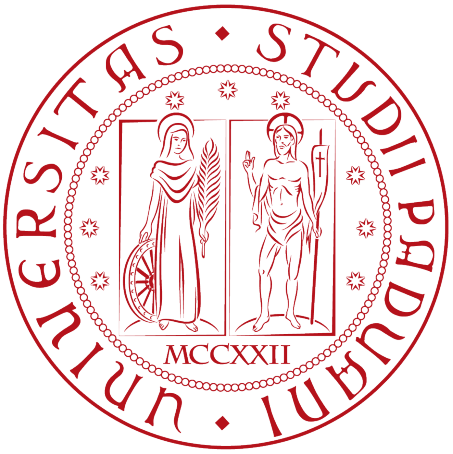
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Sistema di estrazione di testo da documenti per RALM

Tesi di laurea

Relatore

Prof. Alessandro Sperduti

Laureando

Andrea Auletta

Anno Accademico 2022-2023

Coloro che possono immaginare qualsiasi cosa, possono creare l'impossibile

— Alan Turing

Dedicato ai miei genitori e a tutti coloro che continuano a esserci per me.

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Andrea Auletta presso l'azienda Azienda Siav S.p.A. Lo stage è stato collocato in un progetto più ampio che riguarda la progettazione e lo sviluppo di applicativi per l'interazione tramite linguaggio naturale tra utenti e Retrieval Augmented Language Model (RALM). L'obbietivo di questo stage è stato quello di migliorare un prototipo aziendale in maniera tale che potesse dare delle risposte di miglior qualità. Questo è stato possibile farlo studiando alcuni casi possibili di documenti che potrebbero essere messi a disposizione del RALM, convertendo vari elementi semantici (come le tabelle) in testo non strutturato e migliorando la qualità del chunking.

“ Il futuro è in mano ai deboli che si sono fatti coraggio, per farsi coraggio bisogna sapersi guardare dentro”

— Mario Molinari

Ringraziamenti

Innanzitutto vorrei ringraziare il Prof. Alessandro Sperduti, relatore della mia tesi e il mio tutor aziendale Gioele Perin

Desidero ringraziare i miei genitori per essermi stati vicino durante questi anni.

Desidero ringraziare tutti i miei amici per i bei momenti e le esperienze che abbiamo passato assieme e che ci hanno fatto crescere

Vorrei ringraziare anche la mia maestra di canto, Ilaria, grazie alla quale sento di aver avuto una crescita personale molto forte

Padova, Luglio 2023 Andrea Auletta

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

In questo capito viene introdotto il problema per il quale è stato affrontato questo percorso di stage. Viene descritto brevemente come è stato risolto seguendo i diversi vincoli imposti dall'azienda presso la quale ho sviluppato il progetto. Viene presentata l'azienda e viene descritta la struttura del documento.

1.1 Breve analisi del problema

C'è sempre più bisogno di ottenere informazioni da grandi quantità di documenti in una maniera rapida e semplice. Il progetto sviluppato in questo percorso di stage è stato incentrato proprio su questa problematica: andando avanti nel tempo si accumulano moltissimi documenti e dover cercare una singola informazione all'interno delle volte può risultare scomodo. Uno degli approcci più diretti per una persona per cercare un informazione qualsiasi, nella quotidianità, è quella di porre domande ad altre persone. L'azienda presso quale ho svolto lo stage, per far sì che questo problema possa essere risolto tramite quest'ultimo approccio, ha deciso di sfruttare la potenza dei RALM.

1.1.1 RALM

I Large Language Model (LLM) sono dei modelli di apprendimento automatico in grado di generare testi coerenti e informativi. Hanno rivoluzionato il campo dei Natural Language Processing (NLP) e vengono impiegati in diverse applicazioni, un esempio possono essere i chatbot. Questi ultimi sfruttano la capacità dei LLM di interagire tramite il linguaggio naturale con gli utenti. I Retrieval Augmented Language Model (RALM) non sono altro che dei LLM, basati sul *Question answering*[g], che permettono di utilizzare una fonte esterna di conoscenza (come ad esempio una collezione di documenti) per fornire informazioni aggiuntive al modello durante la generazione di testo. Un documento viene suddiviso in parti più piccole chiamate chunk. Il modello effettua una query alla fonte esterna usando l'input dell'utente come chiave di ricerca e riceve una lista formata da un numero determinato di chunk rilevanti (possono avere la risposta al quesito posto). Il numero determinato di chunk viene dato dalla quantità di *Token*[g](elemento individuale all'interno di un testo: parola, parte di parola, punteggiatura) che il modello può ricevere in ingersso. Questi chunk vengono poi utilizzati come contesto aggiuntivo e vengono combinati con l'input originale per

produrre il testo finale. Questo approccio permette di generare testi più informativi, accurati e diversificati sfruttando la conoscenza dovuta alla presenza della fonte esterna.

1.1.2 Il problema

Per poter garantire la qualità delle risposte generate dal RALM è necessario che i documenti a disposizione siano in formato testuale e che il loro contenuto abbia tutta l'informazione neccessaria. I documenti sono disponibili in diversi formati e spesso non sono costituiti semplicemente di testo non strutturato, ma presentano frequentemente vari "elementi semantici" come tabelle, immagini e titoli. Presentano quindi informazione che non era immediatamente estraibile e convertibile in testo (tramite i tool di estrazione del contenuto) utile ai fini dell'interazione col RALM.
Un altro problema che si può presentare è che i chunk rilevanti forniti al RALM non sempre contengono la risposta alla domanda posta, quindi, effettivamente, non sempre sono veramente rilevanti. L'ultimo problema affrontato è che spesso i documenti presentano elementi ripetuti come, ad esempio, una serie di punti presente in un indice (come mostrato in figura 1.1). Queste ripetizioni occuperebbero spazio inutilmente all'interno dei chunk aumentando così il numero di questi ultimi.

Figura 1.1: Esempio di elementi ripetuti da minimizzare nella conversione del contenuto.

1.1.3 Il progetto

Gli elementi sui quali si è lavorato di più in questo stage per migliorare la qualità delle risposte fornite dal RALM sono state le tabelle e i titoli. È stato applicato un metodo di conversione delle tabelle in modo tale da poterle rendere comprensibili al RALM e che potessero comunque mantenere il senso della loro struttura anche se sottoforma di testo non strutturato.
Per migliorare la probabilità di trovare chunk rilevanti vengono aggiunti i titoli ai chunk (ove possibile) in modo tale da potergli dare un senso logico di posizione all'interno del documento.
Le ripetizioni di caratteri vengono semplicemente ridotte a un singolo carattere.

1.2 L'azienda: Siav S.p.A.

Figura 1.2: Logo dell'azienda Siav S.p.A.

Siav S.p.A. è un'azienda informatica specializzata nella dematerializzazione, nella gestione elettronica dei documenti e nei processi digitali. Fondata nel 1990 a Rubano (Padova) è oggi la prima azienda italiana nel settore dell'Enterprise Content Management, e offre software, soluzioni in cloud e servizi di outsourcing per la Gestione Elettronica dei Documenti, il Protocollo Informatico, il Workflow Management, la Fatturazione Elettronica e la Conservazione Digitale (ricavato da *Siav S.p.A.*. URL: https://www.siav.com/it/).

1.3 Organizzazione del testo

Il secondo capitolo decrive nel dettaglio le problematiche affrontate e mostra le varie soluzioni applicate per poter rendere migliore la qualità delle risposte del RALM;

Il terzo capitolo approfondisce le tecnologie utilizzate nel progetto, la progettazione e la codifica di quanto sviluppato giustificando ogni scelta effettuata;

Il quarto capitolo illustra i vari test effettuati sul RALM e mostra i risultati ottenuti ogni modifica attuata;

Nel quinto capitolo vengono scritte le conclusioni riguardo al progetto svolto.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g];

Capitolo 2

Studio e ricerca preliminare

In questo capitolo vengono descritte nel dettaglio tutte le problematiche da risolvere durante lo sviluppo del progetto. Verranno illustrate nel dettaglio le varie metodologie individuate per poter rendere di maggiore qualità le risposte fornite dal RALM tramite il raggiungimento dei diversi obiettivi.

2.1 Analisi dei requisiti

Requisiti

Una prima versione di backend dei servizi di estrazione di testo era già disponibile e lo scopo di questo progetto era quello di migliorare proprio quest'ultimo raggiungendo i seguenti obbiettivi:

- Obbligatori:

  Parsing[g]ad-hoc per documenti in cui le componenti grafiche contribuiscono alla semantica (es. tabelle e immagini);

  Estrapolazione, per alcuni formati dove sia possibile, della struttura logica del documento (es. individuando titoli e paragrafi);

  Pulizia del testo estratto dai documenti (es. eliminazione delle componenti inutili come gli indici e i sommari).

- Desiderabili:

  Interpretazione delle immagini allo scopo di arricchire i Chunk[g]in cui sono contestualizzate.

I formati dei file considerati in questo stage sono stati i seguenti:

- Pdf;
- Docx;
- HTML.

2.1.1 Problematiche

Non sempre il RALM è in grado di fornire una risposta corretta o soddisfacente. Diverse sono state le criticità da risolvere durante lo sviluppo sia per quanto riguarda il rendere più comprensibile il contenuto al RALM, sia per quanto riguarda le conversioni dei documenti nei vari formati. Tika, il tool utilizzato per estrarre il contenuto in testo non strutturato, durante la conversione ha delle perdite sulla struttura del contenuto come ad esempio le perdite di informazioni sulla struttura delle tabelle oppure sulla struttura del documento in sè (titolo, paragrafi, sottoparagrafi). Per agevolare la comprensione del RALM, velocizzare le operazioni di ricerca dell'informazione e fare in modo di aumentare la probabilità che le risposte siano corrette è necessario trovare un modo per mantenere in maniera efficace queste strutture all'interno del testo semplice. Un vantaggio di Tika è che può anche convertire il contenuto in formati strutturati come l'XHTML. Grazie a questa funzione sarebbe già possibile individuare titoli, paragrafi e tabelle del documento ma, purtroppo, non è possibile farlo in tutti i formati: per i file HTML e Docx, dove c'è già una struttura di base, Tika riesce a identificare i vari elementi semantici, mentre con i Pdf la questione risulta più complessa. Inoltre, non sempre, Tika riesce a ottenere la struttura esatta della tabella rispetto a come è rappresentata. L'ultimo punto da analizzare è la questione delle ripetizioni di alcuni caratteri come segni di punteggiatura, newline e spazi: chiaramente Tika non è in grado di riconoscere quando ci sono degli elementi superflui all'interno del contenuto, è necessario quindi capire come fare per ridurre al minimo la quantità di caratteri all'interno del testo.

