# UNIVPM-Synapse: Studio di Fattibilità Tecnica e Architettura Operativa per un Sistema di Intelligence Amministrativa e NLP Avanzato

## 1. Introduzione e Analisi del Contesto Strategico

L'evoluzione delle tecnologie di Elaborazione del Linguaggio Naturale (NLP) e l'avvento dei Large Language Models (LLM) hanno aperto scenari inediti per la digitalizzazione della Pubblica Amministrazione e, in particolare, per il settore universitario. La richiesta di sviluppare un sistema integrato di *Information Retrieval*, *Summarization*, *Translation* e *Question Answering* (QA) per l'Università Politecnica delle Marche (UNIVPM) intercetta un bisogno critico: la gestione della complessità informativa. L'ecosistema informativo di un ateneo moderno è caratterizzato da una vasta mole di dati non strutturati — bandi, decreti, regolamenti, avvisi di scadenza — la cui fruizione è spesso ostacolata da barriere burocratiche, linguistiche e tecniche.

Il presente rapporto tecnico, redatto con un approccio specialistico in architettura delle soluzioni AI e linguistica computazionale, analizza la fattibilità di tale sistema. L'obiettivo non è meramente tecnologico, ma organizzativo: trasformare il "dato inerte" presente nei PDF e nelle pagine HTML dell'ateneo in "conoscenza azionabile" per studenti, ricercatori e personale amministrativo. Attraverso l'analisi dettagliata delle fonti documentali reali dell'UNIVPM [1], il documento delineerà un'architettura software in grado di ingerire, comprendere e restituire informazioni complesse, come le scadenze per le borse di studio o i requisiti per i concorsi di ricercatore, garantendo al contempo il rispetto delle normative sulla privacy (GDPR) e l'affidabilità del dato.

La fattibilità del progetto è confermata dall'analisi, ma è condizionata alla risoluzione di specifiche sfide ingegneristiche legate al parsing di documenti amministrativi complessi e alla gestione delle allucinazioni nei modelli generativi. Nelle sezioni successive, verrà esplorata ogni componente del sistema, dalla strategia di *scraping* etico alla selezione di modelli LLM ottimizzati per la lingua italiana, fino alla definizione di una roadmap implementativa.

---

## 2. Analisi del Dominio Dati UNIVPM e Requisiti

# Funzionali

Per progettare un sistema NLP efficace, è imperativo comprendere la natura dei dati che esso dovrà elaborare. L'analisi dei portali dell'Università Politecnica delle Marche rivela un panorama informativo eterogeneo, dove informazioni cruciali per la carriera accademica sono frammentate su molteplici canali e formati.

## 2.1 Tassonomia e Morfologia delle Fonti Documentali

L'analisi dei frammenti di ricerca ha permesso di categorizzare le tipologie documentali principali che alimenteranno il sistema. Questa classificazione è fondamentale per definire le strategie di ingestione dati.

### 2.1.1 Bandi di Concorso e Personale (Dati ad Alta Strutturazione Implicita)

Una delle categorie più complesse è rappresentata dai bandi di concorso per personale docente e tecnico-amministrativo. Come evidenziato dai dati raccolti [1], questi documenti presentano caratteristiche peculiari:

- **Identificatori Univoci:** Ogni bando è associato a codici specifici (es. "Gruppo scientifico disciplinare 06/MEDS-25", "Settore concorsuale 08/CEAR-10"). L'accuratezza nel recupero di questi codici è vitale; un errore di una cifra può indirizzare l'utente al concorso sbagliato.
- **Scadenze Rigide e Temporali:** I bandi hanno date di scadenza precise (es. "15 dicembre 2025" o "2 maggio 2025").[1] Tuttavia, il sistema deve gestire la logica dello stato: un bando può essere "aperto", "chiuso", o in fase di "approvazione atti". Le informazioni sullo stato sono spesso aggiornate in tabelle HTML dinamiche che il sistema deve monitorare periodicamente.
- **Struttura Complessa:** I documenti originali sono quasi invariabilmente file PDF che contengono tabelle di valutazione, elenchi puntati di requisiti e riferimenti normativi incrociati.

### 2.1.2 Segreteria Studenti e Scadenze Amministrative (Logica Condizionale)

Le informazioni relative alle iscrizioni e alle tasse, come quelle per l'anno accademico 2025/2026, introducono una complessità logica significativa. Non si tratta di dati statici, ma di regole condizionali.

