Lezione 2 - Dati Multimediali: Il Testo

https://www.youtube.com/watch?v=yGEeVNcaNJU

Introduzione

Il MIRS e i DBMS sono molto diversi fra di loro:

- Il MIRS è specializzato nell'indicizzazione e nella ricerca dei dati multimediali
- Per la progettazione di un MIRS è necessario conoscere la struttura e le caratteristiche dei dati multimediali

Il testo

Il Testo Piano

> Il testo piano è il più semplice testo da analizzare. Si chiama "piano" perché non contiene altri attributi che lo possano connotare.

Il testo piano si rappresenta mediante l'ausilio della tabella del codice ASCII, inizialmente limitato a 7 bit, ma successivamente aumentato a 8 bit (extended ASCII)

Sono state successivamente create altre estensioni, come l'UNICODE, successivamente esteso per codifiche a 21 bit (circa 1 milione di caratteri)

Il Testo Strutturato

> Testo al quale sono stati aggiunti degli attributi. Anche detto Testo Formattato

Tipico testo formattato in diversi formati (.pdf, .doc, .tex)

Il Testo Compresso

> Testo che, sfruttando una ridondanza, permette di racchiudere un insieme numeroso di caratteri in un insieme più piccolo

La compressione è effettuata in modo LOSSLESS (senza perdita)

Lo svantaggio della compressione del testo dipende dal fatto che, per trattarlo, saranno necessari algoritmi di compressione, aumentando qundi la complessità di utilizzo

Metodi di compressione:

- Huffman
- Run Length
- Lempel-Ziv-Welch (LZW)

Codifica di Huffman

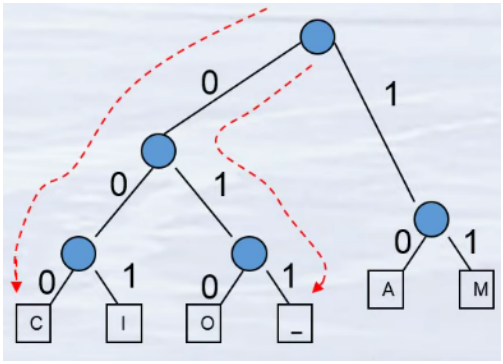
Tecnica basata sull'analisi statistica del dato da comprimere, in particolare sulla frequenza con la quale si ripetono i suoi elementi

La sua prestazione è proporzionata alla varianza delle frequenze (maggiore varianza → maggiore prestazione)

- Esempio di codifica usando la stringa → $S_{txt} = ciao\_mamma$

1. Trasformazione di una stringa in binario usando il codice ASCII (utilizzo 80 bit):

   S = 01000011010010010100111100100000010011010100000101001101010011010100000 1
2. Analisi delle frequenze e ordinamento decrescente
3. Raggruppamenti delle coppie con frequenza minore e creazione di grafici ad albero
4. Costruzione del codice e assegnazione del codice agli archi:
   a. Sinistra → 0
   b. Destra → 1



La nuova stringa ora sarà:

- S = 000001100100111110111110 (Siamo passati da 80 bit a 24)

C’è stato un guadagno del 70% rispetto alla stringa originale. Il codice ottenuto, però, non è univoco (la scelta del numero per gli alberi è arbitrario)

Questo codice è, per costruzione, univocamente decifrabile:

> Non ci sono ambiguità nella decifrazione. Ogni parola non è prefissa di nessun’altra parola (pesca - pescatore)

Codifica Run-Length

Altro metodo di compressione senza perdita.

Definiamo un RUN:

- Un RUN è una sequenza ripetuta di caratteri

Il run è rappresentato da una struct con 3 campi:

- Sc → Carattere speciale per l’identificazione della codifica
- X → Carattere ripetuto nel RUN
- C → Numero di ripetizione del carattere

Le migliori performance si avranno per run-length > 3 (non ≥)

Esempio:

- La stringa “eeeeeeetnnnnnnnn” la trasformo in → “@e7t@n8”

Codifica LZW

Codifica lossless che sfrutta la ripetizione di gruppi di frasi. Quindi è un’estensione del LRE

- Il compressore esamina la presenza delle frasi incontrate con le frasi presenti in un dizionario inizialmente vuoto
  - Se la frase è presente nel dizionario viene restituito in output un codice C associato alla frase F
  - Se la frase non è presente nel dizionario:
    - Inserisce la frase nel dizionario
    - Associa un codice C alla frase
    - Manda in output C