UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN
INFORMATICA

Data Quality Analisys tramite algoritmi di String-Matching

Relatori:
Chiar.mo prof. **Michele SCALERA**
Dott. **Giovanni BRUNO**

Laureando: **Simone De Nisi**

SERLAB
SOFTWARE ENGINEERING RESEARCH LABORATORY
c/o DIPARTIMENTO DI INFORMATICA
via Orabona, 4 - 70126 - Bari
Tel.: + 39.080.5443279 Fax: + 39.080.5442536

INDICE

- Data Quality
- Obiettivo
- Ricerca
- Esperimento
- Sperimentazione
- Risultati
- Sviluppi futuri

Data Quality

Un dato è considerato di qualità se adatto per determinate operazioni, decision making e pianificazione.

La gestione della qualità del dato si costituisce di due fasi:

- Data Quality Assurance:
  - è il processo di profilazione dei dati, volto a scoprire inconsistenze e anomalie nei dati.
- Data Quality Control:
  - è il processo volto a verificare l'adeguatezza del dato, con qualità nota, per un'applicazione, un processo o un determinato scopo.

2 standard ISO per la qualità dei dati:

⇨ ISO/IEC 25012:

- Definisce un modello di qualità in cui vengono stabilite quali sono le caratteristiche che devono prese in considerazione, quando vengono analizzati i dati.

⇨ ISO/IEC 25024:

- Definisce le misure di qualità per quantificare la qualità dei dati in termini di caratteristiche definite nella ISO 25012.

Obiettivo

- → Sviluppare un sistema di Natural Language Processing (NLP) tramite string-matching
  - → NLP è l'elaborazione di una grossa mole di dati in linguaggio naturale

Il sistema deve permettere:

- Effettuare controlli su stringhe;
- Consentire l'analisi di dettaglio tramite generazione di report;
- Consentire l'analisi in grande tramite consultazione di grafici;
- Permettere la gestione delle eccezioni.

Ricerca

Una tecnica utilizzata nel NLP è lo string-matching.

- → L'obiettivo dello string-matching è trovare tutte le occorrenze di un dato pattern come sottostringa di un'altra stringa più lunga.

Dall'analisi della letteratura esistente, è stato possibile suddividere gli algoritmi di string-matching in 3 categorie:

- ⇨ Algoritmi fonetici;
- ⇨ Algoritmi token-based;
- ⇨ Algoritmi basati su confronto.

Algoritmi fonetici

Un algoritmo fonetico codifica la parola in base alla sua pronuncia.

- Algoritmi estremamente complessi:
  - Enorme quantità di regole ed eccezioni.
    - Forte specializzazione

Due ambiti principali:

- Spell checking:
  - vengono cercate le parole che hanno una codifica simile ad una parola misspelled.
- Search functionality:
  - vengono trovate tutte le varianti di una parola con la codifica uguale

- Soundex
  - Confronto fonetico basato sulla lingua inglese
  - Ottimi risultati nella codifica di nomi americani (associato allo U.S. census)
- Metaphone
  - Confronto fonetico multilingua (dalla 2ª versione)
  - Restituisce una duplice codifica per le parole con più varianti
    - Es. 'Smith' restituisce 'SM0' e 'XMT', mentre 'Schmidt' restituisce 'XMT' e 'SMT'
- Caverphone
  - Confronto fonetico basato sulla lingua inglese
  - Ottimizzato per la zona in cui è stato sviluppato (parte sud di Dunedin, Nuova Zelanda
- NYSIIS
  - Specializzato nella codifica di nomi e cognomi dello stato di New York

Algoritmi token-based

In un algoritmo token-based, ogni stringa viene divisa in parole o n-grams (sottostringhe di n lettere consecutive).

- Due stringhe sono uguali se hanno le stesse parole o n-grams.
- Necessaria una tecnica di segmentazione per dividere la stringa:
  - Nelle lingue latine, lo spazio produce ottimi risultati.

Due limiti principali:

- Diversa rappresentazione di una parola:
  - Es. 'icebox' può diventare 'ice-box' o 'ice box';
- Lingue che tendono ad unire più parole insieme:
  - Es. tedesco.

- N-gram
  - Un n-gram è una sequenza continua di n elementi di una stringa
  - Rappresentazione in n-gram di una stringa consente il confronto con un'altra stringa
- Coefficiente di Dice
  - Variazione di N-gram
  - Prende in considerazione solo gli elementi in comune
- Distanza di Jaro-Winckler
  - Dato da (1- 'Distanza di Jaro')
    - La distanza di Jaro tra due stringhe, è il numero minimo di trasposizioni di ogni carattere per trasformare una parola in un'altra
- Coefficiente di Ratcliff/Obershelp
  - Basato sulla Common Longest Subsequence (CLS)

Algoritmi basati su confronto

- Gli algoritmi basati su confronto definiscono una o più operazioni elementari da eseguire per trasformare la stringa A nella stringa B.
- Restituiscono, quindi, un numero, che rappresenta il numero di operazioni elementari necessarie.
- Quanto più basso è il numero restituito, tanto più simili sono le due stringhe.