- **Logica Temporale Progressiva:** I dati mostrano scadenze scaglionate.[3] Ad esempio, il termine ordinario è il 5 novembre 2025. Tuttavia, l'iscrizione è permessa anche

successivamente con una mora progressiva: 25€ entro 60 giorni, 50€ oltre i 60 giorni. Un sistema QA deve essere in grado di calcolare la mora in base alla data corrente della query dell'utente.
- **Eccezioni:** Esistono eccezioni specifiche per i laureandi, che non devono rinnovare l'iscrizione se si laureano entro la sessione straordinaria.[6] Questa logica ("IF laureando THEN no iscrizione") deve essere codificata o appresa dal modello.

### 2.1.3 Borse di Studio e Opportunità (Dati Tabellari e Numerici)

I bandi per le borse di studio [4] sono ricchi di dati quantitativi che richiedono un'estrazione precisa per evitare "allucinazioni" numeriche.
- **Requisiti di Merito:** Vengono specificati voti minimi (es. diploma tra 80/100 e 100/100) e requisiti ISEE.
- **Importi Economici:** Gli importi variano significativamente (es. 2.000€ per le borse STEM e magistrali, 8.000€ per il corso MA.R.I. [8]).
- **Scadenze Prorogabili:** È stato rilevato che alcune scadenze subiscono proroghe (es. la borsa MA.R.I. prorogata al 24 novembre 2025 [4]). Il sistema deve essere in grado di riconoscere e sovrascrivere una data precedente con una nuova informazione di proroga.

## 2.2 Definizione dei Requisiti Utente e Personas

Sulla base delle tipologie di dati analizzate, il sistema deve soddisfare le esigenze di diversi profili utente, ciascuno con specifici requisiti di interazione.

| Persona | Esempio di Query | Requisito Funzionale Sottostante |
|---|---|---|
| **Studente Matricola** | "Quali borse ci sono per Ingegneria Informatica?" | **Filtraggio Semantico:** Associare "Ingegneria Informatica" ai bandi STEM [10] o generali, escludendo quelli per altri dipartimenti. |
| **Dottorando Internazionale** | "What is the deadline for enrollment?" | **Traduzione Cross-Lingua:** Comprendere l'inglese, cercare nei documenti italiani [3], tradurre la risposta e convertire le date nel formato locale. |
| **Personale Amministrativo** | "Dammi un riassunto dei requisiti per il bando RTT | **Summarization Strutturata:** Estrarre punti chiave (SSD, |

| | | |
|---|---|---|
| | MED/25." | Dipartimento, Scadenza) in formato scheda tecnica, ignorando il testo legale standard. |
| **Ricercatore Precario** | "Ci sono nuovi bandi RTT usciti questa settimana?" | **Aggiornamento Temporale:** Capacità di distinguere tra bandi "nuovi" e storici, richiedendo un *recency bias* nel retrieval. |

---

# 3. Modulo di Information Retrieval e Data Ingestion

La qualità delle risposte del sistema dipenderà interamente dalla robustezza della pipeline di acquisizione dati. Dato che UNIVPM non espone API pubbliche per questi dati, è necessario progettare un sistema di *Web Scraping* avanzato e resiliente, operante nel rispetto delle normative.

## 3.1 Strategie di Scraping e Crawling

L'architettura di acquisizione deve differenziare le strategie in base alla dinamicità delle pagine sorgente.

### 3.1.1 Scraper Deterministici per Albi e Bandi

Per le sezioni critiche come "Concorsi" [1] o "Bandi di Gara", dove la struttura HTML è prevedibile (spesso tabelle o liste), l'approccio migliore è l'uso di scraper deterministici. Strumenti come **Playwright** o le librerie Python basate su browser headless sono essenziali per gestire il rendering lato client (JavaScript) che spesso popola queste liste.[11]

- **Meccanismo:** Lo scraper navigherà periodicamente (es. ogni 24 ore) le pagine target, identificando nuovi elementi basandosi su selettori CSS o XPath stabili.
- **Gestione Aggiornamenti:** Sarà implementato un sistema di *hashing* dei contenuti. Se l'hash del contenuto HTML o del PDF allegato cambia, il sistema rileverà un aggiornamento (es. una rettifica al bando o una proroga) e innescherà una nuova indicizzazione.

