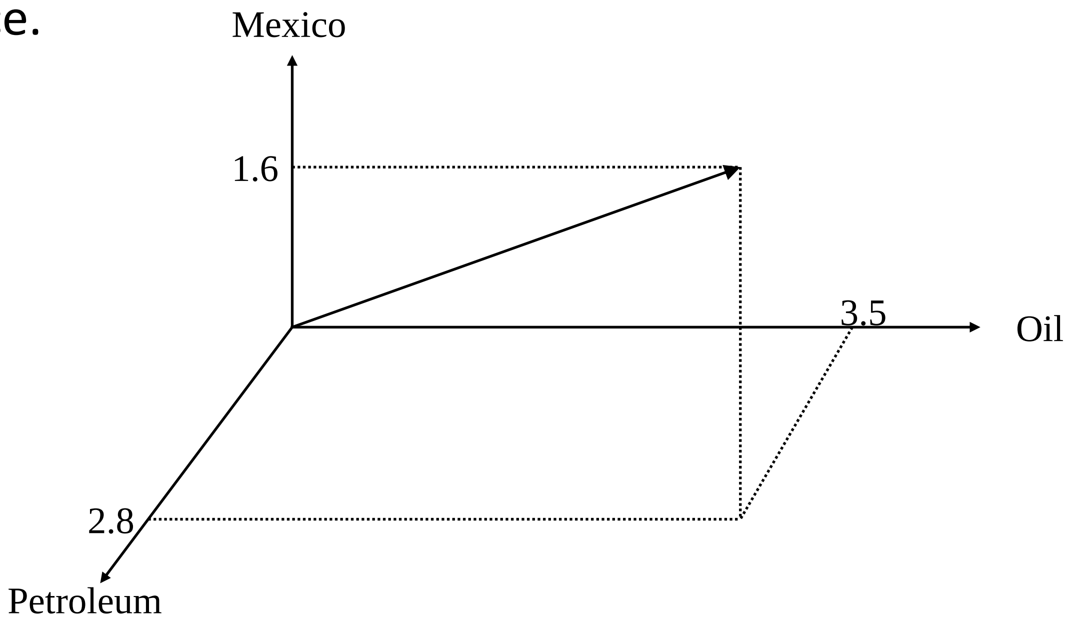
www.specworld.in

www.smartzworld.com

Overview (Cont.)

www.jntuworldupdates.org

– Each processing token can be considered a dimension in an item representation space.



– There are many algorithms that can be used in calculating the weights of Processing Tokens.

- Simple Term Frequency Algorithm
- Inverse Document Weighting
- Signal Weighting
- Discrimination Value

www.specworld.in

www.smartzworld.com

Simple term frequency Algorithm

www.jntuworldupdates.org

- The term frequency $\mathrm{tf}_{t,d}$ of term *t* in document *d* is defined as the number of times that *t* occurs in *d*.
- The weight is equal to the term frequency
  - TF (Term Frequency) : Frequency of Occurrence of PT in an existing item
  - TOTF (Total Term Frequency) : Frequency of Occurrence of PT in an existing database
  - IF /DF (Item / Document Frequency) : No. of Unique items in the database that contain the processing token
- Emphasize the use of particular PT within an item
  - "computer" occurs 15 times within an item ➔ a weight of 15
- Problems: normalization between items and use of the PT within the database
  - The longer an item is, the more often a PT may occur within the item
  - Use of absolute value biases weights toward longer items.
  - Hence Normalization is done
  - Ex

(1+log(TF)/1+log(average(TF))

(1-slope)*pivot + slope * no. of unique terms

Pivot : avg no. of unique terms occurring in the collection.

Simple Term Frequency Algorithm contd..

Maximum Term Frequency

- In the first technique, the term frequency for each word is divided by the maximum frequency of the word in any item. This normalizes the term frequency values to a value between zero and one.
- problem : the maximum term frequency can be so large that it decreases the value of term frequency in short items to too small a value and loses significance.

Logaritmetic Term Frequency

- In this technique the log of the term frequency plus a constant is used to replace the term frequency. The log function will perform the normalization when the term frequencies vary significantly due to size of documents.

COSINE function

- COSINE function used as a similarity measure can be used to normalize values in a document.
- This is accomplished by treating the index of a document as a vector and divide the weights of all terms by the length of the vector. This will normalize to a vector of maximum length one.
- This uses all of the data in a particular item to perform the normalization and will not be distorted by any particular term.

www.jntuworldupdates.org

Simple Term Frequency Algorithm contd…

- The problem occurs when there are multiple topics within an item. The COSINE technique will normalize all values based upon the total length of the vector that represents all of topics.
- If a particular topic is important but briefly discussed, its normalized value could significantly reduce its overall importance in comparison to another document that only discusses the topic.

penalizing long documents

- Singhal did experiments that showed longer documents in general are more likely to be relevant to topics then short documents.
- **normalization was making all documents appear to be the same length.**
- To compensate, a **correction factor** was defined that is based upon document length that maps the Cosine function into an adjusted normalization function.
- The function determines the document length crossover point for longer

  documents where **the probability of relevance equals the probability of retrieval, (given a query set).**
- This value called the "pivot point" is used to apply an adjustment to the normalization process.