- Distanza di Hamming
  - La distanza di Hamming tra due stringhe di ugual lunghezza è il numero di posizioni nelle quali i simboli corrispondenti sono diversi
  - Misura il numero di sostituzioni necessarie per convertire una stringa nell'altra
- Distanza di Lavenshtein
  - La distanza di Levenshtein tra due stringhe A e B è il numero minimo di modifiche elementari che consentono di trasformare la A nella B
    - Aggiunta di un carattere
    - Eliminazione di un carattere
    - Sostituzione di un carattere

Algoritmi ibridi

Ci sono inoltre, altri algoritmi, che utilizzano più tecniche:

- Language Tool
  - Profilazione delle parole. Es: Automobile, -> nome, femminile, singolare
  - Word frequency
  - Regole Xml
- JaSpell
  - Alberi di ricerca ternari (TST)
  - Word frequency
  - Metaphone

Per i controlli relativi agli indirizzi sonoutil sfruttati i servizi offerti da Google Maps

- Common Longest Subsequence (CLS)
- Web Service REST
- Limite di 2500 richieste al giorno

Esperimento

Il sistema, sviluppato interamente in Java, ha un duplice obiettivo:

- ⇨ Individuare errori e proporre correzioni tramite gli algoritmi individuati in letteratura
- ⇨ Allineare gli indirizzi al database di Google Maps

In entrambi i casi è possibile consultare sia i grafici (analisi macro), sia i report (analisi micro).

Inoltre, è possibile la gestione delle eccezioni, cioè parole che vengono escluse dai controlli.

Il download del report invece, dà informazioni circa gli errori individuati e la possibile correzione, oltre a dati riguardo l'analisi effettuata.

10 analyzed sentences.
10 sentences with mistakes.
308 milliseconds spent.

Durante lo sciopero, ci sono stati dei disrdini
Errore potenziale ai caratteri 39-47: Trovato un probabile errore di battitura
Correzione(i) suggerita(e): [disordini]

Un'altra importante funzionalità offerta è la possibilità di allineare il proprio database di indirizzi a quello di Google. In questo caso, eventuali errori vengono individuati e otteniamo una (o più) stringa secondo la formattazione di Google.

```
Via Benucci 8 Barletta
Possibile correzione:
Via Benucci, 8, 76121 Barletta BT, Italia
```

Sperimentazione

Nella fase di sperimentazione sono stati utilizzati:

- NACE rev. 2
  - Rilasciato dal NACE, conta circa 4000 stringhe;
- Vista del database degli studenti UniBa
  - Conta circa 4500 tuple, tuttavia:
    - Nome e cognome -> 4500 tuple
    - Data di nascita -> 4500 tuple
    - Corso di laurea -> 220 tuple.

Pertanto, sono state utilizzate circa 13000 stringhe.

In ogni fase della sperimentazione, sono stati analizzati tutti gli errori segnalati, con l'obiettivo di individuare tutti gli acronimi, le parole abbreviate e i falsi positivi, cioè quelle parole che venivano segnalate come errori, ma che in realtà, non lo erano. In particolare:

- Tutti gli acronimi sono stati estesi
- Tutte le abbreviazioni sono state estese
- Tutti i falsi positivi sono stati aggiunti alle rispettive liste di eccezioni.

Risultati

	Frasi prive di errore adeguatamente rilevate		Frasi con errore adeguatamente rilevate		Falsi positivi		Errori non rilevati		Correzione non adeguata	
	LT	JS	LT	JS	LT	JS	LT	JS	LT	JS
NACE Rev. 2	3743	3395	82	25	2	351	1	28	3	32
Nome e cognome	4325	4167	3	4	0	158	6	3	0	2
Data di nascita	4334	4334	0	0	0	0	0	0	0	0
Corso di laurea	200	183	2	0	0	17	0	2	0	0

I risultati ottenuti certificano la superiorità di Language Tool con una precisione globale del 98,5% mentre, JaSpell, raggiunge una precisione del 90,35%.

Il dataset utilizzato di 13000 stringhe conteneva 97 errori complessivi.

Dei 97 casi di errore, il sistema, è stato in grado di rilevarne e correggerne adeguatamente 90, cioè circa il 92.5%, aumentando così la qualità generale del dataset iniziale.

Sviluppi futuri

- Il progetto sviluppato rappresenta il primo step di una linea di ricerca del laboratorio SERLAB.
- Esso si presta, ovviamente, ad innumerevoli evoluzioni sia nell'ottica di espansione delle funzionalità che in quella dell'aggiunta di nuovi servizi.
- Per esempio, si potrebbero creare controlli ad hoc per altri tipi di dati, per esempio il codice fiscale, al fine di aumentare la qualità dei dati presenti nei database controllati dal sistema.
- Inoltre, il sistema potrebbe offrire la funzionalità di deduplica, aspetto chiave nella qualità dei dati.
- Tali sviluppi futuri sono già oggi tema di altre tesi di laurea sviluppate in ambito Serlab.