2.2 Ricerca e studio preliminare

2.2.1 Table Question Answering

Nel Table Question Answering (TQA) le domande poste dall'utente cercano di avere una risposta precisa con i dati ricavati da delle tabelle. L'obiettivo è quello di migliorare l'accesso e la comprensione delle informazioni strutturate contenute nelle tabelle.

Linearizzazione delle tabelle

Una tabella può assumere moltissime strutture e per questo è stato deciso di considerare solamente le tabelle che avessero come prima riga un'intesazione orizzontale e nelle righe successive i vari dati (esempio: tabella 2.1).

Cibo	Quantità	Energia(KCal)	Carboidrati(g)	Proteine(g)
Pennette rigate	100g	359	71	13
Latte	100ml	47	4,9	3,2
Banana	100g	89	23	1,1

Tabella 2.1: Esempio di tabella presa in considerazione (valori approssimativi).

Come specificato precedentemente le tabelle al momento dell'estrazione venivano linearizzate in testo semplice perdendo alcune informazioni necessarie per la lettura effettuata dal RALM.
Per esempio la tabella 2.1 verrebbe linearizzata in questa maniera:

Cibo Quantità Energia(KCal) Carboidrati(g) Proteine(g) Pennette rigate 100g 359 71 13 Latte 100ml 47 4,9 3,2 Banana 100g 89 23 1,1

La tabella linearizzata perde quindi le informazioni sulla struttura e come risultato abbiamo una serie di valori posti senza avere troppo senso in fase di lettura per un RALM.

L'idea

Dopo un attenta ricerca effettuata su vari documenti scientifici sono riuscito ad individuare un modo semplice ed efficace per mantenere l'informazione nella tabella linearizzata e la sua struttura:

- All'inizio di ogni riga viene scritto "Riga n->" dove n sta per il numero della riga;
- Per ogni cella presente nella tabella vengono concatenati il valore dell'intestazione della colonna dov'è presente il valore e il valore della cella separati dal carattere ":";
- Ogni cella viene poi separata dall'altra con il carattere "|".
- A chiudere la riga viene inserito il carattere di escape per newline.

Quindi la tabella 2.1 viene linearizzata in questo modo:

Riga0->Cibo: Pennette rigate|Quantità:100g|Energia(KCal):359|Carboidrati(g):71|Proteine(g):13|
Riga1->Cibo: Latte|Quantità:100ml|Energia(KCal):47|Carboidrati(g):4,9|Proteine(g):3,2|
Riga2->Cibo: Banana|Quantità:100g|Energia(KCal):89|Carboidrati(g):23|Proteine(g):1,1|

Nel metodo scritto sul seguente articolo scientifico "Alon Talmor et al. «Multimodalqa: Complex question answering over text, tables and images». In: *arXiv preprint arXiv:2104.06039* (2021) (paragrafo 4 *Models*, sottosezione *Table QA Module*)" utilizza come sepratori di cella il carattare ';' e come speratore dal numero di riga alla prima cella il carattere ':'. I caratteri ':' e ';' potrebbero comparire con molta più frequenza all'interno dei valori delle celle e in questo modo si avrebbe un doppio significato per questi ultimi, cioè sarebbero sia un separatore che una parte di dato, il RALM quindi potrebbe avere dei problemi nel riconoscere questa differenza.

2.2.2 Chunking

Procedimento Il *chunking* è il procedimento mediante il quale in contenuto testuale di un documento viene suddiviso in parti più piccole chiamate chunk. Per poter fornire una risposta ben strutturata viene utilizzato un *Chat-Completion Model*[g] che riceve in ingresso una richiesta e tramite quest'ultima è in grado di costruire una risposta ben strutturata. La richiesta è formata dalla domanda dell'utente seguita dai chunk a cui il motore di ricerca ha dato lo score più alto. Il modello in ingresso può prendere un

numero limitato di *token* quindi non gli si può dare l'intero contenuto del documento, ha bisogno di questa suddivisione.
Per poter capire al meglio il significato delle parole presenti in un determinato contesto è necessario dover spezzare i chunk in *token*. Il *token* indica una singola unità linguistica o comunque un elemento individuale all'interno di un testo e può rappresentare per esempio una parola, un simbolo di punteggiatura o anche una parte di una parola. Qui di seguito viene spiegato come il motore di ricerca è in grado di assegnare lo score ai vari chunk e quindi come si riesce a capire quali sono i chunk che come contenuto avranno con maggior probabilità la risposta che si sta cercando.

Recupero delle informazioni

Okapi Best Matching 25 (BM25) BM25 è una ranking function usata dai motori di ricerca, è un algoritmo di tipo *Bag-of-Words*[g]e calcola un punteggio per ogni chunk presente in base alla frequenza di vari termini presenti nella query di ricerca (informazioni ricavate da *Wikipedia - Okapi BM25*. URL: https://en.wikipedia.org/wiki/Okapi_BM25).

Vector search Tramite la vector search vengono generate rappresentazioni vettoriali dei dati ed è possibile calcolare la similarità tra i vettori. Per poter riconoscere il vero significato attribuito ad una parola i chunk vengono suddivisi a loro volta in token. Grazie a questi token è possibile calcolare la distanza tra i vettori tramite. Il tipo di distanza utilizzata in questo progetto è stata la *Cosine Distance*.

Cosine similarity La *Cosine similarity* misura l'angolo tra due vettori in uno spazio multidimensionale (con l'idea che due vettori simili puntino in direzioni simili). La cosine similarity e la cosine distance hanno una relazione inversa: all'aumentare della distanza la similarità diminuisce e viceversa.
Dati due vettori A,B la *cosine similarity* viene calcolata come segue:

$$\text{Cos(A,B)} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

$$\text{Cosine distance} = 1 - Cos(A, B)$$

(Informazioni ricavate da *Weaviate - What are Distance Metrics in Vector Search?* URL: https://weaviate.io/blog/distance-metrics-in-vector-search).

Ricerca ibrida Il tipo di ricerca applicata in Weaviate per questo progetto è stata la ricerca ibridia che sfrutta sia BM25 che la *Vector search* per poter stabilire uno score per i chunk.

Reciprocal Rank Fusion (RRF) L'RRF score è il calcolo attreverso il quale riusciamo ad avere uno score unico per la ricerca ibrida. Ad ogni documento viene assegnato un punteggio che equivale alla somma dei reciproci dei suoi piazzamenti nelle varie ranked list ottenute tramite gli altri algoritmi utilizzati nella ricerca.

$$\text{RFF} = \sum_{d \in D} \left(\frac{1}{k + r(d)}\right)$$

Per esempio, nella tabella 2.2, vengono dati tre chunk A, B, C e vengono classificati nella seguente maniera tramite i due ranking algorithm:

Posizione	BM25	Vector Search
1	A	B
2	B	C
3	C	A

Tabella 2.2: Esempio di ranking per i documenti A, B, C tramite BM25 e Vector Search.

Gli RRF score dei documenti A, B, C sono i seguenti:

Chunk	RRF score
A	$1/1 + 1/3 = 1.3$
B	$1/2 + 1/1 = 1.5$
C	$1/3 + 1/2 = 0.83$

Tabella 2.3: Esempio di calcolo del RRF score per i documenti A, B, C.

Nella tabella 2.3 abbiamo quindi che il miglior chunk da considerare è il chunk B seguito poi dal chunk A e dal C.

(Informazioni ricavate da *Weaviate - Hybrid Search Explained.* URL: https://weaviate.io/blog/hybrid-search-explained).

L'idea

Per migliorare la qualità del chunking sono state attuate due strategie:

- Il chunk viene strutturato come segue: all'inizio sarà presente una lista di titoli consevutivi in ordine gerarchico per individuare il contesto del chunk. In base alla grandezza del chunk viene definita una quantità di token che definisce la lunghezza della lista dei titoli. Se questa grandezza viene superata si scarta il titolo più alto in ordine gerarchico fino a quando i token effettivi saranno minori rispetto alla quantità prestabilita. La grandezza scelta per la lista dei titoli equivale ai due ottavi della grandezza del chunk mentre il resto viene lasciato per il contenuto. Il contenuto del paragrafo viene concatenato separando le tabelle dal testo:
  - Se viene trovata una tabella si cerca di inserire le intere righe all'interno del chunk in modo tale da non perdere informazioni sui dati;
  - Se viene trovato del testo normale viene utilizzata la sliding window per inserire parte del contenuto del chunk precedente all'inzio del nuovo chunk.

  Quando viene utilizzata la sliding window quindi si creerà l'*Overlap*[g]del contenuto. L'overlap è utile per due motivi:
  - Se ci sono n chunk consecutivi nella lista dei chunk migliori è possibile unirli in modo deterministico, inoltre anche il modello è in grando di riconoscere questa continuità;

- Senza di esso ci sarebbero delle frasi spezzate prive di significato all'interno dei chunk.

Come esempio un documento potrebbe avere la seguente struttura:

1 Titolo
testo
tabella
1.1 Sottotitolo
testo
1.1.1 Sottotitolo
testo
1.1.2 Sottotitolo
testo

Assumiamo che ogni titolo è composto da 1 token, ogni componente testuale è composta da 6 token e ogni riga della tabella da 5 token. Il chunk può essere composto al massimo da 8 token.

- Chunk1: 1 Titolo:testo (7 token)
- Chunk2: 1 Titolo: prima riga della tabella (6 token)
- Chunk3: 1 Titolo: seconda riga della tabella (6 token)
- Chunk4: 1 Titolo| 1.1 Sottotitolo: testo (8 token)
- Chunk5: 1.1 Sottotitolo | 1.1.1 Sottotitolo: testo (8 token -> Qui viene eliminato "1 Titolo" perchè sennò si sforerebbe la grandezza prestabilita)
- Chunk6: 1.1 Sottotitolo | 1.1.2 Sottotitolo: testo (8 token -> Come nel chunk precedente viene eliminato "1 Titolo")

- Pulizia del testo: le serie di caratteri (come spazi, newline, '-', '*', altro), scelti prima di effettuare la pulizia, vengono sostituiti con un unico carattere dello stesso tipo. Ogni carattere è composto da un singolo token e quindi i chunk nel caso in cui non si eliminassero queste ripetizioni avrebbero dei caratteri superflui che occuperebbero spazio inutilmente, ci sarebbe il rischio di creare più chunk del dovuto.

2.3 Pianificazione del lavoro

Le attività per sviluppare il lavoro e le ore previste per ognunga di esse sono riportate nella tabella successiva. Come si può notare dal diagramma di Gantt diverse attività si sovrappongono e questo accade perchè diverse di queste ultime hanno dei punti in comune.

