Università degli Studi di Roma Tre

Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche

Homework 5 per "Ingegneria dei Dati"

Data Integration su un insieme di dataset nel dominio delle aziende

Autori	*Matricole*
Luca Borrelli	559443
Rainer Cabral Ilao	560695

Sommario

Il progetto consiste nell'integrazione di 16 dataset contenenti informazioni su aziende, che presentavano un'elevata eterogeneità. L'obiettivo era:

- Analizzare le sorgenti dati e individuare le principali differenze e incongruenze.
- Definire uno schema mediato con almeno 20 attributi.
- Allineare gli schemi delle sorgenti allo schema mediato.
- Popolare lo schema mediato con i dati.
- Effettuare il record linkage e valutare la qualità del matching.

Per raggiungere questi obiettivi, sono stati utilizzati diversi strumenti e metodologie, tra cui **LLM per il mapping degli schemi**, **blocking strategies**, e il **Record Linkage Toolkit**.

Per maggiori dettagli, il codice sorgente del progetto è disponibile al seguente link GitHub, cliccando sull'icona si viene reindirizzati direttamente al repository.

`https://github.com/lukebo01/homework5`

Indice

1 Analisi delle sorgenti dati

I 16 dataset analizzati presentavano notevoli differenze nei nomi delle colonne, nei formati dei dati e nella granularità delle informazioni. L'unico attributo comune a tutti i dataset era il **name** dell'azienda. Tuttavia, altre informazioni come l'*industry*, la *location* e il *market capitalization* erano presenti in diversi formati e livelli di dettaglio.

Per affrontare questa eterogeneità, abbiamo effettuato un'analisi manuale per individuare corrispondenze tra le colonne dei diversi dataset. Ad esempio:

- In alcuni dataset, la localizzazione era un unico campo (*"New York, USA"*), mentre in altri era divisa in più colonne (*address, city, country*).
- La struttura delle informazioni finanziarie variava: alcune sorgenti fornivano solo il *market cap*, mentre altre avevano anche *revenue, net profit* e *assets*.
- Alcuni dataset contenevano informazioni specifiche su social media o URL aziendali, altri no.

1.1 Definizione dello schema mediato

Per gestire queste eterogeneità, abbiamo definito uno schema mediato contenente i seguenti 20 attributi principali:

Tabella 1.1: Schema Mediato Finale

Attributo	**Descrizione**
company_name	Nome dell'azienda
industry	Settore aziendale
headquarters_address	Indirizzo della sede
headquarters_city	Città della sede
headquarters_country	Paese della sede
year_founded	Anno di fondazione
ownership	Tipo di proprietà
company_number	Numero identificativo aziendale
employee_count	Numero di dipendenti
market_cap_usd	Capitalizzazione di mercato in USD
total_revenue_usd	Fatturato totale in USD
net_profit_usd	Utile netto in USD
total_assets_usd	Attivi totali in USD
company_website	URL del sito aziendale
social_media_links	Link ai social media
representative_name	Nome del CEO o rappresentante
total_raised	Capitale totale raccolto
company_description	Descrizione dell'azienda
company_stage	Stato dell'azienda (startup, corporate, etc.)
share_price	Prezzo delle azioni
legal_form	Forma legale dell'azienda

1.2 Allineamento delle sorgenti e popolazione dello schema mediato

Per allineare i dataset allo schema mediato, abbiamo utilizzato un **approccio basato su LLM (Large Language Models)**, in particolare **Gemini Flash**, che è stato interrogato con prompt contenenti i nomi delle colonne e un estratto di dati per inferire corrispondenze tra i campi.

Dopo la generazione iniziale del mapping JSON, lo abbiamo affinato manualmente per garantire la coerenza dei dati.

Successivamente, è stato sviluppato uno script Python per trasformare ogni dataset nel formato dello schema mediato. L'algoritmo prevedeva:

- Lettura dei file raw e parsing dei nomi delle colonne.
- Applicazione delle regole di trasformazione definite nel JSON di mapping.
- Salvataggio delle tabelle trasformate in formato CSV.