### 3.1.2 Discovery Orizzontale con Trafilatura

Per intercettare notizie o avvisi pubblicati in sezioni meno strutturate (es. blog dei dipartimenti

o news in homepage), si raccomanda l'integrazione di **Trafilatura**.[12]

- **Vantaggi:** Trafilatura è ottimizzato per l'estrazione del "main text" (il contenuto principale), rimuovendo automaticamente boilerplate, menu di navigazione e footer che introdurrebbero rumore semantico nel database vettoriale. La sua capacità di gestire sitemap XML e feed RSS permette una scoperta efficiente di nuovi URL senza dover scansionare l'intero dominio a forza bruta.

### 3.1.3 Gestione delle Fonti Esterne ed Eterogenee

Il sistema deve gestire anche fonti esterne al dominio univpm.it. Ad esempio, i bandi per disabili e categorie protette sono gestiti anche tramite portali nazionali come InPA [13] o aggregatori come ConcorsiPubblici.[2]

- **Strategia:** Monitorare questi portali tramite i loro motori di ricerca interni (filtrando per "Università Politecnica delle Marche") per garantire ridondanza e completezza. Se il sito UNIVPM è temporaneamente non aggiornato, InPA potrebbe avere l'informazione più recente.

## 3.2 La Sfida dell'Estrazione da PDF (Document Intelligence)

La maggior parte dei dati amministrativi risiede in file PDF, spesso scansionati o con layout complessi. Le librerie di estrazione standard falliscono nel preservare la struttura logica necessaria per un RAG accurato.

### 3.2.1 Analisi Comparativa delle Librerie di Parsing

Sulla base delle evidenze tecniche [15], si delinea la seguente gerarchia di strumenti per l'ecosistema Python:

| Libreria | Caso d'Uso Ottimale | Limitazioni Rilevate | Applicazione in UNIVPM |
|---|---|---|---|
| **PyPDF2 / PDFMiner** | Testo semplice, nativo digitale. | Perde totalmente la struttura delle tabelle; mescola colonne di testo adiacenti. | Inadatto per i bandi complessi. |
| **Camelot / Tabula** | PDF nativi con tabelle grigliate. | Richiede parametri di configurazione manuali; fallisce su scansioni o tabelle senza bordi. | Utile per allegati tecnici strutturati (es. piani di studio). |
| **PDFPlumber** | Layout analysis | Buono per estrarre | Buono per il |

| | | | |
|---|---|---|---|
| | generale. | testo preservando posizioni, ma lento su documenti grandi. | pre-processing. |
| **Unstructured.io** | Pipeline all-in-one. | Integra diverse strategie ma può essere pesante computazionalmente. | Ottima base per la pipeline generica. |

### 3.2.2 Soluzione Proposta: Pipeline Ibrida Multimodale

Per superare i limiti dei parser tradizionali, specialmente con tabelle complesse come quelle dei requisiti ISEE o dei punteggi concorsuali [17], si propone un approccio innovativo basato su **Vision-Language Models (VLM)**.

1. **Preprocessing:** Il documento viene analizzato per determinare se è "text-based" o "image-based".
2. **OCR Selettivo:** Se scansionato, si utilizza un motore OCR avanzato. Strumenti come **Amazon Textract** [18] o, in ambito open source, **DocTR** offrono prestazioni superiori su documenti multilingua (italiano incluso) rispetto al classico Tesseract.[19]
3. **Estrazione Tabellare via LLM:** Le pagine contenenti tabelle vengono convertite in immagini e passate a un modello multimodale (es. GPT-4o, Claude 3.5 Sonnet o un modello open locale come **Llava** o **Qwen-VL**). Il prompt istruirà il modello a trascrivere la tabella visiva in formato Markdown o JSON strutturato.[20] Questo passaggio è cruciale per permettere al sistema QA di rispondere a domande come "Qual è il voto minimo per la borsa X?", che richiede la lettura accurata di una riga specifica in una tabella.

## 3.3 Considerazioni Legali: Web Scraping e GDPR in Italia

L'acquisizione massiva di dati da portali universitari impone una rigorosa analisi di conformità al GDPR e alle linee guida del Garante Privacy italiano.[22]

### 3.3.1 Dati Personali nelle Graduatorie

I documenti scaricati (es. esiti concorsi, graduatorie borse di studio) contengono spesso Dati Personali Identificabili (PII) come nomi, cognomi e matricole.