2.3.1 Pianificazione delle attività

Numero attività	Attività	Ore previste
1	Studio introduttivo su Natural Language Processing e Large Language Model	16
2	Studio delle tecniche di estrazione di testo e dei principali tool nell'ambito dell'NLP	16
3	Studio dell'attuale implementazione del chatbot basato su retrueval-augmented LLM	16
4	Analisi dei requisiti con studio delle casistiche da gestire	30
5	Progettazione delle varie componenti richieste nel paragrafo	70
6	Implementazione del software	100
7	Test e sperimentazione del software	24
8	Documentazione	48

Tabella 2.4: Tabella di pianificazione delle attività.

2.3.2 Diagramma di Gantt delle attività

Viene mostrato il diagramma di Gantt delle attività svolte durante le nove settimane dello stage.

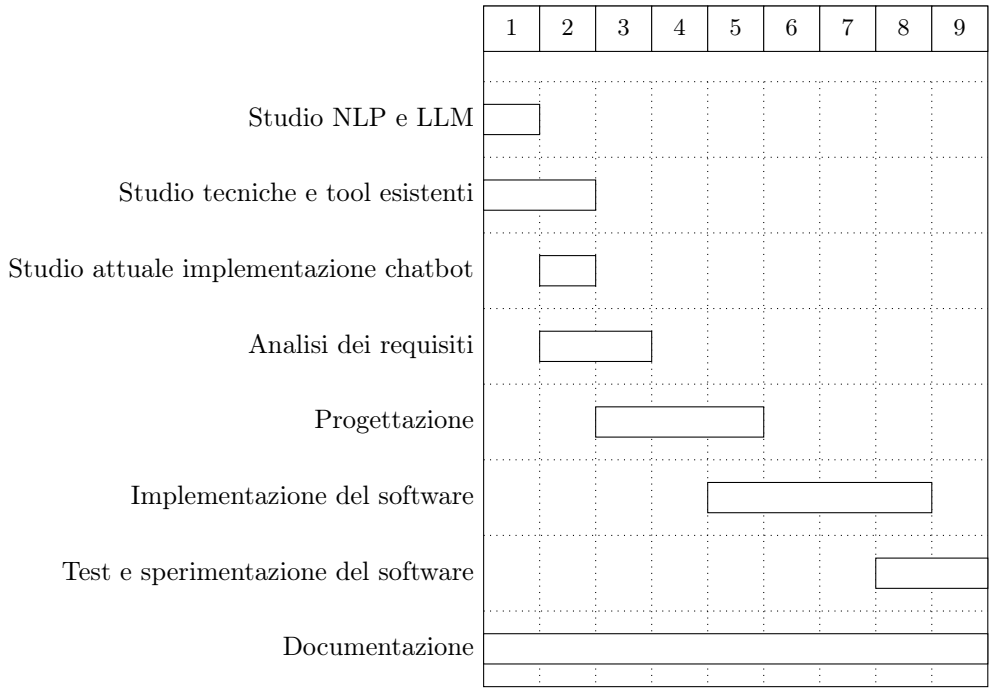


Figura 2.1: Diagramma di Gantt delle attività.

Capitolo 3

Progettazione e codifica

In questo capitolo vengono discusse le scelte progettuali effettuate per poter sviluppare al meglio il lavoro in questione

3.1 Tecnologie e strumenti

Il linguaggio di programmazione scelto per sviluppare il codice necessario a completare gli obiettivi richiesti è stato **Python** in quanto offre diverse librerie utili per l'apprendimento automatico e il trattamento del linguaggio naturale e utilizzandolo insieme a **Jupiter Notebook**, un'applicazione che permette di scrivere documenti misti con testo e codice eseguibile, è stato molto più facile realizzare e documentare le analisi sui dati.
Per poter estrarre il contenuto testuale dai vari documenti è stato scelto di continuare a utilizzare **Tika-python** che veniva già utilizzato nel primo prototipo fornito dall'azienda, ma per estrarre il testo non strutturato. Grazie a Tika, è possibile convertire il contenuto del testo in XHTML da diversi formati di file, questo è stato molto utile per poter individuare i vari elementi semantici all'interno dei documenti. Come scritto precedentemente, però, questa funzione di Tika non opera correttamente con tutti i tipi di formato e non estrae correttamente la struttura delle tabelle. Per ovviare al problema delle tabelle è stato scelto di utilizzare dei tool di estrazione appositi come **Pandas**, **pdfplumber** e **python-docx** rispettivamente per estrarre tabelle da file HTML, Pdf e Docx. Questi tool di estrazione appositi, essendo stati creati proprio per questa funzione, restituiscono le tabelle dei documenti con una struttura sicuramente più simile a quella presente nel documento rispetto a come estrarrebbe Tika. Per portare poi le tabelle ad un livello di astrazione che fosse uguale per tutti i tipi di formato, le tabelle sono state convertite in *DataFrame* di Pandas (struttura dati bidimensionale che può contenere dati di diversi tipi).

Per poter lavorare con l'XHTML fornito da Tika con più semplicità è stato utilizzato **BeautifulSoup**, una libreria Python che facilita l'estrazione e la manipolazione di dati da questo tipo di file, consente di creare e modificare tag all'interno del codice con il quale si sta lavorando e di estrarre il contenuto presente nei tag stessi.

Il Chat-Completion Model viene preso da **OpenAI API**, un'API fornita da OpenAI tramite la quale si possono sfruttare i vari modelli offerti per la generazione di testo in linguaggio naturale, mentre il motore di ricerca utilizzato è **Weaviate**, un database

vettoriale utile per la ricerca dei dati basata sulla loro semantica e sulle loro relazioni.

3.2 Progettazione e codifica

3.2.1 L'idea

Grazie a Tika riusciamo a convertire il contenuto dei documenti in XHTML in modo tale da poter lavorare con del testo strutturato, nei documenti dove è già presente una struttura al di sotto, Tika riesce a convertire correttamente il contenuto dei documenti in un XHTML ed è in grado di individuare le intestazioni (e i loro livelli gerarchici h1,h2,...) e tabelle e grazie a questo non è stato difficile riuscire a implementare le funzioni utili per lavorare con i file HTML e Docx. Per i Pdf, invece, le cose sono state più complesse, gli unici tool individuati che riescono a convertire bene il contenuto dei pdf (come "Aspose") in testo strutturato richiedono una licenza a pagamento e quindi è stato utilizzato comunque Tika. In questo caso il contenuto viene rappresentato tramite dei tag *div* che rappresentano le singole pagine e ogni riga viene trascritta tramite dei tag *p*.

Qui di seguito vengono descritte le parti di codice sviluppate e le motivazioni per le quali sono state prese determinate scelte.

3.2.2 L'architettura

La parte di codice sviluppata per questo progetto integra la parte di logica dell'applicazione di un backend già esistente.

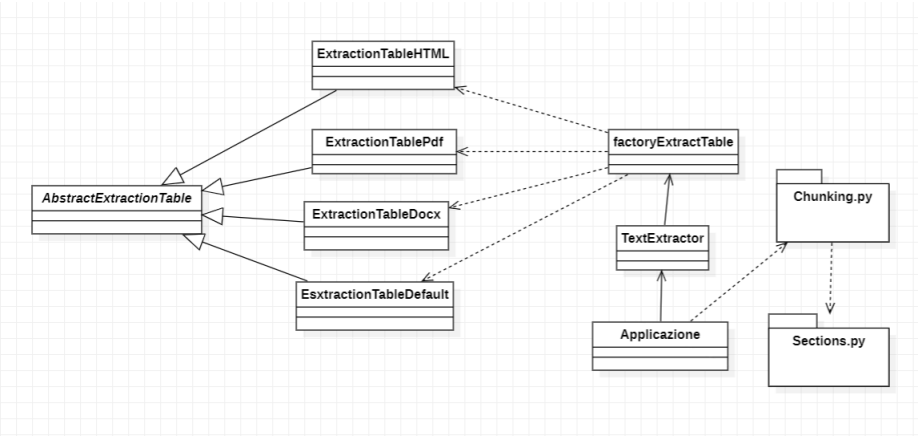


Figura 3.1: Architettura della parte di codice implementata nel backend.

L'algoritmo prevede come risultato la generazione dei chunk secondo i criteri specificati nel paragrafo 2.2.2, però, prima di poterli creare c'è bisogno di rielaborare i diversi elementi semantici presi in considerazione. La classe **TextExtractor** definisce una funzione **extractText** che prepara il contenuto del documento alla generazione dei chunk. Inizialmente estrae e converte il testo in XHTML grazie a Tika.

La parte successiva alla conversione è quella che riguarda l'estrazione delle tabelle dai documenti e la sostituzione di queste ultime linearizzate all'interno del testo che è una parte di algoritmo che varia per i diversi tipi di formato, per questo è

stato utilizzato il Design pattern[g] **Strategy**[g]: viene definita una classe astratta **AbstractExtractionTable** che prevede l'implementazione di alcune funzioni come **ExtractTable** e **replaceTab**, queste,estraggono la tabella e la sostituiscono con la versione linearizzata all'interno del documento a seconda del formato preso in considerazione (**ExtractionTableHTML, ExtractionTableDocx, ExtractionTablePdf**) utilizzando i tool appropriati per l'estrazione delle tabelle come Pandas, python-docx e pdfplumber. La funzione replaceTab, per i Pdf, funziona in maniera differente rispetto a quella sviluppata per HTML e Docx in quanto, per questi ultimi due formati, le tabelle vengono individuate da Tika, rappresentate dai tag *table* all'interno del testo strutturato e sostituite tramite BeautifulSoup. Invece, per i Pdf, non vengono utilizzati tag *table* quindi, sono state sfruttate le espressioni regolari per trovare e sostituire le tabelle all'interno del testo, queste espressioni regolari vengono create recuperando i valori delle celle prese dalle tabelle estratte precedentemente. Nel caso in cui il formato del file inserito non rientri fra quelli presi in considerazione o non sia ancora stata implementata una classe per quest'ultimo, viene utilizzata la classe **ExtractionTableDefault**: è stato deciso di implementare questa classe per avere comunque delle possibilità nella sostituizione delle tabelle nel caso in cui si riesca a convertire discretamente il documento in XHTML tramite Tika. Se vengono trovate poi delle tabelle all'interno del contenuto tramite BeautifulSoup, allora si potranno estrarre tramite Pandas e successivamente sostiure con la linearizzazione. Per instanziare l'oggetto del tipo ExtractionTable corretto viene utilizzata la funzione **factoryExtractTable**.

Prima di concludere con la funzione extractText bisogna considerare il caso in cui Tika non riesca a convertire correttamente il contenuto in XHTML come spiegato nel paragrafo 3.2.1. Nel caso in cui Tika non riesca a convertire correttamente, quindi non saranno presenti header, viene inserito un tag *h1* che contiene il titolo del file come primo figlio del tag *body*. Inoltre per semplificare il lavoro che dovrà essere attuato poi in fase di chunking, tutti i tag figli presenti nei *div* che rappresentano le pagine vengono estratti e resi figli del *body*, questo viene fatto in quanto, tramite BeautifulSoup, si passerà da un tag al suo sibling successivo piuttosto che al tag successivo (che potrebbe essere anche un figlio): al momento dell'estrazione del contenuto se viene estratto il contenuto di un tag *table* verrà estratto tutto il testo presente nella tabella, se passo al tag successivo (che potrebbe essere un *thead*) estrarrò di nuovo il contenuto del figlio avendo così delle informazioni duplicate.

