HomeWork 3
Ingegneria dei Dati

Magneto: Combining Small and Large Language Models for Schema Matching

Presentazione a cura di:
Alessandro di Mario,
Pietro Barone,
Guido Petrolito,
Douglas Ruffini.

Il problema (Schema Matching)

Cos'è lo schema matching: Trovare quali colonne di due dataset diversi rappresentano lo stesso concetto.

Lo schema matching rappresenta uno dei problemi fondamentali nell'ambito dell'integrazione dei dati. Ogni volta che si desidera combinare informazioni provenienti da sorgenti eterogenee, diventa necessario stabilire quali attributi di una tabella corrispondano semanticamente agli attributi di un'altra. Specialmente quando i dati provengono da contesti reali, rumorosi e semanticamente complessi

Problema: i nomi sono diversi, i valori diversi, serve comprensione semantica.

Perché è difficile nei dati realiCaso reale dall'articolo (dominio biomedico):10 dataset700+ attributi standard (GDC)Colonne con nomi ambiguiNecessaria conoscenza di dominioMetodi classici (stringhe, regole, grafi) → falliscono.

L'articolo "Magneto: Combining Small and Large Language Models for Schema Matching" affronta questo problema introducendo un nuovo framework che sfrutta in modo complementare le capacità dei Small Language Models (SLM) e dei Large Language Models (LLM).

Perché usare i Language Models

Gli LLM/SLM rappresentano il testo come vettori semantici (embedding).
Colonne simili → vettori vicini → cosine similarity alta.
Ma:

SLM	LLM
Veloci	Lenti/costosi
Serve training	Zero-shot
Poco contesto	Forte semantica

Gli SLM, pur essendo efficienti, richiedono dati di addestramento per essere efficaci; gli LLM, invece, pur avendo una conoscenza semantica molto ampia, sono limitati da costi computazionali, latenza e vincoli di finestra di contesto.

Idea chiave di Magneto

Combinare SLM e LLM in una pipeline a due fasi
SLM → Candidate Retrieval (veloce, filtra)
LLM → Reranking (preciso, semantico)

LLM usati solo dove servono.

Magneto propone quindi una pipeline a due fasi — retrieval e reranking — in cui gli SLM vengono utilizzati per filtrare un insieme ristretto di candidati e gli LLM per riclassificare tali candidati con maggiore precisione semantica. A questo si aggiunge un elemento innovativo: l'utilizzo degli LLM per generare automaticamente i dati di training necessari al fine-tuning degli SLM, eliminando la necessità di dataset annotati manualmente.

Fase 1: Candidate Retriever (SLM)

Gli SLM vengono utilizzati per calcolare embedding di colonna e produrre, per ogni colonna sorgente, una lista classificata di possibili colonne target.

Ogni colonna → serializzata (nome + valori)
Convertita in embedding
Confronto tramite cosine similarity
Produce Top-k candidati

Riduce drasticamente lo spazio di ricerca.

Serializzazione e campionamento

Un aspetto cruciale è come trasformare una colonna tabellare in una sequenza di token interpretabile dal modello.

Magneto introduce diverse strategie di serializzazione:

- S default: nome colonna + valori separati da [SEP]
- S verbose: aggiunta di prefissi descrittivi
- S repeat: ripetizione del nome colonna per enfatizzarlo

Viene inoltre introdotto un campionamento prioritario dei valori basato su frequenza e hashing, per selezionare valori rappresentativi riducendo il rumore.

Tecniche:
serializzazione intelligente
campionamento valori rappresentativi
riduzione rumore

Fase 2: Reranking (LLM)

LLM riceve solo i Top-k candidati:
Prompt:“Quanto sono semanticamente simili queste due colonne? Score 0–1”

Gli LLM intervengono solo sui migliori k candidati per ogni colonna, assegnando punteggi di similarità tramite prompt progettati ad hoc.
Questo consente di sfruttare la conoscenza semantica dell’LLM minimizzando il numero di chiamate API e rispettando i limiti di contesto.

Few-shot learning.

LLM fa confronti semantici profondi impossibili per SLM.

Addestrare gli SLM senza dati annotati

Questa è una delle innovazioni più importanti.
Gli LLM vengono utilizzati per generare colonne sinteticamente diverse ma semanticamente equivalenti, che fungono da esempi positivi per un processo di apprendimento contrastivo autosupervisionato.

Il training utilizza:

- Triplet Loss
- Online Triplet Mining
- Classi di colonne derivate dalla stessa ancora

Questo permette di costruire embedding altamente discriminativi senza dati annotati manualmente.

Verifica sperimentale:

Il nuovo benchmark rappresenta uno scenario reale estremamente complesso:

- 10 dataset reali biomedici
- fino a 179 colonne sorgente
- 736 colonne target
- alta variabilità sintattica e semantica

A differenza dei benchmark esistenti, questo non è “saturato” e mette realmente alla prova gli algoritmi.

Magneto supera tutte le baseline tradizionali e recenti (COMA, SimFlooding, ISResMat, Unicorn) sia su GDC che su Valentine.
La configurazione migliore risulta essere Magneto-ft-llm.
Anche le versioni zero-shot mostrano prestazioni superiori grazie alle tecniche di serializzazione e campionamento.

Compromesso tra accuratezza e costo

Uno dei contributi chiave dell'articolo è dimostrare che è possibile ottenere alta accuratezza senza costi proibitivi, usando gli LLM solo dove necessario.
Magneto riesce a posizionarsi in una zona ottimale del trade-off tra runtime e accuratezza.

Magneto rappresenta un passo avanti significativo nello schema matching,

dimostrando che la combinazione intelligente di SLM e LLM può superare i limiti di entrambi.

L'uso degli LLM per generare dati di training, la pipeline a due fasi, le tecniche di serializzazione e il nuovo benchmark rendono questo lavoro particolarmente rilevante per applicazioni reali di integrazione dei dati, specialmente in domini complessi come quello biomedico.

Gli LLM non sostituiscono gli algoritmi tradizionali. Li potenziano quando usati nel punto giusto della pipeline. Questa è l'idea centrale dell'articolo.