- **Rischi:** L'indicizzazione di questi dati senza una base giuridica specifica (come il legittimo interesse bilanciato o un compito di interesse pubblico esplicito) espone a sanzioni. Il Garante italiano ha già sanzionato attività di scraping indiscriminato che

raccoglievano dati personali per finalità diverse da quelle originali.[23]

- **Mitigazione Obbligatoria:** Il sistema deve implementare una pipeline di **Anonymization/Redaction** *prima* che i dati vengano salvati nel database vettoriale.
    - Utilizzo di librerie come **Microsoft Presidio** o modelli NER (Named Entity Recognition) specifici per l'italiano (es. spacy-it) per identificare e oscurare nomi di persone, mantenendo visibili solo i ruoli (es. "Il Vincitore") e i dati amministrativi.

### 3.3.2 Proprietà Intellettuale e Diritto Sui Generis

Sebbene i bandi siano atti pubblici, la struttura organizzata del database del sito web potrebbe essere protetta dal diritto *sui generis* sulle banche dati.[25]

- **Strategia:** Il crawling deve essere "gentile" (rispetto del robots.txt, rate limiting) per non impattare sulla disponibilità del servizio (evitando accuse di DoS). Inoltre, il sistema non deve replicare il sito ("mirroring"), ma agire come un indice intelligente che rimanda sempre alla fonte originale per la verifica finale.

---

# 4. Architettura Semantica: Embedding e Vettorializzazione

Una volta estratto e pulito, il testo deve essere convertito in una rappresentazione matematica (vettori) che ne catturi il significato semantico, permettendo al sistema di trovare informazioni rilevanti anche senza una corrispondenza esatta delle parole chiave.

## 4.1 Selezione del Modello di Embedding per l'Italiano

La scelta del modello di embedding è critica. Un modello addestrato solo sull'inglese potrebbe non cogliere le sfumature del linguaggio burocratico italiano (es. la differenza tra "immatricolazione" e "iscrizione").
L'analisi dei benchmark attuali 27 suggerisce diverse opzioni ad alte prestazioni:

| Modello | Tipo | Vantaggi per UNIVPM | Note |
|---|---|---|---|
| **Voyage-3-large** | Commerciale | Leader attuale per rilevanza e multilinguismo.[27] Ottimo su domini tecnici. | Costo per token. |
| **BGE-M3 (BAAI)** | Open Source | Supporto multilingue eccellente, gestisce | Ideale per deployment locale. |

| | | | |
|---|---|---|---|
| | | sequenze lunghe (8192 token) e modalità ibrida (dense + sparse). | |
| **Multilingual-E5-Large** | Open Source | Ottimo bilanciamento tra performance e dimensione. | Standard industriale solido. |
| **Italian-Legal-BERT** | Specifico | Addestrato su corpus giuridici italiani.[30] | Finestra di contesto piccola (512 token), meno adatto a documenti lunghi rispetto a BGE-M3. |

**Raccomandazione:** Si suggerisce l'adozione di **BGE-M3**. La sua capacità di gestire finestre di contesto ampie è fondamentale per elaborare interi articoli di regolamenti senza doverli frammentare eccessivamente, preservando la coerenza locale. Inoltre, la sua natura ibrida (vettoriale + keyword) aiuta a recuperare codici specifici (es. "SSD MED/25") che i modelli puramente semantici talvolta "diluiscono".

### 4.2 Strategia di Chunking (Segmentazione)

Non è possibile vettorializzare un intero bando in una volta sola. Il testo deve essere diviso in segmenti (*chunks*).

- **Chunking Semantico:** Invece di tagliare il testo ogni 500 parole, si utilizzerà un approccio strutturale che rispetta i confini logici del documento (sezioni, articoli, commi). Per i bandi UNIVPM, questo significa creare un chunk per ogni "Articolo" del bando, includendo nel metadato il titolo del bando e la data di scadenza. Questo contesto arricchito [31] migliora drasticamente la precisione del recupero.

---

## 5. Modulo di Generazione e Question Answering (RAG)

Il cuore "intelligente" del sistema è il modulo RAG (*Retrieval-Augmented Generation*), che combina i documenti recuperati con la capacità generativa di un LLM.

### 5.1 Selezione del Large Language Model (LLM)

Il modello generativo deve possedere eccellenti capacità nella lingua italiana, un'ampia finestra di contesto per analizzare molteplici documenti recuperati, e forti capacità di

ragionamento per interpretare regole complesse.