Al fine di semplificare le operazioni da effettuare sul codice XHTML è stato deciso di creare un modulo Python **Section.py** che espone la funzione **makeSection**: questa funzione converte il codice in una struttura ad albero chiamata *Sezione*. Ogni oggetto Sezione è composto dal titolo del paragrafo (che viene ricavato dagli header presenti nell'XHTML) e dal suo contenuto. Il contenuto è una lista di elementi che può contenere testo o può contenere altre sezioni nel caso in cui ci siano dei sottoparagrafi.

Infine, grazie alla funzione **chunking** presente nel modulo **Chunking.py** viene generata la lista di chunk del documento. Ogni oggetto Chunk è composto da:

- Titolo: titolo del paragrafo al quale corrisponde il suo contenuto;
- parentsTit (lista di stringhe): contiene tutti i titoli superiori in senso gerarchico al paragrafo;
- contenuto (stringa): porzione di contenuto del paragrafo.

- page (intero): numero della pagina da dove è stato preso il contenuto del chunk (ancora non utilizzato, messo per successivi aggiornamenti del codice).

In questo momento i titoli, i parent dei titoli e il contenuto non venogono ancora concatenati, così, dopo essere entrati in possesso della lista dei Chunk, chi ci lavora può gestirli come preferisce. L'operazione che viene fatta nel momento della creazione del chunk è la pulizia del testo: la funzione **cleanText** usa delle espressioni regolari che ricercano sequenze di determinati caratteri uguali e li riduce ad un singolo carattere del tipo individuato, viene chiamata all'interno della funzione chunking.

La parte dell'**applicazione** effettiva in questo momento è implementata tramite un Jupiter Notebook ed è il prototipo che è stato reso disponibile all'inizio dello stage dall'azienda. Sono state aggiunte le chiamate alle funzioni implementate per ottenere le tabelle linearizzate e il chunking migliorato all'interno. La procedura che viene eseguita per far funzionare il prototipo è la seguente:

1. Viene dato un insieme di documenti;
2. Ogni documento viene convertito in XHTML tramite **extractText** che a sua volta linearizza le tabelle;
3. Vengono creati i chunk tramite la funzione **chunking**;
4. Al contenuto vengono concatenati i titoli dei chunk uniti con i titoli parent tramite la funzione **createTitleForChunk** da Chunking.py come spiegato nel paragrafo 2.2.2;
5. I vari chunk vengono caricati sul motore di ricerca;
6. Quando viene posta una domanda, vengono restituiti i 10 chunk con lo score più alto (hybrid search);
7. La richiesta (domanda + n chunk, dove n è determinato dalla quantità di token che può prendere in ingresso il Chat-Completion Model) viene consegnata al Chat-Completion Model che elabora le informazioni e restituisce una risposta alla domanda.

Capitolo 4

Verifica e validazione

In questo capitolo vengono fornite le motivazioni sul perchè i test sono stati effettuati in una determinata maniere, vengono riportati e discussi i dati e le prove effettuate sul RALM per ogni modifica aggiunta

4.1 Tipologia di Test

Inizialmente si pensava di effettuare i test sulle risposte del RALM tramite l'utilizzo di diverse metriche automatiche come *BERTScore, BARTScore e ROUGE* e altre metriche individuate nell'articolo "Kalpesh Krishna, Aurko Roy e Mohit Iyyer. «Hurdles to progress in long-form question answering». In: *arXiv preprint arXiv:2103.06332* (2021)" assieme alla valutazione umana. Per questioni di tempistiche, però, si è deciso di valutare la varie risposte solamente al livello manuale, umano. La valutazione umana, purtroppo, non è sempre affidabile però è stato il modo più semplice e rapido per capire di quanto è migliorata la qualità delle risposte.

4.2 Test

Come fonte di informazioni per il RALM sono state scelte cinque pagine di Wikipedia e sono state convertite nei formati desiderati (HTML, Pdf, Docx). La pagine scelte sono composte di testo e altri elementi semantici, tra cui le tabelle che hanno la struttura della tabella 2.1 spiegata nel paragrafo Linearizzazione delle tabelle

I test sono stati effettuati tramite una valutazione umana secondo i seguenti criteri:

- La risposta deve essere ben strutturata, scritta in maniera corretta e comprensibile;
- La risposta deve essere corretta e completa.

Sulla base di ciò a ogni risposta viene assegnato un valore tra 0 (risposta non corretta/strutturata male) e 1 (risposta corretta e strutturata correttamente)

4.2.1 Domande

Sono stati creati due set da 10 domande l'uno, le domande sono state create in maniera tale da poter interrogare il modello sui dati presenti nelle tabelle dei vari documenti:

- Set di domande semplici: domande che interrogano il RALM su una piccola quantità di celle delle tabelle;
- Set di domande complesse: domande che interrogano il RALM su diverse celle e che chiedono di fare diverse operazioni tra i dati presenti (es: somma, media).

Set di domande semplici

N.	**Nome pagina**	**Domanda**	**Risposta corretta**
1	Italia	Quali sono le 3 città italiane con più abitanti?	Roma, Milano, Napoli
2	Italia	In Italia, tra commercio, turismo e trasporti chi rappresenta un numero maggiore di imprese?	Il commercio
3	Avanti un altro!	In quali paesi è stato esportato Avanti un altro?	Spagna, Albania, Ungheria, Vietnam, Brasile, Bulgaria, Cile, Paraguay, Canada, Turchia, Polonia, Romania
4	Avanti un altro!	In che canale è stato trasmesso Avanti un altro?	Canale 5
5	SPAR	In che anno è arrivato SPAR in italia?	1959
6	SPAR	Quanto volume d'affari ha SPAR in Belgio	626.307
7	The Space Cinema	Quante sale ha il The Space Cinema di Padova?	14
8	The Space Cinema	Quanti posti a sedere ha il The Space Cinema di Parma centro	1402
9	Sonic(serie)	In che anno è uscito il videogioco "Sonic Battle"?	2003
10	Sonic(serie)	Dov'è apparso Tails per la prima volta?	Sonic the hedghog 2

Tabella 4.1: Set di domande semplici poste al RALM.

Set di domande complesse

N.	Nome pagina	Domanda	Risposta corretta
1	Italia	Quali sono i comuni italiani con più di 500000 abitanti?	Roma, Milano, Napoli, Torino, Palermo e Genova
2	Italia	Quali sono i comuni italiani che hanno più 300000 abitanti ma meno di 387843?	Bologna, Firenze, Bari, Catania
3	Avanti un altro!	Mediamente quanti telespettatori ci sono stati per ogni edizione di avanti un altro?	circa 3.5 milioni
4	Avanti un altro!	In quali paesi in cui è stato esportato avanti un altro è ancora in corso?	Canada, Romania
5	SPAR	In quali paesi è arrivato SPAR dal 2000 in poi?	Russia, Mauritius, Cipro, Ucraina, Cina, Croazia
6	SPAR	In quali paesi le strutture SPAR hanno meno di 10000 metri quadrati di superficie?	Croazia, Mauritious, Zambia, India, Albania
7	The Space Cinema	Quali sono i the space situati in provincia di Torino?	Beinsasco, Torino - Parco Dora
8	The Space Cinema	Quali sono i the space con più di 13 sale?	Limena, Roma Paroc de' Medici, Vimercate
9	Sonic(serie)	Chi è eggman e quando ha fatto la sua prima apparizione?	Eggman è un uomo esperto di ingegneria robotica che vuole conquistare il mondo di Sonic, antagonista principale della serie. Ha fatto la sua prima apparizione in "Sonic the hedghog".
10	Sonic(serie)	Elenca tutti i giochi mobile di Sonic dal 2015 in poi	Sonic runners, sonic dash 2: sonic boom, sonic runners adventure, sonic forces speed battle, sonic speed simulaor

Tabella 4.2: Set di domande complesse poste al RALM.

Nei seguenti paragrafi vengono riportate le risposte del RALM a ogni aggiornamento significativo del codice per denotare l'effettivo miglioramento della qualità delle risposte:

1. Prima versione: estrazione del contenuto semplice (senza linearizzazione delle tabelle, senza correzione del chunking e pulizia del testo);

2. Seconda verione: estrazione del contenuto con la linearizzazione delle tabelle;

3. Terza versione: estrazione del contenuto con la linearizzazione delle tabelle e la correzione del chunking e pulizia del testo;

4.2.2 Test sui file HTML

Prima versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Roma, Milano e Napoli sono le tre città italiane con più abitanti.	1
2	Il settore che presenta più imprese nel terziario italiano è il settore dei servizi.	0
3	Avanti un altro! è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro è stato trasmesso sul canale 5.	1
5	SPAR è arrivato in Italia nel 1959.	1
6	SPAR in Belgio	0
7	Il the space di Padova ha 14 sale	1
8	Il The Space di Parma Centro ha 7 sale e 1402 posti a sedere	1
9	Il videogioco Sonic Battle è stato pubblicato da Sega nel 2003.	1
10	Tails è un compagno di Sonic che appare per la prima volta in Sonic the Hedgehog 2 nel 1992.	1

Tabella 4.3: Risposte al set di domande semplici date dalla prima versione del RALM (File HTML).

Correttezza: 8/10

Prima versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Palermo, Cagliari, Trento, Venezia, Trieste, Bologna, Firenze, Perugia, Ancona, Roma, Napoli, Bari, Potenza, Catanzaro, Matera, Belluno	0
2	Napoli, Palermo, Genova, Bologna, Firenze e Bari.	0
3	In media si contano 3.995.000 spettatori per edizione.	0
4	Il programma "Avanti un altro!" è ancora in corso in Italia, mentre negli altri paesi in cui è stato esportato è chiuso.	0
5	SPAR è arrivata in Belgio, Danimarca, Francia, Germania, Irlanda, Italia, Lituania, Paesi Bassi, Polonia, Regno Unito, Russia, Sudafrica, Spagna, Svizzera, Ungheria, Zimbabwe.	0
6	SPAR è presente in Croazia, Slovenia, Austria, Germania, Regno Unito, Italia, Sudafrica, Norvegia, Giappone, Cina e Croazia.	0
7	The Space Cinema è una catena di cinema presente in provincia di Torino. Ha 8 sale e 1295 posti a sedere.	0
8	The Space Cinema è una delle catene di cinema più grandi in Italia, con 34 diverse strutture e 347 sale.	0
9	Eggman è un personaggio che è appena comparso nella serie Sonic. È uno scienziato malvagio che cerca di conquistare il mondo.	0
10	Sonic mobile games from 2015 onwards: Sonic Jump (2015), Sonic Dash (2013), Sonic Jump Fever (2014), Sonic Runners (2015), Sonic Dash 2: Sonic Boom (2015), Sonic Runners Adventure (2017), Sonic Forces: Speed Battle (2017), Sonic Speed Simulator (2022)	0

Tabella 4.4: Risposte al set di domande complesse date dalla prima versione del RALM (File HTML).