Infine, è stato creato un ulteriore script per combinare tutte le tabelle in un unico dataset unificato.

1.3 Creazione della Ground Truth

Per la creazione della *ground truth* necessaria alla valutazione del matching tra i record aziendali, è stato implementato un processo automatizzato, successivamente revisionato manualmente per correggere e perfezionare i risultati.

1.3.1 Creazione del File di Input

Il primo passo è stato la preparazione di un file raw contenente un subset dei dati rilevanti, come il nome dell'azienda, il settore, il paese della sede legale e la fonte del dato. Questo file è stato creato partendo dal dataset finale mediato, filtrando e ordinando i dati per nome azienda. Successivamente, i file di dati grezzi sono stati letti e convertiti in formato JSON per facilitare il loro utilizzo nei passaggi successivi.

1.3.2 Calcolo della Similarità

Per determinare le coppie di record potenzialmente riferite alla stessa azienda, è stata utilizzata la funzione `SequenceMatcher` della libreria Python `difflib`, che calcola un valore numerico di similarità tra due stringhe. La similarità tra i nomi delle aziende è stata calcolata per ogni coppia di record consecutivi, e solo le coppie con una similarità compresa tra 0.6 e 0.7 sono state prese in considerazione.

1.3.3 Creazione della Ground Truth Iniziale

Le coppie che soddisfacevano il criterio di similarità sono state inserite in una lista, marcando inizialmente ogni coppia come un *match* (ovvero come appartenente alla stessa azienda). I dati aggiuntivi, come il settore e il paese della sede legale, sono stati gestiti correttamente, includendo eventuali valori non disponibili (indicati come "nan") o valori di tipo lista.

1.3.4 Revisione Manuale

Successivamente, la *ground truth* è stata revisionata manualmente per verificare e correggere eventuali errori nei *match*. In questa fase, sono stati impostati i valori di `True` e `False` per i *match* in modo da perfezionare il dataset finale utilizzato per l'analisi del record linkage.

2 Tecniche di Blocking e Pairwise Matching

L'analisi e l'integrazione dei dataset aziendali ha richiesto l'implementazione di strategie avanzate di **record linkage** per identificare correttamente le corrispondenze tra record duplicati o riferiti alla stessa entità. Il processo si è articolato in due fasi principali:

- **Blocking**: riduzione dello spazio di ricerca per limitare i confronti tra record.
- **Pairwise Matching**: confronto tra coppie di record utilizzando metriche di similarità avanzate.

Queste tecniche hanno ottimizzato il matching dei dati bilanciando precisione, recall e costo computazionale.

2.1 Strategie di Blocking

Il **blocking** è una tecnica fondamentale per ridurre il numero di confronti tra record, evitando un confronto esaustivo tra tutte le possibili coppie di dati. Senza blocking, il numero di confronti necessari sarebbe proporzionale a $O(n^2)$, rendendo il processo computazionalmente proibitivo per dataset di grandi dimensioni.

Sono state implementate tre strategie di blocking:

- **Blocking per Industry**: suddivisione dei record in blocchi basati sull'attributo *industry*, riducendo i confronti tra aziende di settori diversi.
- **Blocking per Città della Sede Legale**: suddivisione dei record in blocchi basati sulla *headquarters_city*, limitando i confronti alle aziende con sede nella stessa città.
- **Sorted Neighbourhood Blocking**: ordinamento dei record per nome azienda, applicazione di una finestra scorrevole di ampiezza 5, confrontando solo i record all'interno della stessa finestra per gestire variazioni nei nomi aziendali (errori tipografici, abbreviazioni, ecc.).

2.2 Pairwise Matching

Dopo aver ridotto lo spazio di confronto con le tecniche di blocking, è stato necessario definire metriche di similarità per determinare le corrispondenze tra i record.