### 5.1.1 Analisi dei Candidati

Sulla base dei benchmark e delle release recenti [32], i candidati principali sono:

- **Mistral Large 2 (mistral-large-2407):**
  - *Punti di Forza:* Finestra di contesto di 128k token, addestramento nativo su italiano, eccellenti capacità di ragionamento logico e matematico (superiori a Llama 3 405B in alcuni benchmark di coding e math).[32] È particolarmente "cauto" nel generare risposte, riducendo le allucinazioni, caratteristica vitale in ambito legale/amministrativo.
  - *Deployment:* Disponibile via API (La Plateforme) o Azure, garantendo scalabilità.
- **Llama 3.1 (70B/405B):**
  - *Punti di Forza:* Modello open-weight estremamente potente, con forti capacità multilingue e tool-use.[34] La versione 70B è un ottimo compromesso per l'hosting locale se si desidera sovranità totale sui dati.
  - *Utilizzo:* Ideale se l'ateneo dispone di infrastruttura GPU (es. NVIDIA A100/H100) on-premise.
- **Modelli Italiani (es. Italia-9B, Minerva):**
  - Sebbene promettenti [37], spesso mancano della capacità di ragionamento profondo e della finestra di contesto massiva dei modelli di frontiera (Mistral/Llama/GPT-4) necessarie per compiti complessi di RAG su documenti lunghi.

**Scelta Raccomandata: Mistral Large 2** per la sua combinazione di competenza linguistica europea, context window (cruciale per leggere interi regolamenti) e affidabilità. In alternativa, **Llama 3.1 70B** per scenari self-hosted.

## 5.2 Architettura RAG Avanzata: Agentic RAG

Un semplice sistema RAG ("trova testo -> rispondi") non è sufficiente per domande come "Conviene fare la domanda per la borsa X o Y?". Si propone un'architettura **Agentic RAG**:

1. **Query Analysis:** L'LLM analizza la domanda. Se è semplice ("Quando scade X?"), esegue una ricerca diretta. Se è complessa ("Confronta i requisiti di X e Y"), pianifica più ricerche.
2. **Hybrid Retrieval:** Combina ricerca vettoriale (per concetti) e ricerca per parole chiave (BM25) per trovare codici bando o acronimi specifici.
3. **Re-ranking:** I documenti trovati vengono riordinati da un modello *Cross-Encoder* (es. bge-reranker-v2-m3) per garantire che i paragrafi letti dall'LLM siano i più pertinenti.
4. **Generation with Citations:** L'LLM genera la risposta citando esplicitamente le fonti (es. "Vedi Art. 4 del Bando [Link]"). Questo è fondamentale per la trasparenza

amministrativa.

---

# 6. Modulo di Summarization e Traduzione

## 6.1 Summarization Gerarchico

Per i bandi molto lunghi, si implementerà una strategia di riassunto gerarchico. Il documento viene diviso in capitoli; ogni capitolo viene riassunto; i riassunti parziali vengono aggregati in una scheda sintetica finale.

- **Template di Output:** I riassunti non saranno testo libero, ma JSON strutturati contenenti campi chiave: Titolo, Scadenza, Requisiti, Importo, Modalità Candidatura. Questo permette di visualizzare i dati in dashboard grafiche.

## 6.2 Traduzione e Internazionalizzazione

Per gli utenti internazionali, il sistema deve abbattere la barriera linguistica.

- **Approccio:** Invece di tradurre i documenti staticamente (costoso), si utilizzerà la traduzione *on-the-fly* durante la generazione della risposta.
- **Tecnologia:** I moderni LLM come Llama 3.1 e Mistral Large 2 hanno dimostrato prestazioni di traduzione competitive con sistemi dedicati per le lingue europee.[36]
    - *Flusso:* Utente chiede in Inglese -> Sistema cerca nei documenti Italiani -> LLM elabora le informazioni italiane e genera la risposta direttamente in Inglese. Questo riduce il rischio di errori di traduzione nel documento sorgente ("lost in translation") e mantiene la risposta ancorata alla fonte ufficiale.
    - *Fallback:* Per lingue meno comuni, si può integrare il modello open source **NLLB-200** (No Language Left Behind) di Meta [40], specializzato nella traduzione di oltre 200 lingue.

---

# 7. Bozza di Progetto: Roadmap e Stima Risorse

Di seguito viene presentata una pianificazione operativa per lo sviluppo del sistema.