Correttezza: 0/10

Seconda versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Roma, Milano e Napoli sono le prime tre città italiane per popolazione.	1
2	Il commercio è il settore che presenta più imprese in Italia, seguito da turismo, trasporti e logistica, e altri servizi.	1
3	Avanti un altro! è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro è stato trasmesso sul canale 5.	1
5	SPAR è arrivato in Italia nel 1959.	1
6	SPAR ha un volume d'affari di 626.307 migliaia di € in Belgio.	1
7	Il the space di Padova ha 14 sale	1
8	The Space Cinema di Parma Centro ha 1402 posti a sedere.	1
9	Il videogioco Sonic Battle è uscito nel 2003.	1
10	Tails ha fatto la sua prima apparizione in Sonic the Hedgehog 2 (8-bit) nel 1992. Da allora, è apparso in un totale di 56 giochi.	1

Tabella 4.5: Risposte al set di domande semplici date dalla seconda versione del RALM (File HTML).

Correttezza: 10/10

Seconda versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le città italiane con più di 500.000 abitanti sono Milano, Roma, Torino, Palermo, Napoli e Genova.	1
2	Bologna, Firenze, Bari, Catania	1
3	Non è chiaro quanto sia l'edizione di Avanti un altro, ma si stima che ci siano stati 3 milioni di telespettatori per edizione.	0
4	Il gioco "Avanti un altro!" è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile e Bulgaria.	0
5	I Paesi in cui è arrivato SPAR dal 2000 in poi sono Russia, Mauritius, Cipro, Ucraina, Cina e Croazia.	1
6	I paesi in cui le strutture SPAR hanno meno di 10000 metri quadrati di superficie sono la Polonia, l'Australia, l'Ucraina, la Croazia e lo Zimbabwe.	0
7	I nomi dei the space situati in provincia di Torino sono Montesilvano, Napoli, Nola, Parma Campus, Parma Centro e Pradamano.	0
8	I cinema The Space con più di 13 sale sono: Rozzano, Salerno, Sestu, Silea.	0
9	Eggman è un personaggio che appare nella serie Sonic. È un uomo esperto di ingegneria robotica che vuole conquistare il mondo di Sonic. Ha fatto la sua prima apparizione in Sonic the Hedgehog nel 1991.	1
10	I giochi mobile di Sonic dal 2015 in poi sono: Sonic runners (2015), Sonic Dash 2: Sonic Boom (2015), Sonic Runners Adventure (2017), Sonic Forces: Speed Battle (2017) e Sonic Speed Simulator (2022).	1

Tabella 4.6: Risposte al set di domande complesse date dalla seconda versione del RALM (File HTML).

Correttezza: 5/10

Terza versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le tre città italiane con più abitanti sono Roma, Milano e Napoli.	1
2	Il settore del commercio rappresenta il maggior numero di imprese in Italia, con 1546 aziende.	1
3	Avanti un altro è stato esportato in diversi paesi, tra cui Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro è stato trasmesso sul canale 5.	1
5	SPAR è arrivato in Italia nel 1959.	1
6	Il volume d'affari di SPAR in Belgio è di 626.307 migliaia di €.	1
7	Il The Space Cinema di Padova ha 14 sale.	1
8	The Space di Parma Centro ha 1402 posti a sedere.	1
9	Il videogioco Sonic Battle è uscito nel 2003.	1
10	La prima apparizione di Tails è avvenuta nel videogioco Sonic the Hedgehog 2 (8-bit) nel 1992.	1

Tabella 4.7: Risposte al set di domande semplici date dalla seconda terza versione del RALM (File HTML).

Correttezza: 10/10

Terza versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	I comuni italiani con più di 500,000 abitanti sono Roma, Milano, Napoli e Torino.	0.7
2	I comuni italiani che hanno più di 300.000 abitanti ma meno di 387.843 sono Bologna, Firenze, Bari e Catania.	1
3	Mediamente, ogni edizione di "Avanti un altro!" ha avuto circa 3.5 milioni di telespettatori.	1
4	Il format "Avanti un altro!" è stato esportato e attualmente è in corso in Canada e Romania.	1
5	SPAR è arrivato in numerosi paesi a partire dal 2000, tra cui Mauritius, Cipro, Ucraina, Russia, Cina, Slovenia, Croazia, Ungheria e Repubblica Ceca.	0
6	Le strutture SPAR hanno meno di 10000 metri quadri di superficie in Albania, Croazia, Ucraina, Botswana e Namibia.	0
7	I cinema The Space situati in provincia di Torino sono Beinasco e Torino - Parco Dora.	1
8	I cinema The Space che hanno 13 o più sale sono Rozzano, Salerno, Sestu e Silea.	0
9	Dr. Eggman, conosciuto anche come Robotnik, è un uomo esperto di ingegneria robotica e l'antagonista principale della serie Sonic. Ha fatto la sua prima apparizione nel gioco Sonic the Hedgehog nel 1991.	1
10	I giochi mobile di Sonic usciti dal 2015 in poi sono Sonic Runners, Sonic Dash 2: Sonic Boom, Sonic Runners Adventure, Sonic Forces: Speed Battle, Sonic Speed Simulator (2022).	1

Tabella 4.8: Risposte al set di domande complesse date dalla terza versione del RALM (File HTML).

Correttezza: 6.7/10

4.2.3 Test sui file Docx

Prima versione - Risposte al set di domande semplici

N.	Risposta fornita dal RALM	Valutazione
1	Roma, Milano e Napoli.	1
2	Il settore terziario, composto principalmente da commercio, trasporti e logistica, turismo e altri servizi, rappresenta la maggior parte delle imprese italiane.	0
3	Il format è stato ideato da Paolo Bonolis e Stefano Santucci ed è stato esportato in: Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro! è un game show italiano in onda su Canale 5. Il programma va in onda dallo studio 1 del Centro Titanus Elios di Roma.	1
5	Il marchio SPAR è arrivato in Italia nel 1959.	1
6	SPAR ha un volume d'affari di 29,7 miliardi di euro in Belgio.	0
7	The Space Cinema di Padova ha 14 sale.	1
8	Il The Space di Parma Centro ha 7 sale e una capienza massima di 1.402 posti a sedere.	1
9	Il videogioco Sonic Battle è stato lanciato nel 2003.	1
10	Tails è apparso per la prima volta in Sonic the Hedgehog 2 nel 1992.	1

Tabella 4.9: Risposte al set di domande semplici date dalla prima versione del RALM (File Docx).

Correttezza: 8/10

Prima versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Roma, Milano, Napoli, Torino, Palermo, Genova, Bologna, Firenze, Bari, Catania	0
2	Napoli, Palermo, Genova, Bologna, Firenze e Bari.	0
3	Nelle prime quattro stagioni era presente in studio come responsabile della sicurezza il buttafuori Mediaset Girolamo Di Stolfo, noto come Big Jimmy, poi ricordato da Bonolis al termine della prima puntata della quinta edizione, in seguito alla sua morte avvenuta il 13 luglio 2015 per aneurisma cerebrale.	0
4	Il programma "Avanti un altro!" è ancora in corso in Italia, mentre per gli altri paesi non è specificato	0
5	SPAR è presente in 32 paesi in tutto il mondo.	0
6	I paesi in cui le strutture SPAR hanno meno di 10000 metri quadrati di superficie sono l'Albania, l'Austria, la Germania, il Regno Unito, l'Italia, la Sudafrica, il Giappone, la Spagna e l'Irlanda.	0
7	There are 9 The Space Cinemas located in the province of Turin.	0
8	The Space Cinema è una catena di multisale cinematografiche che opera in Italia. È la seconda catena di cinema più grande sul territorio italiano, superata solo dal circuito United Cinemas International.	0
9	Il Dr. Robotnik alias Eggman è un uomo esperto di ingegneria robotica che vuole conquistare il mondo di Sonic, antagonista principale della serie.	0.5
10	I giochi mobile di Sonic dal 2015 sono: Sonic Dash 2: Sonic Boom, Sonic Runners Adventure, Sonic Forces: Speed Battle, Sonic Speed Simulator.	1

Tabella 4.10: Risposte al set di domande complesse date dalla prima versione del RALM (File Docx).

Correttezza: 1.5/10

Seconda versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le 3 città italiane con più abitanti sono: Roma, Milano e Napoli.	1
2	Il settore che presenta più imprese nel settore terziario italiano è il commercio.	1
3	Il format Avanti un altro! è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro è stato trasmesso nel preserale di Canale 5	1
5	1959	1
6	626.307	1
7	The Space Cinema ha 14 sale e 3178 posti a sedere.	1
8	1402	1
9	Il videogioco Sonic Battle è uscito nel 2003.	1
10	Tails è apparso per la prima volta in Sonic the Hedgehog 2 nel 1992.	1

Tabella 4.11: Risposte al set di domande semplici date dalla seconda versione del RALM (File Docx).

Correttezza: 10/10

Seconda versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le città italiane con più di 500000 abitanti sono Roma, Milano, Napoli, Torino e Palermo.	0.8
2	8|Comune:Firenze|Città metropolitana: Firenze|Regione: Toscana|Abitanti :361 619|riga9->Pos.:9|Comune:Bari|Città metropolitana: Bari|Regione: Puglia|Abitanti :315 948| riga10->Pos.:10|Comune:Catania|Città metropolitana: Catania|Regione: Sicilia|Abitanti :301 104|	0
3	La media di spettatori per ogni edizione di "Avanti un altro!" è di circa 3.5 milioni.	1
4	Il gioco "Avanti un altro!" è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile e Bulgaria.	0
5	SPAR è presente in Croazia, Italia, Spagna, Francia, Belgio, Germania, Paesi Bassi, Sudafrica, Irlanda, Zimbabwe, Giappone, Argentina, Norvegia, Mauritius, Cipro, Ucraina, Lituania, Polonia, Australia e Nuova Zelanda.	0
6	I paesi in cui le strutture SPAR hanno meno di 10000 metri quadrati di superficie sono Polonia, Australia, Ucraina, Croazia e Zimbawe.	0
7	Per la provincia di Torino i cinema The Space si trovano a Torino - Parco Dora e Vimercate.	0.5
8	The Space Cinema è una catena di multisale cinematografiche con sede in Italia.	0
9	Eggman è un personaggio che appare per la prima volta in Sonic the Hedgehog nel 1991. È un esperto di ingegneria robotica che vuole conquistare il mondo di Sonic.	1
10	Sonic Runners (2015), Sonic Dash 2: Sonic Boom (2017), Sonic Runners Adventure (2017), Sonic Forces: Speed Battle (2017), Sonic Speed Simulator (2022)	1

Tabella 4.12: Risposte al set di domande complesse date dalla seconda versione del RALM (File Docx).

