Ingegneria dei Dati:
Knowledge Extraction

Mattia Micaloni, Carlo Proserpio, Alessandro Di Girolamo

Project repository: https://github.com/Xhst/de-projects/tree/main/project-4

1. Introduzione

La presente relazione ha l'obiettivo di illustrare la soluzione adottata per l'estrazione di informazioni scientifiche direttamente dalle tabelle presenti negli articoli reperiti su *arXiv*[1], definite come *claim*. Un claim è un insieme di informazioni di una tabella, riguardanti una specifica metrica (o risultato) della tabella. Di solito quindi, un'intera tabella è definita da più claim. Le informazioni contenute in un claim - escludendo la metrica - sono chiamate *specifiche* e sono definite come una coppia `(nome, valore)`. In certi casi, se non ci sono metriche nella tabella, estraiamo solo le specifiche. Il formato di claim è il seguente (il carattere "|" funge da separatore):

```
|{|name1, value1|, ... , |nameN, valueN|}, Metric, Outcome|

Esempio:

"|{|Model Type, General LLM|, |Model Name, ChatGPT-3.5-turbo|, |Parameter
   Size, 175B|, |Dataset, Spider dev|, |Difficulty Level, 1|}, Execution
   Match , 0.760|"
```

Tale approccio è stato concepito con l'intento di ottenere risultati accurati e significativi, in grado di rappresentare in modo preciso le informazioni e le soluzioni proposte nei diversi paper analizzati. Successivamente, la soluzione proposta è stata oggetto di una valutazione approfondita, al fine di verificarne la validità generale, mediante l'impiego di metriche consolidate.

2. Estrazione dei claim

In questa sezione vengono illustrati i passaggi chiave per l'estrazione dei claim dai file *JSON* estratti nel Progetto 1[2].

[1] https://ar5iv.org/
[2] https://github.com/Xhst/de-projects/tree/main/project-1

2.1 Pulizia e formato tabella

In primo luogo, le tabelle contenute nei vari file sono state pulite utilizzando la libreria Python *BeautifulSoup*, con l'obiettivo di rimuovere tag e attributi superflui che non avrebbero avuto alcun impatto sulla struttura della tabella né sulle informazioni di interesse. Successivamente, le tabelle sono state estratte attraverso la libreria *Pandas* e sono state infine convertite in formato testuale idoneo a facilitare il prompting al modello di linguaggio.

2.2 Modello di linguaggio

Il nucleo della soluzione risiede nell'adozione di un modello di linguaggio avanzato. Questa scelta è stata resa possibile grazie all'utilizzo di Groq, una piattaforma che offre un'interfaccia per interrogare diversi modelli di linguaggio.
Più precisamente, è stato impiegato il modello *llama-3.3-70b-versatile* (con temperatura impostata a 0), al fine di ottenere risposte "standardizzate" riducendo la variabilità nelle risposte stesse, mantenendo una maggiore coerenza.
Al modello sono state fornite informazioni generali tramite un prompt di sistema, con l'obiettivo di "immergerlo" nel contesto specifico di interesse. Successivamente, sono stati presentati alcuni esempi di tabelle, complete di descrizioni e riferimenti, insieme alle risposte (ovvero i *claim*) che il modello avrebbe dovuto restituire. Questo processo ha contribuito ad affinare la capacità predittiva del modello per il compito in oggetto.
Una volta completato questo processo di *prompt engineering*, al modello viene passata direttamente la tabella elaborata, insieme alla sua descrizione e i riferimenti pertinenti. Il modello, in risposta, restituisce i claim nel formato stabilito nella sezione 1.

2.3 Creazione claim

Successivamente, la risposta del modello, espressa in formato testuale, è stata ulteriormente elaborata attraverso un apposito *script*. Questo ha permesso la generazione di file *JSON*, ciascuno contenente i *claim* estratti da una specifica tabella, strutturati secondo il seguente formato:

```
[
"0": {
  "specifications": {
    "0": {"name": "Model type", "value": "General LLM"},
    "1": {..}
  },
  "Measure": "Execution Match",
  "Outcome": "0.760"
},
"1": {..},
]
```

2.4 Articoli e tabelle analizzate

In totale, sono stati selezionati 10 articoli, ognuno dei quali conteneva mediamente 3 o 4 tabelle. Complessivamente, sono state elaborate 31 tabelle, tutte utilizzate per la valutazione della soluzione descritta fino a questo punto.