## 7.1 Fasi di Sviluppo (Cronoprogramma 6 Mesi)

| Fase | Durata | Attività Principali | Milestone |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **1. Inception & Data** | Mese 1 | Analisi legale GDPR; Setup infrastruttura scraping (Crawlee/Playwright); Raccolta dataset iniziale bandi (storico 12 mesi). | Dataset grezzo acquisito; Pipeline Privacy configurata. |
| **2. Pipeline Core** | Mese 2-3 | Implementazione parsing PDF ibrido (Unstructured + Vision LLM); Creazione indici vettoriali (Qdrant + BGE-M3). | API di Ingestion funzionante; Dati strutturati nel DB. |
| **3. RAG & Logic** | Mese 4 | Sviluppo logica RAG e Prompt Engineering su Mistral Large 2; Test su domande complesse (date, requisiti). | Chatbot Backend funzionante (Alpha). |
| **4. Interface & Integration** | Mese 5 | Sviluppo Frontend Web (Streamlit/React); Integrazione modulo traduzione; Dashboard di amministrazione. | Beta Release per test limitato. |
| **5. Testing & Deploy** | Mese 6 | User Acceptance Testing (UAT) con studenti campione; Security Audit; Deploy in produzione. | Go-Live 1.0. |

## 7.2 Stima dei Costi e Risorse Tecnologiche

- **Personale:** 1 ML Engineer (Backend/RAG), 1 Data Engineer (Scraping/Pipelines), 1 Frontend Dev, 1 Esperto Legale (consulenza part-time).
- **Infrastruttura (Opzione Ibrida):**
    - *Storage & Vector DB:* Qdrant Cloud o istanza gestita (~50-100€/mese).
    - *Compute (Scraping/OCR):* Istanze CPU/GPU spot per i batch di ingestione notturni.
    - *Inference LLM:* API Mistral o OpenAI per la fase iniziale (Pay-per-token) per minimizzare l'investimento iniziale (CapEx). Passaggio a Llama 3.1 70B self-hosted su server d'ateneo (es. 2x NVIDIA A100) se i volumi aumentano, per abbattere i

costi operativi (OpEx).

### 7.3 Rischi e Mitigazione

1. **Volatilità dei Siti Web:** Se UNIVPM cambia il layout del portale, gli scraper si rompono.
    - *Mitigazione:* Monitoraggio proattivo degli errori di scraping e architettura modulare per aggiornamenti rapidi dei selettori.
2. **Allucinazioni su Scadenze:** L'LLM potrebbe inventare una data.
    - *Mitigazione:* Implementare un "Verificatore Deterministico" post-generazione che controlla se le date citate nel testo generato esistono effettivamente nei documenti recuperati. Obbligo di disclaimer: "Controlla sempre il bando ufficiale".
3. **Compliance GDPR:** Rischio di indicizzare dati sensibili.
    - *Mitigazione:* Redazione automatica rigorosa (PII masking) e policy di esclusione per documenti contenenti parole chiave come "graduatoria provvisoria" o "esiti".

## 8. Conclusione

La realizzazione del sistema NLP per l'Università Politecnica delle Marche è **tecnicamente fattibile** e rappresenta un'opportunità strategica di modernizzazione. Le tecnologie attuali (RAG, Vision-Parsing, LLM Multilingua) sono sufficientemente mature per gestire la complessità dei dati amministrativi italiani, a patto di adottare un approccio ingegneristico rigoroso nella fase di ingestione e pulizia dei dati.
Il valore aggiunto risiede nella capacità di trasformare un labirinto burocratico in un'interfaccia conversazionale semplice, accessibile e multilingue. Raccomandiamo di avviare un Proof of Concept (PoC) focalizzato su un singolo dominio verticale (es. "Borse di Studio e Tasse") per validare la pipeline di estrazione tabellare e l'efficacia del modello di linguaggio prima di estendere il sistema all'intero corpus documentale dell'ateneo.