Correttezza: 4.3/10

Terza versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le tre città italiane con più abitanti sono Roma, Milano e Napoli.	1
2	Il settore che rappresenta un numero maggiore di imprese in Italia è il settore del commercio	1
3	Avanti un altro è stato esportato in diversi paesi, tra cui Cile, Paraguay, Canada, Turchia, Spagna, Albania, Polonia, Romania, Ungheria, Bulgaria, Vietnam e Brasile.	1
4	Avanti un altro è stato trasmesso su Canale 5.	1
5	Lo SPAR è arrivato in Italia nel 1959.	1
6	Il volume d'affari di SPAR in Belgio è di 626.307 euro.	1
7	Il The Space Cinema di Padova ha 14 sale.	1
8	Il The Space Cinema di Parma Centro ha 1402 posti a sedere.	1
9	Il videogioco Sonic Battle è uscito nel 2003.	1
10	La prima apparizione di Tails è avvenuta come personaggio giocabile nel videogioco arcade "Sonic the Hedgehog 2" uscito nel 1992.	1

Tabella 4.13: Risposte al set di domande semplici date dalla terza versione del RALM (File Docx).

Correttezza: 10/10

Seconda versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Roma, Milano, Napoli, Torino, Palermo, Genova, Bologna, Firenze, Bari, Catania	0
2	I comuni italiani che hanno più di 300000 abitanti ma meno di 387843 sono Bari e Firenze.	0.5
3	La media di telespettatori per ogni edizione di Avanti un altro è di circa 3,5 milioni.	1
4	Il format di "Avanti un altro" è attualmente ancora in corso in Canada e Romania.	1
5	SPAR è arrivato in vari paesi dal 2000 in poi, tra cui: Croazia, Italia, Ungheria, Repubblica Ceca, Slovenia, Sudafrica, Mauritius, Cipro, Ucraina, Russia, Argentina, Polonia, Lituania, Finlandia, Cina e Svizzera.	0
6	Le strutture SPAR hanno meno di 10000 metri quadri di superficie in Polonia, Ucraina, Croazia, Slovenia, Paesi Bassi, Repubblica Ceca, Svizzera, Zambia, Cina, Ungheria, Francia, Belgio, Grecia, Russia, Svezia, Botswana, Namibia, Albania e India.	0
7	I cinema The Space situati in provincia di Torino sono: - The Space Cinema a Beinasco - The Space Cinema a Torino	1
8	I cinema The Space che hanno 13 o più sale sono: - Vimercate con 16 sale - Roma Parco de' Medici con 18 sale - Limena con 14 sale	1
9	Eggman è un antagonista principale della serie di Sonic, è un esperto di ingegneria robotica che vuole conquistare il mondo di Sonic.. Ha fatto la sua prima apparizione nel videogioco Sonic the Hedgehog nel 1991.	1
10	- Sonic runners (2015) - Sonic Dash 2: Sonic Boom (2015) - Sonic runners adventure (2017) - Sonic forces: speed battle (2017) - Sonic speed simulator (2022)	1

Tabella 4.14: Risposte al set di domande complesse date dalla terza versione del RALM (File Docx).

Correttezza: 6.5/10

4.2.4 Test sui file Pdf

Prima versione - Risposte al set di domande semplici

N.	Risposta fornita dal RALM	Valutazione
1	Esistono 4 908 strutture museali ed archeologiche in Italia, più diffuse in Toscana (553), Emilia-Romagna (454) e Lombardia (433). Le prime 10 città sono Roma (121), Firenze (69), Torino (49), Milano (47), Bologna (46), Trieste (41), Genova (40), Napoli (38), Venezia (37), Siena (34).	0
2	Il settore terziario in Italia è composto principalmente da turismo, commercio, trasporti e logistica, e altri servizi.	0
3	paesi in cui è stato esportato Avanti un altro sono: Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Canale 5 nella fascia preserale di lunedì 5 settembre 2011.	0
5	Il primo supermercato SPAR in Italia è stato aperto nel 1960.	0
6	Volume d'affari: 4.156.400 migliaia di € superficie: 1.434 m^2 superficie media m^2/esercizio: 949.114	0
7	The Space Cinema di Padova ha 14 sale.	1
8	Parma Centro ha 7 sale da cinema con 1402 posti a sedere.	1
9	il videogioco Sonic Battle è uscito nel 2003	1
10	Tails è apparso per la prima volta in Sonic the Hedgehog 2 nel 1992.	1

Tabella 4.15: Risposte al set di domande semplici date dalla prima versione del RALM (File Pdf).

Correttezza: 5/10

Prima versione - Risposte al set di domande complesse

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le città italiane con più di 500mila abitanti sono Roma, Milano, Napoli, Torino, Palermo, Genova, Bologna, Firenze, Bari e Catania.	0
2	Roma, Milano, Napoli, Torino, Palermo, Genova, Bologna, Firenze, Bari, e Catania.	0
3	La prima edizione di Avanti un altro! Ha avuto 3.433.000 spettatori, mentre la puntata con il maggior numero di spettatori è stata quella della settima edizione, con 4.394.000 spettatori.	0
4	Non è specificato in quali paesi Avanti un altro! è ancora in corso.	0
5	Il marchio SPAR è presente in Croazia dal 2005, grazie alla struttura di ASPIAG. Altre aperture hanno riguardato le città di Sisak e Zagabria. ...	0
6	I Paesi in cui le strutture SPAR hanno meno di 10000 metri quadrati di superficie sono l'Albania, l'Austria, la Germania, il Regno Unito, l'Italia, la Sudafrica, la Norvegia, il Giappone e la Spagna.	0
7	The Space Cinema Beinasco e The Space Cinema Torino - Parco Dora sono situati in provincia di Torino.	1
8	The Space è una catena di multisale cinematografiche operante in Italia. ...	0
9	Eggman è un personaggio che appare nella serie di Sonic. Eggman è uno scienziato brillante, ma malvagio, che costruisce macchine e robot per cercare di conquistare il mondo. Eggman è un nemico principale di Sonic e i due si affrontano spesso. Eggman è apparso per la prima volta nel 1991 in "Sonic the Hedgehog".	1
10	Sonic the Hedgehog 3 Sonic & Knuckles Sonic the Hedgehog 4 Episodio 1 Sonic the Hedgehog 4 Episodio 2 Sonic Mania Sonic Mania Plus Sonic Mania Adventures Sonic Forces Sonic Lost World ...	0

Tabella 4.16: Risposte al set di domande complesse date dalla prima versione del RALM (File Pdf).

Correttezza: 2/10

Seconda versione - Risposte al set di domande semplici

N.	**Risposta fornita dal RALM**	**Valutazione**
1	Le tre città italiane con più abitanti sono Roma, Milano e Napoli.	1
2	Il settore con più imprese nel settore terziario italiano è il settore dei servizi, con oltre il 56% delle imprese totali.	0
3	Avanti un altro! è stato trasmesso in Italia, Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Avanti un altro! è stato trasmesso su Canale 5.	1
5	Spar è arrivato in Italia nel 1959.	1
6	Il volume d'affari di SPAR in Belgio è di 626.307 migliaia di €.	1
7	The Space Cinema ha 347 sale in 34 strutture in Italia.	0
8	Il the space di parma centro ha 7 sale e una capienza complessiva di 1402 posti.	1
9	Sonic Battle è un videogioco uscito nel 2003.	1
10	Tails è apparso per la prima volta in Sonic the Hedgehog 2 nel 1992.	1

Tabella 4.17: Risposte al set di domande semplici date dalla seconda versione del RALM (File Pdf).

Correttezza: 8/10

Seconda versione - Risposte al set di domande complesse

N.	Risposta fornita dal RALM	Valutazione
1	Le città con più di 500000 abitanti sono Roma, Milano, Napoli, Torino e Palermo.	0.8
2	I comuni italiani che hanno più di 300000 abitanti ma meno di 387843 sono Bari e Firenze.	0.5
3	Nelle prime quattro stagioni è stato registrato un pubblico di 98 telespettatori. Dalla quinta edizione in poi, il numero di telespettatori è cresciuto fino a raggiungere i 243 nella dodicesima stagione.	0
4	Il format di "Avanti un altro!" è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	0
5	SPAR è arrivato in diversi paesi dal 2000 in poi, tra cui la Cina, la Lituania, la Russia, l'Ucraina e la Croazia.	1
6	Spar è una catena di negozi di alimentari con punti vendita in Europa, Sudafrica, Asia, Oceania e America. In Italia, Spar opera principalmente con due insegne: Eurospar e Interspar.	0
7	The Space Cinema è una catena di cinema con sede in Italia. The Space Cinema ha 35 strutture e un totale di 353 sale in Italia.	0
8	The Space Cinema è una catena di multisale cinematografiche operante in Italia. È la seconda catena di cinema più grande sul territorio italiano, superata solo dal circuito United Cinemas International.	0
9	Eggman è uno dei nemici principali di Sonic e ha fatto la sua prima apparizione nel gioco Sonic the Hedgehog del 1991. È un esperto di ingegneria robotica che vuole conquistare il mondo	1
10	Sonic the Hedgehog è una serie di videogiochi creata da Sega. La serie è iniziata nel lontano 1991 con il videogioco omonimo e dal 2020 conta più di 30 videogiochi. I protagonisti della serie sono Sonic the Hedgehog, il suo fedele amico Tails Prower e Knuckles the Echidna. Il primo videogioco della serie, Sonic the Hedgehog, è stato un grande successo commerciale e ha venduto oltre 15 milioni di copie in tutto il mondo.	1

Tabella 4.18: Risposte al set di domande complesse date dalla seconda versione del RALM (File Pdf).

Correttezza: 3.3/10

Terza versione - Risposte al set di domande semplici

N.	Risposta fornita dal RALM	Valutazione
1	Le 3 città italiane con più abitanti sono Roma, Milano e Napoli.	1
2	Il settore del commercio rappresenta il numero maggiore di imprese in Italia.	1
3	Avanti un altro è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	1
4	Il programma Avanti un altro! è stato trasmesso sul canale Canale 5.	1
5	La catena SPAR è arrivata in Italia nel 1959.	1
6	Il volume d'affari di SPAR in Belgio è di 626.307 euro.	1
7	Il The Space di Padova ha 14 sale.	1
8	Il The Space di Parma centro ha 1402 posti a sedere.	1
9	Il videogioco Sonic Battle è uscito nell'anno 2003.	1
10	La prima apparizione di Tails è stata nel gioco "Sonic the Hedgehog 2" (8-bit) nel 1992.	1

Tabella 4.19: Risposte al set di domande semplici date dalla terza versione del RALM (File Pdf).