Simple Term Frequency Algorithm contd…

www.jntuworldupdates.org

- The theory is based upon straight lines so it is a matter of determining slope of the lines.

New normalization = (slope)*(old normalization)+k

- K is generated by the rotation of the pivot point to generate the new line and the old normalization = the new normalization at that point.

- Substituting pivot for both old and new value in the above formula we can solve for K at that point.

- Then using the resulting formula for K and substituting in the above formula produces the following formula:

New normalization = (slope)*(old normalization)+(1.0-slope)*(pivot)

Slope and pivot are constants for any document/query set.

- **problem:** Cosine function favors short documents over long documents and also favors documents with a large number of terms.
- In documents with large number of terms the Cosine factor is approximated by the square root of the number of terms.
- This suggests that using the ratio of the logs of term frequencies would work best for longer items in the calculations:

www.specworld.in

www.smartzworld.com

Simple Term Frequency Algorithm contd

If log(TF) is used instead of the normal frequency then TF is not a significant factor. In documents with large number of terms the Cosine factor is approximated by the square root of the number of terms. This suggests that using the ratio of the logs of term frequencies would work best for longer items in the calculations:

(1 + log(TF))/(1 + log(average(TF))

This leads to the final algorithm that weights each term by the above formula divided by the pivoted normalization:

((1 + log(TF))/(1 + log(average(TF))/(slope)(No. unique terms) + (1-slope)*(pivot)

Singhal demonstrated the above formula works better against TREC data then TF/MAX(TF) or vector length normalization. The effect of a document with a high term frequency is reduced by the normalization function by dividing the TF by the average TF and by use of the log function.

www.jntuworldupdates.org

Document frequency

- Rare terms are more informative than frequent terms
- Consider a term in the query that is rare in the collection (e.g., *arachnocentric*)
- A document containing this term is very likely to be relevant to the query *arachnocentric*
- → We want a high weight for rare terms like *arachnocentric.*

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Document frequency, continued

- Frequent terms are less informative than rare terms
- Consider a query term that is frequent in the collection (e.g., *high*, *increase*, *line*)
- A document containing such a term is more likely to be relevant than a document that doesn't
- But it's not a sure indicator of relevance.
- → For frequent terms, we want high positive weights for words like *high*, *increase*, *and line*
- But lower weights than for rare terms.
- We will use document frequency (df) to capture this.

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

idf weight

- df$_t$ is the document frequency of t: the number of documents that contain t
  - df$_t$ is an inverse measure of the informativeness of t
  - df$_t$ $\leq N$
- We define the idf (inverse document frequency) of t by

$$\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$$

  - We use log (N/df$_t$) instead of N/df$_t$ to "dampen" the effect of idf.

Will turn out the base of the log is immaterial.

www.specworld.in

www.smartzworld.com

idf example, suppose N = 1 million

term	df_t	idf_t
calpurnia	1	
animal	100	
sunday	1,000	
fly	10,000	
under	100,000	
the	1,000,000	

$$\text{idf}_t = \log_{10}(N/\text{df}_t)$$

There is one idf value for each term *t* in a collection.

www.jntuworldupdates.org

•The term “computer” represents a concept used in an item, but it does not help a user find the specific information being sought since it returns the complete database.

•This leads to the general statement enhancing weighting algorithms that the <u>weight assigned to an item should be inversely proportional to the frequency of occurrence of an item in the database.</u>

www.specworld.in

www.smartzworld.com

Inverse Document Frequency (IDF)

www.jntuworldupdates.org

- The weight equal to the frequency of occurrence of the processing token in the database
- $WEIGHT_{ij}=Tf_{ij}*[Log_2(n)-Log_2(IF_j)+1]$
  - $WEIGHT_{ij}$: assigned to term "j"in item "i"
  - TF_{ij} : frequency of term "j" in item "i"
  - IF_{ij} : number of items in the database that have term "j" in them
  - n : number of items in the database

	n	TF	IF
Oil	2048	4	128
Mexico	2048	8	16
Refinery	2048	10	1024

Assume that the term “oil” is found in 128 items, “Mexico” is found in 16 items and “refinery” is found in 1024 items. If a new item arrives with all three terms in it, “oil” found 4 times, “Mexico” found 8 times, and “refinery found 10 times and there are 2048 items in the total database, Figure 5.4 shows the weight calculations using inverse document frequency.