#### Bibliografia


1. Ultime notizie dei concorsi - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Concorsi
2. Concorsi attivi Università Politecnica delle Marche 2025: Bandi per ente pubblico, accesso eseguito il giorno novembre 28, 2025, https://www.concorsipubblici.com/concorsi/ente/ent/universita-politecnica-delle-marche-33558
3. Segreteria studenti Ingegneria - Immatricolazione/iscrizione C.L. Magistrale - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Servizi_agli_studenti/Segreterie_Studenti/Ingegneria/Immatricolazione_iscrizione_CL_Magistrale
4. Borse di studio per le matricole Univpm - UnivpmOrienta, accesso eseguito il

giorno novembre 28, 2025, https://www.orienta.univpm.it/borse-di-studio-per-le-matricole-univpm/
5. Bandi di concorso per ricercatore a tempo determinato - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Ateneo/Bandi_concorsi_e_gare/Concorsi/Personale_docente/Riepilogo_concorsi_ricercatori_tempo_determinato/Concorsi_ricercatori_tempo_determinato_-_in_corso_di_svolgimento
6. Iscrizione anni successivi al primo -Corsi di laurea triennali - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Didattica/Immatricolazioni_tasse_borse_lauree/Immatricolazione_Iscrizione_Corsi_di_laurea_triennali
7. Economia scadenze A. A. 2025-2026 - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Universita_Politecnica_delle_Marche_Home/5_passi_per_iscriverti_a_UNIVPM/Segreterie_Studenti/Economia_1/Economia_scadenze_A_A_2025-2026_-Segreteria_studenti_economia
8. Bando 30 borse di studio per le matricole della LT Management per la Valorizzazione Sostenibile delle Aziende e delle Risorse Ittiche 2025 - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Course_catalogue/Undergraduate_Degree_in_Computer_and_Automation_Engineering/Offerta_formativa_1/Corso_di_Laurea_in_Management_per_la_valorizzazione_sostenibile_delle_aziende_e_delle_risorse_ittiche/Bando_30_borse_di_studio_per_le_matricole_della_LT_Management_per_la_Valorizzazione_Sostenibile_delle_Aziende_e_delle_Risorse_Ittiche_2025
9. Borse studio matricole Corso laurea magistrale (ad accesso libero) 2025 - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Pensioni/Tipologie_di_pensione_e_requisiti_di_accesso/Referenti_per_lOrientamento/Checklist_for_international_students/Borse_studio_studentesse_studenti/Borse_studio_matricole_Corso_laurea_magistrale_ad_accesso_libero_2025
10. Borse studio studentesse immatricolate Corso laurea triennale STEM 2025 - UNIVPM, accesso eseguito il giorno novembre 28, 2025, https://www.univpm.it/Entra/Servizi_agli_studenti/Borse_studio_studentesse_studenti/Borse_studio_studentesse_immatricolate_Corso_laurea_triennale_STEM_2025
11. scraper · GitHub Topics, accesso eseguito il giorno novembre 28, 2025, https://github.com/topics/scraper
12. adbar/trafilatura: Python & Command-line tool to gather text and metadata on the Web: Crawling, scraping, extraction, output as CSV, JSON, HTML, MD, TXT, XML - GitHub, accesso eseguito il giorno novembre 28, 2025, https://github.com/adbar/trafilatura
13. Concorso riservato collaboratore Ancona 2025 presso Università Politecnica delle Marche : Bando pubblico per 6 posti, accesso eseguito il giorno novembre 28, 2025, https://www.concorsipubblici.com/universita-politecnica-delle-marche-concorso