Correttezza: 10/10

Seconda versione - Risposte al set di domande complesse

N.	Risposta fornita dal RALM	Valutazione
1	I comuni con più di 500 000 abitanti in Italia sono: - Roma, con 2 749 031 abitanti - Milano, con 1 349 930 abitanti - Napoli, con 921 142 abitanti - Torino, con 848 748 abitanti - Palermo, con 635 439 abitanti - Genova, con 561 203 abitanti	1
2	I comuni italiani che hanno più di 300.000 abitanti ma meno di 387.843 sono Bologna, Firenze, Bari e Catania.	1
3	Non è specificato quante persone siano state in media telespettatori per ogni edizione di "Avanti un altro".	0
4	Il format di "Avanti un altro!" è stato esportato in Spagna, Albania, Ungheria, Vietnam, Brasile, Paraguay, Cile, Bulgaria, Canada, Turchia, Polonia e Romania.	0
5	SPAR è arrivato in diversi paesi dal 2000 in poi, tra cui Sudafrica, Russia, Polonia, Australia, Ucraina, Croazia, Cina, Giappone, Spagna, Irlanda, Danimarca, Finlandia, Ungheria, Francia, Belgio, Grecia, Slovenia, Repubblica Ceca, Svizzera, Paesi Bassi, Svezia, Botswana, Namibia e Zimbabwe.	0
6	Le strutture SPAR hanno meno di 10000 metri quadri di superficie in Albania.	0
7	I cinema The Space situati in provincia di Torino sono: Torino - Parco Dora, Beinasco.	1
8	I cinema The Space che hanno 13 o più sale sono: - Roma Parco de' Medici con 18 sale - Rozzano con 13 sale - Salerno con 11 sale - Sestu con 12 sale - Silea con 12 sale - Surbo con 9 sale - Terni con 9 sale - Torino - Parco Dora con 8 sale - Trieste con 7 sale - Verona - La Grande Mela con 8 sale - Vicenza con 9 sale - Vimercate con 16 sale.	0
9	Eggman è un personaggio della serie di Sonic, è l'antagonista principale e un esperto di ingegneria robotica che vuole conquistare il mondo di Sonic. Ha fatto la sua prima apparizione nel 1991.	1
10	Dopo il 2015 sono stati rilasciati i seguenti giochi mobile di Sonic: - Sonic the Hedgehog: Battle Racers (2018) - Sonic the Hedgehog: Crash Course (2018) - Sonic the Hedgehog: Dice Rush (2019) - Monopoly Gamer: Sonic The Hedgehog (2019)	0

Tabella 4.20: Risposte al set di domande complesse date dalla terza versione del RALM (File Pdf).

Correttezza: 4/10

Grafico Risposte corrette su versione

Qui di seguito ci sono i grafici che rappresentano l'andamento della quantità delle risposte corrette che vengono date dal RALM nelle sue varie versioni e per i tre formati prestabiliti.

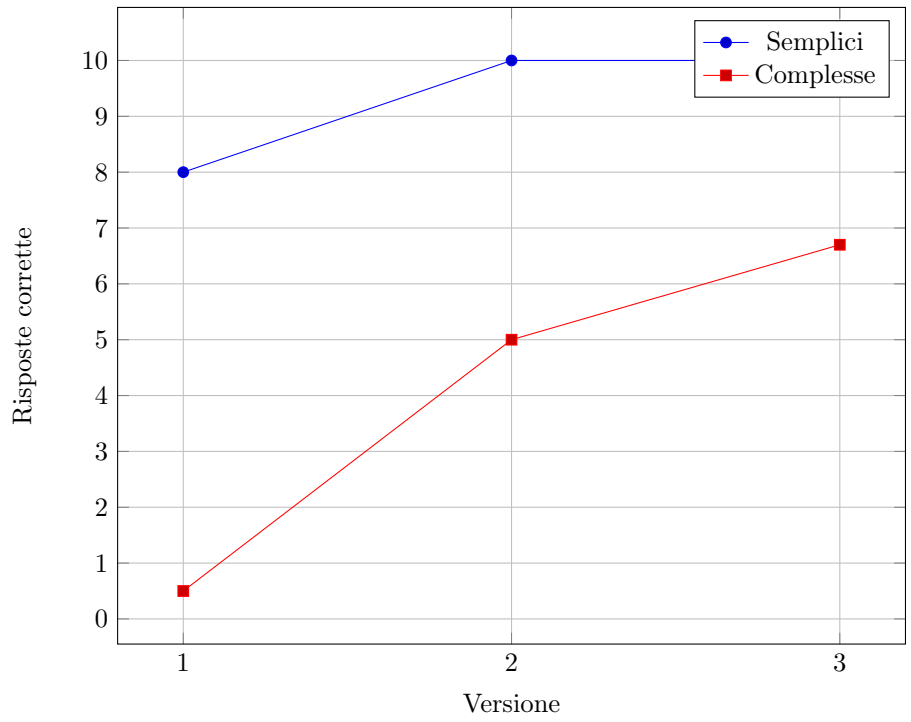


Figura 4.1: Grafico risposte corrette su versione - HTML.

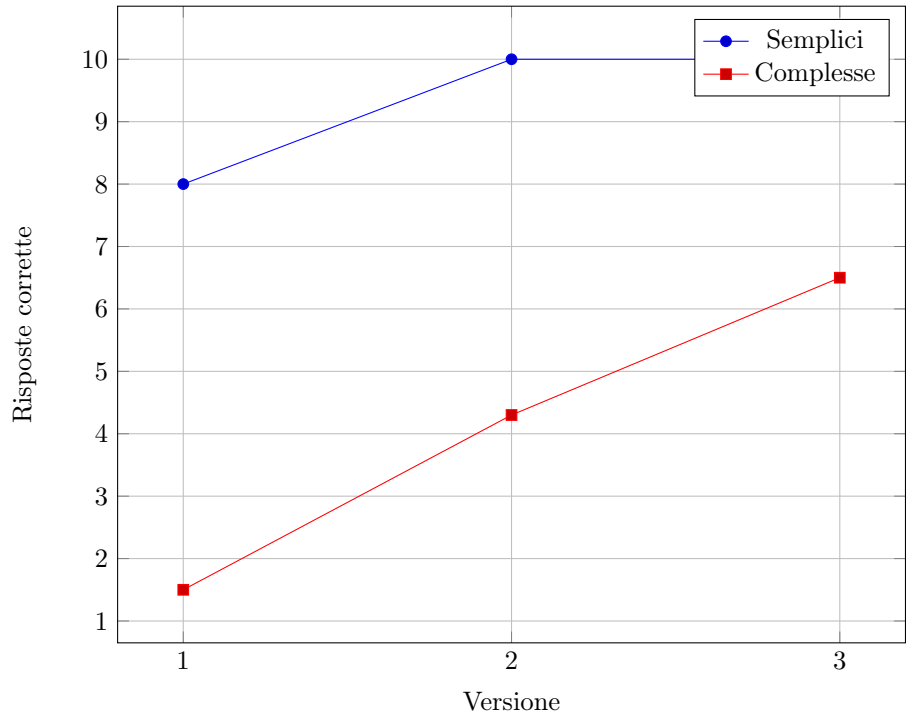


Figura 4.2: Grafico risposte corrette su versione - Docx.

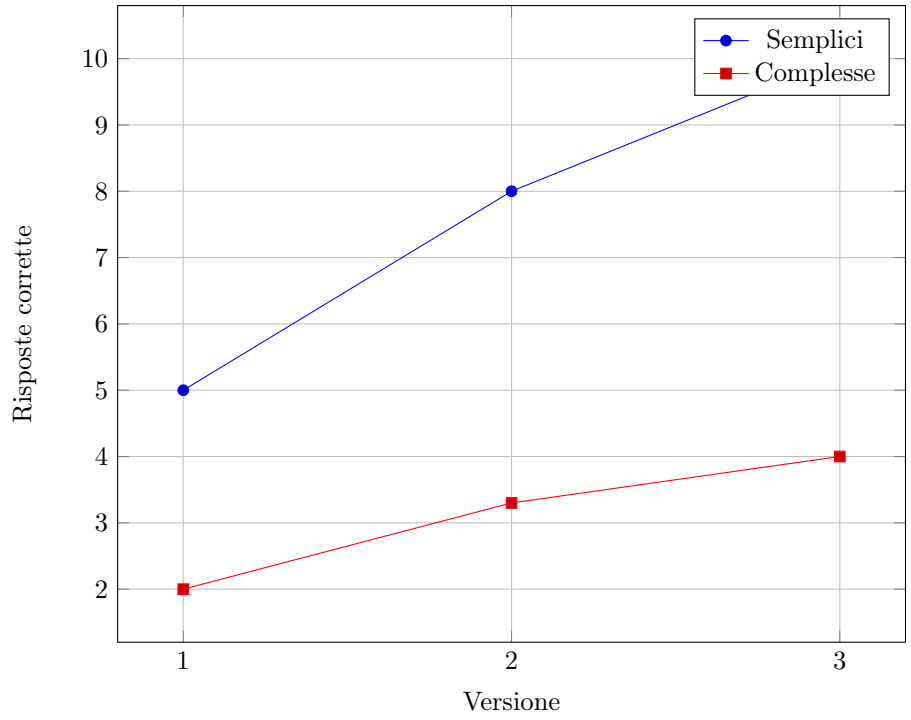


Figura 4.3: Grafico risposte corrette su versione - Pdf.

4.3 Considerazioni

Come si può notare dai grafici 4.1, 4.2 e 4.3 per le domande semplici, al livello base, c'era già un'alta percentuale di risposte corrette e già da dopo la linearizzazione delle tabelle sia per gli HTML che per i Docx i risultati sono stati ottimi. Per i Pdf, invece, il 10/10 è stato raggiunto dopo aver milgiorato il chunking e i casi in cui la risposta fornita dal RALM non è corretta, come le numero 2 e 7 presenti nella tabella 4.17 sono dovuti al fatto che la tabella viene spezzata e divisa in chunk diversi in maniera non giusta e quindi vengono perse delle informazioni sulla struttura o addirittura dati di alcune righe per esempio. Questa cosa sarebbe potuta succedere anche con i documenti HTML e Docx.