3. Profiling

Il progetto svolto ha anche incluso una parte dedicata all'estrazione di statistiche dai claim estratti. In particolare, sono stati generati file in formato *.csv*, ognuno contenente una colonna denominata *key*, che rappresenta i valori in analisi, e una colonna *count*, che indica il numero di occorrenze di ciascun valore. Questo approccio è stato adottato per identificare e analizzare la distribuzione dei diversi termini presenti nei dati, come i nomi e i valori delle specifiche nei claim e le metriche.

4. Allineamento

L'allineamento consiste nel raggruppare le specifiche (nomi e valori) e le metriche che, pur rappresentando la stessa entità, sono sintatticamente diverse.
Per svolgere questo compito abbiamo prima realizzato gli embedding di nomi e valori delle specifiche e delle metriche insieme all'outcome per poi fare un clustering usando HDBSCAN. Questo passaggio ci ha permesso di raggruppare le entità che molto probabilmente sono la stessa. Abbiamo poi manualmente revisionato i cluster e spostato gli elementi posizionati erroneamente. Ad ogni cluster pulito abbiamo assegnato un nome utilizzato per rappresentare l'entità.
Al termine dell'allineamento è stato anche rieseguito il profiling utilizzando i nuovi valori allineati. Il tempo manuale richiesto per circa 30 metriche, 30 nomi e 300 valori è stato di circa 30 minuti.

5. Valutazione estrazione

Per valutare l'estrazione è stato necessario produrre un *Ground Truth* e scegliere delle metriche opportune.

5.1 Ground Truth

Per la costruzione della *Ground Truth*, si è scelto di considerare tutti i paper estratti, includendo quindi tutte le tabelle analizzate. Tale decisione è stata motivata dal fatto che il numero complessivo di tabelle, pari a 31, risulta gestibile manualmente entro tempi di lavoro accettabili.
L'approccio adottato per generare i claim della *Ground Truth* è consistito nel salvare le

risposte fornite dal modello e successivamente modificarle, ove necessario, attraverso un confronto diretto con il contenuto del paper di riferimento. Questo processo ha permesso di correggere o approfondire i risultati generati automaticamente.
Al termine di questa procedura, è stato generato un nuovo corpus di file JSON, composto dai *claim* ideali, i quali rappresentano una versione accurata e manualmente validata delle informazioni estratte. Il tempo impiegato per la creazione del *Ground Truth* è stato di circa 5 ore.

5.2 Metriche di valutazione

Per la valutazione sono state utilizzate le seguenti metriche: *Precision*, *Recall*,*F-measure* (o F1 score).

- **Precision**: Valuta l'accuratezza dei positivi predetti.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

- **Recall**: Misura la proporzione di esempi positivi correttamente identificati rispetto al totale degli esempi positivi.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

- **F1**: Media armonica di *Precision* e *recall*, la quale fornisce una misura unica per bilanciare i due.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.3 Risultati

Per il calcolo dei risultati sono stati considerati due approcci distinti. Il primo approccio, denominato ***Claims*** *evaluation*, prevede una valutazione complessiva dei claim: il claim viene considerato errato se anche una sola delle specifiche risulta differente rispetto a quelle presenti nel *Ground Truth* (o se manca una specifica, ad esempio). Il secondo approccio, chiamato ***Claims Parts*** *evaluation*, si concentra sulla valutazione delle singole specifiche e metriche all'interno di ciascun claim, contando quante di esse sono corrette (ovvero presenti nel *Ground Truth*). Questo approccio consente una valutazione più flessibile rispetto al primo. É da notare come i risultati mostrati nella tabella 1 sono ottenuti tramite un match esatto dei claim o delle parti dei claim con quelli del *Ground Truth*, quindi con un metodo di valutazione meno rigido sicuramente le metriche sarebbero state più alte.

	Claims	Claims Part
Precision	0.53	0.74
Recall	0.52	0.73
F1	0.52	0.73

Tabella 1: Precison, Recall e F1 per i due differenti metodi di valutazione.

6. Considerazioni

La soluzione sviluppata per il presente progetto ha evidenziato come i modelli di linguaggio, come quello utilizzato, possano rappresentare uno strumento promettente per l'estrazione di dati e per la successiva manipolazione degli stessi, come dimostrato dai risultati ottenuti finora.

Un possibile sviluppo futuro potrebbe prevedere l'impiego di modelli più potenti, possibilmente addestrati su dataset che rispecchiano da vicino il task in oggetto. Inoltre, sarebbe utile un ulteriore *fine-tuning* incentrato sulla struttura e sulla forma dei documenti utilizzati nel progetto, al fine di migliorare le prestazioni predittive del modello.
Un tale miglioramento consentirebbe al modello di fornire informazioni, (come i *claim* in questo caso), più precise e complete, ottenendo così risultati migliori nelle valutazioni effettuate.