- **Misura di Similarità sui Nomi delle Aziende**: utilizzo dell'algoritmo **Jaro-Winkler** per il confronto dei nomi aziendali, che gestisce errori tipografici e trasposizioni di caratteri. La metrica assegna un punteggio tra 0 e 1, con una soglia di 0.90 per considerare due nomi come simili.
- **Confronto Esatto per Settore e Città**: confronto esatto per gli attributi *industry* e *headquarters_city*, aumentando la probabilità di corrispondenza quando entrambi gli attributi sono uguali.
- **Definizione delle Soglie di Matching**: se la similarità Jaro-Winkler è superiore a 0.90, il match è confermato. Se la similarità è tra 0.80 e 0.92, il confronto esatto su settore e città viene utilizzato per decidere se i record corrispondono.

2.3 Pipeline basata su Modello Transformer

Per migliorare ulteriormente il processo di entity matching, è stata implementata una seconda pipeline basata su un modello di deep learning, utilizzando un Transformer, in particolare un Distillated BERT, addestrato su un dataset di confronto tra nomi aziendali.

2.3.1 Addestramento del Modello

L'addestramento del modello Transformer è stato effettuato su un dataset esterno contenente coppie di nomi di aziende già etichettate come match (1) o non match (0). Questo dataset ha permesso al modello di apprendere rappresentazioni più complesse delle relazioni tra i nomi aziendali, superando le limitazioni dei semplici algoritmi di similarità testuale come Jaro-Winkler.

Il dataset di addestramento comprendeva una varietà di nomi aziendali, con differenze dovute a:

- Abbreviazioni (es. "International Business Machines" vs. "IBM").
- Errori tipografici e trasposizioni di caratteri.
- Differenze di formattazione (es. "Apple Inc." vs. "Apple").
- Nomi simili ma non corrispondenti (es. "Twitter" vs. "Twitter Careers").

2.3.2 Valutazione delle Prestazioni

Dopo l'addestramento, il modello è stato applicato al dataset target, generando una lista di coppie predette come match (1) o non match (0). I risultati sono stati confrontati con la ground truth per calcolare le metriche di performance.

Risultati della Pipeline Basata sulle coppie generate da recordLinkage

- **Numero di coppie in pairwise**: 98,726
- **Numero di coppie in ground truth**: 159
- **Numero di coppie in ground truth con label a 1**: 80
- **Numero di coppie in intersezione**: 145
- **Numero di coppie nell'intersezione con label 1**: 73
- **Precision**: 0.5034
- **Recall**: 0.9125
- **F1-Score**: 0.6489

Risultati della Pipeline Basata su Transformer

- **Numero di coppie predette come match dal modello**: 1,484
- **Numero di coppie in ground truth**: 159
- **Numero di coppie in ground truth con label a 1**: 80
- **Numero di coppie in intersezione**: 14
- **Numero di coppie corrette (True Positives)**: 11
- **Precision**: 0.7857
- **Recall**: 0.1375
- **F1-Score**: 0.2340

2.3.3 Analisi Comparativa

I risultati mostrano una netta differenza tra le due pipeline:

- La pipeline basata su Jaro-Winkler ha ottenuto un recall molto alto (91.25%), garantendo un'ampia copertura dei match corretti, ma con una precisione relativamente bassa (50.34%), indicando la presenza di numerosi falsi positivi.
- La pipeline basata sul Transformer ha migliorato significativamente la precisione (78.57%), riducendo i falsi positivi, ma ha avuto un recall molto basso (13.75%), indicando che molte coppie corrette non sono state rilevate.

Questa differenza suggerisce che:

- La pipeline basata su regole e Jaro-Winkler è più conservativa e garantisce un'elevata copertura, ma può includere falsi positivi.
- Il modello Transformer viene applicato alle coppie candidate generate dalla pipeline tradizionale e viene utilizzato per classificare ogni coppia come match (1) o non match (0), determinando un filtraggio basato su apprendimento supervisionato anziché su soglie fisse.

2.4 Conclusioni

L'analisi comparativa delle due pipeline dimostra che entrambe le tecniche hanno punti di forza e debolezze:

- Se l'obiettivo è massimizzare il recall (evitare di perdere possibili match), la pipeline basata su Jaro-Winkler è più efficace.
- Se l'obiettivo è ridurre i falsi positivi e ottenere match più affidabili, la pipeline basata su Transformer è più promettente, ma richiede un addestramento più mirato per migliorare il recall.