Per le domande complesse invece abbiamo un notevole miglioramento per gli HTML e i Docx e un discreto miglioramento per i Pdf. La differenza tra i valori finali è dovuta sicuramente al fatto che per i Pdf non si è in grado di trovare una struttura del testo tramite gli strumenti di estrazione a disposizione, quindi si perde la possibilità di inserire titoli e titoli parent all'interno del chunk stesso e di poter riconoscere la loro posizione all'interno della struttura del testo.

Capitolo 5

Conclusioni

5.1 Raggiungimento degli obbiettivi

Per quanto possibile sono stati completati tutti e tre gli obiettivi obbligatori. Come si può notare dai grafici riportati alla fine dello scorso capitolo per i documenti HTML e Docx c'è stato un notevole miglioramento della qualità delle risposte da parte del RALM. Per i Pdf invece il miglioramento è stato discreto. Questa differenza è dovuta al fatto che non riuscendo a convertire correttamente il file pdf in XHTML si perdono informazioni sulla struttura del contenuto. Per esempio nei chunk non è possibile individuare i titoli effettivi di un paragrafo.

5.2 Conoscenze acquisite

Come conoscenze teoriche ho approfondito meglio concetti su NLP e LLM come il chunking, questo grazie allo studio del RALM. Di interessante, ho appreso anche il funzionamento del ranking e, quindi, come vengono assegnati gli score ai chunk tramite l'algoritmo di ricerca ibrida. Al livello pratico, invece, durante questo percorso di stage ho sicuramente accresciuto le mie conoscenze sul linguaggio di programmazione Python che prima conoscevo in maniera abbastanza basilare, ora invece conosco anche diversi strumenti utili per l'estrazione di informazoni da documenti come tika, pdfplumber, Pandas e python-docx e strumenti per manipolare codice XHTML come BeautifulSoup. Ho imparato anche come utilizzare i modelli che fornisce OpenAI e come utlizzare il motore di ricerca Weaviate per fornire gli score ai chunk. Al livello personale invece ho appreso come migliorare il mio metodo di lavoro sotto il punto di vista dell'organizzazione, del problem solving e della collaborazione.

5.3 Materiale prodotto

5.3.1 Documentazione

Qui di seguito viene riportata la tabella che tratta brevemente della documentazione prodotta durante lo stage.

Titolo	Descrizione	Collegamenti
Appunti su tabelle	Contiene esempi su linearizzazioni di tabelle utili per agevolare il TQA.	
Estrazione delle tabelle da documenti	Contiene prove ed esempi che riguardano il funzionamento dei vari tool per l'estrazioni delle tabelle (Pandas, pdfplumber, python-docx).	
Sostituzione tabella del documento con tabella linearizzata	Contiene informazioni su come il contenuto dei documenti viene convertito da Tika e le varie soluzioni di replace applicate nei casi in cui converte correttamente e non il contenuto.	
Progettazione	Contiene tutte le informazioni riguardanti la progettazione software del prodotto e le motivazioni sulle scelte effettuate.	ExtractionTable, TextExtractor, Section, Chunking
ExtractionTable	Contiene tutte le informazioni dettagliate riguardanti la progettazione delle classi di estrazione delle tabelle	
TextExtractor	Contiene tutte le informazioni dettagliate riguardanti la progettazione delle classi di estrazione del contenuto dai documenti	
Section	Contiene tutte le informazioni dettagliate riguardanti la progettazione del modulo Section	
Chunking	Contiene tutte le informazioni dettagliate riguardanti la progettazione del modulo Chunking	
Idee chunking	Contiene diverse idee su come migliorare il chunking e le motivazioni a favore della scelta applicata.	
Test	Contiene i risultati raccolti dopo aver posto le domande sulle cinque pagine di Wikipedia prese in considerazione per i tre formati.	

Tabella 5.1: Tabella dei documenti prodotti durante lo stage.

Per scrivere la documentazione necessaria è stata utilizzata Evernote, un'applicazione per scrivere annotazioni in maniera semplice e veloce. Il punto forte di di Evernote è stata la possibilità di poter inserire dei collegamenti fra le varie note che sono state create: ad esempio il documento "Progettazione" ha dei riferimenti agli altri documenti che spiegano nel dettaglio le varie parti dell'architettura.

5.3.2 Codice sviluppato

Qui di seguito viene riportata la tabella che descrive brevemente i file di codice sviluppati durante il progetto.

Titolo file	Righe	Formato	Descrizione
AbstractExtractionTable	65	Python	Classe astratta che si occuppa della parte di algoritmo che gestisce l'etrazione, della linearizzazione e della sostituzione delle tabelle all'interno dei documenti.
ExtractionTableHTML	30	Python	Implementazione della classe astratta AbstractExtractionTable, si occupa delle operazioni sulle tabelle per i file HTML.
ExtractionTableDocx	24	Python	Implementazione della classe astratta AbstractExtractionTable, si occupa delle operazioni sulle tabelle per i file Docx.
ExtractionTablePdf	76	Python	Implementazione della classe astratta AbstractExtractionTable, si occupa delle operazioni sulle tabelle per i file Pdf.
ExtractionTableDefault	22	Python	Implementazione della classe astratta AbstractExtractionTable, si occupa delle operazioni sulle tabelle per i file che non hanno una classe specifica che implementa AbstractExtractionTable.
ExtractText	62	Python	Contiene le funzioni utili alla preparazione del contenuto per il chunking.
FactoryExtractText	22	Python	Contiene la funzione "builder" che istanzia un oggetto di tipo ExtractionTable del formato del documento sul quale si sta lavorando.
Section	105	Python	Contiene le funzioni per convertire codice XHTML in un albero "Sezione".
Chunking	303	Python	Contiene le funzioni per convertire documento in una lista di chunk, contiene funzioni per lavorare con i chunking.
ChunkExample		Jupyter Notebook	Contiene esempi sul funzionamento del codice sviluppato, partendo dalla sostituzione delle tabelle al chunking.
ExtractPdfTest		Jupyter Notebook	Contiene esempi sul funzionamento di alcuni tool di estrazione di tabelle dai documenti Pdf.
VectorStoreGeneration		Jupyter Notebook	Prototipo aziendale RALM moodificato con le aggiunte riguardanti le tabelle e il chunking.

Tabella 5.2: Tabella che riguarda i file di codice prodotti durante lo stage.

5.4 Consuntivo

Qui di seguito viene riportata la tabella che presenta il consuntivo:

Numero attività	Attività	Ore effettuate
1	Studio introduttivo su Natural Language Processing e Large Language Model	16
2	Studio delle tecniche di estrazione di testo e dei principali tool nell'ambito dell'NLP	16
3	Studio dell'attuale implementazione del chatbot basato su retrueval-augmented LLM	16
4	Analisi dei requisiti con studio delle casistiche da gestire	30
5	Progettazione delle varie componenti richieste nel paragrafo	66
6	Implementazione del software	96
7	Test e sperimentazione del software	24
8	Documentazione	40

Tabella 5.3: Tabella consuntivo.

Rispetto al preventivo presentato nella tabella 2.4 sono state effettuate alcune ore in meno dovute alla visualizzazione di alcuni corsi online che riguradavano norme aziendali, comunque le quantità orarie non discostano in maniera significativa.

Gli obbiettivi obbligatori presentati nella sezione 2.1 sono stati tutti raggiunti, mentre, l'obbiettivo desiderabile che riguradava l'interpretazione delle immagini non è stato completato per mancanza di tempo.

5.5 Valutazione personale

Per quanto mi riguarda sono molto soddisfatto del percorso di stage svolto. Durante questo periodo ho avuto l'opportunità di approfondire le mie conoscienze nel campo del LLM (dei RALM in particolare). Questa esperienza mi ha offerto una visione pratica del lavoro nel settore a tutti gli effetti e mi ha permesso di mettere in pratica ciò che ho imparato durante gli studi.

Oltre ad ampliare le mie conoscenze tecniche, ho anche sviluppato la mie abilità nel problem solving e nella collaborazione.

In conclusione posso dire che mi ha permesso di capire che quello che voglio fare è continuare a studiare le intelligenze artificiali, è un campo che mi affascina molto. Sono sicuro che quanto svolto mi sarà sicuramente utile in futuro.

Acronimi e abbreviazioni

BM25 Okapi Best Matching 25. 7, 43

LLM Large Language Model. 1, 43

NLP Natural Language Processing. 1, 43

RALM Retrieval Augmented Language Model. iii, 43

RRF Reciprocal Rank Fusion. 7, 43

TQA Table Question Answering. 5, 43

Glossario

Bag-of-Words Rappresentazione semplificata di un documento o di un testo in cui si ignora l'ordine delle parole e si considera la presenza o l'assenza dei vari termini. 7

Chat-Completion Model Modello di generazione del linguaggio artificiale che viene utilizzato per completare o generare testo in linguaggio naturale all'interno di una conversazione. 6

Chunk Piccola porzione di testo estratta da un documento. 4

Design pattern Soluzione progettuale generale ad un problema riccorrente, serve per risolvere problemi di progettazione che possono presentarsi diverse volte. 13

Overlap Sovrapposizione di elementi. Nel caso del chunking, l'overlap corrisponde nella parte finale e nella parte iniziale di due chunk consecutivi. 8

Parsing In informatica, il parsing è la tecnica che permette di estrapolare, decomporre e comprendere la struttura sintattica e semantica delle informazioni significative. 4

Question answering Campo dell'informatica e dell'intelligenza artificiale che si occupa di sviluppare sistemi in grado di comprendere e rispondere a domande poste dagli utenti in linguaggio naturale. 1

Strategy Pattern che tenta di isolare un algoritmo all'interno di un oggetto, in maniera tale da risultare utile in quelle situazioni dove sia necessaria modificare dinamicamente l'algoritmo stesso. 13

Token Singola unità linguistica o elemento individuale all'interno di un testo che può rappresentare per una parola, un simbolo di punteggiatura o anche una parte di una parola. 1

Bibliografia

Articoli consultati

Krishna, Kalpesh, Aurko Roy e Mohit Iyyer. «Hurdles to progress in long-form question answering». In: *arXiv preprint arXiv:2103.06332* (2021) (cit. a p. 15).

Talmor, Alon et al. «Multimodalqa: Complex question answering over text, tables and images». In: *arXiv preprint arXiv:2104.06039* (2021) (cit. a p. 6).

Siti web consultati

API OpenAI - models. URL: https://platform.openai.com/docs/introduction.

Siav S.p.A. URL: https://www.siav.com/it/ (cit. a p. 3).

Weaviate - Hybrid Search Explained. URL: https://weaviate.io/blog/hybrid-search-explained (cit. a p. 8).

Weaviate - What are Distance Metrics in Vector Search? URL: https://weaviate.io/blog/distance-metrics-in-vector-search (cit. a p. 7).

Wikipedia - Okapi BM25. URL: https://en.wikipedia.org/wiki/Okapi_BM25 (cit. a p. 7).