-riservato-collaboratore-ancona-2025
14. Università Politecnica delle Marche - selezione pubblica - riservata Legge 68/1999 - inPA, accesso eseguito il giorno novembre 28, 2025, https://www.inpa.gov.it/bandi-e-avvisi/dettaglio-bando-avviso/?concorso_id=04f6ac5c72454369a1df02bf35161356
15. Python Libraries to Extract Tables From PDF: A Comparison - Unstract, accesso eseguito il giorno novembre 28, 2025, https://unstract.com/blog/extract-tables-from-pdf-python/
16. RAG — Three Python libraries for Pipeline-based PDF parsing - AI Bites, accesso eseguito il giorno novembre 28, 2025, https://www.ai-bites.net/rag-three-python-libraries-for-pipeline-based-pdf-parsing/
17. Extracting structured table data from PDF files for RAG implementation, accesso eseguito il giorno novembre 28, 2025, https://community.latenode.com/t/extracting-structured-table-data-from-pdf-files-for-rag-implementation/34799
18. What is OCR Software? - AWS, accesso eseguito il giorno novembre 28, 2025, https://aws.amazon.com/what-is/ocr-software/
19. Python OCR libraries for converting PDFs into editable text - Ploomber, accesso eseguito il giorno novembre 28, 2025, https://ploomber.io/blog/pdf-ocr/
20. From PDF tables to insights: An alternative approach for parsing PDFs in RAG - Elastic, accesso eseguito il giorno novembre 28, 2025, https://www.elastic.co/search-labs/blog/alternative-approach-for-parsing-pdfs-in-rag
21. How to parse PDF docs for RAG - OpenAI Cookbook, accesso eseguito il giorno novembre 28, 2025, https://cookbook.openai.com/examples/parse_pdf_docs_for_rag
22. The state of web scraping in the EU - IAPP, accesso eseguito il giorno novembre 28, 2025, https://iapp.org/news/a/the-state-of-web-scraping-in-the-eu
23. The Italian Data Protection Authority puts a stop to 'web scraping' - Morri Rossetti & Franzosi, accesso eseguito il giorno novembre 28, 2025, https://morrirossetti.it/en/insight/publications/the-italian-data-protection-authority-puts-a-stop-to-web-scraping.html
24. Lawfulness of the mass processing of publicly accessible online data to train large language models - Oxford Academic, accesso eseguito il giorno novembre 28, 2025, https://academic.oup.com/idpl/article/14/4/326/7816718
25. Web Scraping: A Private Law Perspective - i-lex, accesso eseguito il giorno novembre 28, 2025, https://i-lex.unibo.it/article/download/18875/17437/75073
26. The normative challenges of data scraping: legal hurdles and steps forward - UniTo, accesso eseguito il giorno novembre 28, 2025, https://iris.unito.it/retrieve/8668e9fa-1fa8-4059-a672-0a30d1ccaa49/18905-Articolo-75189-2-10-20240110.pdf
27. The Best Embedding Models for Information Retrieval in 2025 - DEV Community, accesso eseguito il giorno novembre 28, 2025, https://dev.to/datastax/the-best-embedding-models-for-information-retrieval-in-

2025-3dp5
28. 5 Best Embedding Models for RAG: How to Choose the Right One - GreenNode, accesso eseguito il giorno novembre 28, 2025, https://greennode.ai/blog/best-embedding-models-for-rag
29. 9 Best Embedding Models for RAG to Try This Year - ZenML Blog, accesso eseguito il giorno novembre 28, 2025, https://www.zenml.io/blog/best-embedding-models-for-rag
30. Italian Legal BERT · Models - Dataloop AI, accesso eseguito il giorno novembre 28, 2025, https://dataloop.ai/library/model/dlicari_italian-legal-bert/
31. Long Context RAG Performance of LLMs | Databricks Blog, accesso eseguito il giorno novembre 28, 2025, https://www.databricks.com/blog/long-context-rag-performance-llms
32. Mistral Large 2 | Open Laboratory, accesso eseguito il giorno novembre 28, 2025, https://openlaboratory.ai/models/mistral-large-2
33. Large Enough | Mistral AI, accesso eseguito il giorno novembre 28, 2025, https://mistral.ai/news/mistral-large-2407
34. Meta's New Llama 3.1 AI Model: Use Cases & Benchmark - Research AIMultiple, accesso eseguito il giorno novembre 28, 2025, https://research.aimultiple.com/meta-llama/
35. Ultimate Guide - The Best Open Source LLM For Italian In 2025 - SiliconFlow, accesso eseguito il giorno novembre 28, 2025, https://www.siliconflow.com/articles/en/best-open-source-LLM-for-Italian
36. How Well Does Llama 3.1 Perform for Text and Speech Translation? - Slator, accesso eseguito il giorno novembre 28, 2025, https://slator.com/how-well-does-llama-3-1-perform-for-text-speech-translation/
37. Italian LLM Models - a adrianoamalfi Collection - Hugging Face, accesso eseguito il giorno novembre 28, 2025, https://huggingface.co/collections/adrianoamalfi/italian-llm-models
38. Italian LLMs - a saiteki-kai Collection - Hugging Face, accesso eseguito il giorno novembre 28, 2025, https://huggingface.co/collections/saiteki-kai/italian-llms
39. The Best LLMs for AI Translation in 2025 - PoliLingua.com, accesso eseguito il giorno novembre 28, 2025, https://www.polilingua.com/blog/post/best-llm-ai-translation.htm
40. The Best LLMs for Translation: A Guide to Choosing the Right Model - Crowdin, accesso eseguito il giorno novembre 28, 2025, https://crowdin.com/blog/best-llms-for-translation