$$\textbf{Weight}_{oil} = 4 * (\text{Log}_2(2048) - \text{Log}_2(128) + 1) = 4 * (11 - 7 + 1) = 20$$

$$\textbf{Weight}_{Mexico} = 8 * (\text{Log}_2(2048) - \text{Log}_2(16) + 1) = 8 * (11 - 4 + 1) = 64$$

$$\begin{aligned}\textbf{Weight}_{refinery} &= 10 * (\text{Log}_2(2048) - \text{Log}_2(1024) + 1) = \\ &\quad 10 * (11 - 10 + 1) = 20\end{aligned}$$

with the resultant inverse document frequency item vector = (20, 64, 20)

www.jntuworldupdates.org

Effect of idf on ranking

- Does idf have an effect on ranking for one-term queries, like
  - iPhone
- idf has no effect on ranking one term queries
  - idf affects the ranking of documents for queries with at least two terms
  - For the query capricious person, idf weighting makes occurrences of capricious count for much more in the final document ranking than occurrences of person.

Collection vs. Document frequency

- The collection frequency of *t* is the number of occurrences of *t* in the collection, counting multiple occurrences.
- Example:

Word	Collection frequency	Document frequency
insurance	10440	3997
try	10422	8760

- Which word is a better search term (and should get a higher weight)?

www.jntuworldupdates.org

Signal Weighting

- IDF does not account the term frequency distribution of the PT in the items that contain the term
- The distribution of the frequency of processing tokens within an item can affect the ability to rank items

Item Distribution	SAW	DRILL
A	10	2
B	10	2
C	10	18
D	10	10
E	10	18

❖An instance of an event that occurs all the time has less information value than an instance of a seldom occurring event

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Signal Weighting (Cont.)

- In information theory, the information content value of an object is inversely proportional to the probability of occurrence of the item
  - INFORMATON = -Log$_2$(p)
    - p is the probability of occurrence of event p
      - p = 0.5% → INFORMATION = -Log$_2$(0.005) = -(-10) = 10
      - p = 50% → INFORMATION = -Log$_2$(0.5) = -(-1) = 1
- If there are many independent occurring event

$$AVE\_INFO = -\sum_{k=1}^{n} p_k Log_2(p_k)$$

  - Maximum when the value for every p_k is the same
  - p_k can be defined as $TF_{ik}/TOTF_k$

www.jntuworldupdates.org

Signal Weighting (Cont.)

$$Singal_k = Log_2(TOTF) - AVE\_INFO$$

$$Weight_{ik} = TF_{ik} * Signal_k$$

$$Weight_{ik} = TF_{ik} * \left[Log_2(TOTF_k) - \sum_{i=1}^{n} {}^{TF_{ik}}\!\big/_{TOTF_k} * Log_2(TF_{ik} / TOTF_k) \right]$$

$$Singal_{SAW} = Log_2(50) - \left\lfloor 5 * \frac{10}{50} * Log_2(10/50) \right\rfloor$$

$$Singal_{DRILL} = Log_2(50) - \left[\begin{array}{l} \frac{2}{50} * Log_2(2/50) + \frac{2}{50} * Log_2(2/50) + \\ \frac{18}{50} * Log_2(18/50) + \frac{10}{50} * Log_2(10/50) + \frac{18}{50} * Log_2(18/50) \end{array} \right]$$

Similarity Measure

- Measure the similarity between a query and a document
- Similarity measure examples

$$SIM(DOC_i, QUERY_j) = \sum_k (DTerm_{i,k})(QTerm_{j,k})$$

$$SIM(DOC_i, QUERY_j) = \frac{\sum_k (DTerm_{i,k})(QTerm_{j,k})}{\sqrt{\sum_k (DTerm_{i,k})^2 * \sum_k (QTerm_{j,k})^2}}$$

www.jntuworldupdates.org

Problems with Weighting Schemes

- The two weighting schemes, IDF and signal, use total frequency and item frequency factors which makes them dependent on distributions of PTs within the DB
  - These factors are changing dynamically
- Approaches to compensate for changing values
  - Ignore the variances and calculates weights based on current values, with the factors changing over time. Periodically rebuild the complete search database
  - Use a fixed value while monitoring changes in the factors. When the changes reach a certain threshold, start using the new value and update all existing vectors with the new value
  - Store the invariant values (e.g. TF) and at search time calculate the latest weights for PTs in items needed for search terms

www.jntuworldupdates.org

Problems with Weighting Schemes (Cont.)

- Side effect of maintaining currency in the DB for term weights
  - The same query over time returns a different ordering of items
  - A new word in the DB undergoes significant changes in its weight structure from initial introduction until its frequency in the DB reaches a level where small changes do not have significant impact on changes in weight values

www.specworld.in

www.smartzworld.com

www.jntuworldupdates.org

Problems with Vector Model

- A major problem comes in the vector model when there are multiple topics being discussed in a particular item
  - Assume an item has an in-depth discussion of “oil” in “Mexico” and also “coal” in “Pennsylvania”
    - This item results in a high value in a search for “coal in Mexico”
- Cannot handle “proximity searching”