

AI ACADEMY

Applicare l'Intelligenza Artificiale nello sviluppo software

AI ACADEMY

LLM via API (multi-vendor) 20/06/2025

INTRODUZIONE DELL'ISTRUTTORE

Tamas Szakacs

Formazione

- Laureato come programmatore matematico
- MBA in management

Principali esperienze di lavoro

- Amministratore di sistemi UNIX
- Oracle DBA
- Sviluppatore di Java, Python e di Oracle PL/SQL
- Architetto (solution, enterprise, security, data)
- Ricercatore tecnologico e interdisciplinare di IA

Dedicato alla formazione continua

- Teorie, modelli, framework IA
- Ricerche IA
- Strategie aziendali
- Trasformazione digitale
- Formazione professionale

email: tamas.szakacs@proficegroup.it

MOTIVI E RIASSUNTO DEL CORSO

L'**Intelligenza Artificiale (AI)** è oggi il motore dell'innovazione in ogni settore, grazie alla sua capacità di analizzare dati, automatizzare processi e generare nuove soluzioni. Questo corso offre una panoramica completa e pratica sullo sviluppo di applicazioni AI moderne, guidando i partecipanti dall'ideazione al rilascio in produzione.

Attraverso una **combinazione di teoria chiara ed esercitazioni pratiche**, saranno affrontate le tecniche e gli strumenti più attuali: **machine learning, deep learning, reti neurali, Large Language Models (LLM), Transformers, Retrieval Augmented Generation (RAG)** e progettazione di agenti AI.

Le competenze acquisite saranno applicate in progetti concreti, dallo sviluppo di chatbot all'integrazione di modelli generativi, fino al deploy di soluzioni AI in ambienti reali e collaborativi.

Il percorso è pensato per chi vuole imparare a progettare, valutare e integrare sistemi AI di nuova generazione, con particolare attenzione alle best practice di programmazione, collaborazione in team, sicurezza, valutazione delle performance ed etica dell'AI.

DURATA: 17 GIORNI

OBIETTIVI

Il percorso formativo è progettato per **giovani consulenti junior**, con una conoscenza base di programmazione, che stanno iniziando un percorso professionale nel settore AI.

L'obiettivo centrale è fornire una panoramica pratica, completa e operativa sull'intelligenza artificiale moderna, guidando ogni partecipante attraverso tutte le fasi fondamentali.



OBIETTIVI

- Allineare conoscenze AI, ML, DL di tutti i partecipanti
- Saper usare e orchestrare modelli LLM (closed e open-weight)
- Costruire pipeline RAG complete (retrieval-augmented generation)
- Progettare agenti AI semplici con strumenti moderni (LangChain, tool calling)
- Capire principi di valutazione, robustezza e sicurezza dei sistemi GenA
- Migliorare la produttività come sviluppatori usando tool GenAI-driven
- Padroneggiare best practice di sviluppo, versioning e deploy AI
- Introdurre i fondamenti di Graph Data Science e Knowledge Graph
- Ottenere capacità di valutazione dei modelli e metriche
- Comprensione dell'etica e dei bias nei modelli di intelligenza artificiale
- Approfondire le normative di riferimento: AI Act, compliance e governance AI

Il corso è **estremamente pratico** (circa il 40% del tempo in esercitazioni hands-on, notebook, challenge e hackathon), con l'utilizzo di Google Colab, GitHub, e tutti gli strumenti necessari per lavorare su progetti reali e simulati.

STRUTTURA DELLE GIORNATE – PROGRAMMA BREVE

Tutte le giornate sono di 8 ore (9:00-17:00), con 1 ora di pausa suddivisa (mezz'ora pranzo, due pause da 15 min durante la mattina e il pomeriggio).

La progettazione sintetica delle giornate:

Giorno	Tema	Breve descrizione
1	Git & Python clean-code	Collaborazione su progetti reali, versionamento, codice pulito e testato
2	Machine Learning Supervised	Modelli supervisionati per predizione e classificazione
3	Machine Learning Unsupervised	Clustering, riduzione dimensionale, scoperta di pattern
4	Prompt Engineering avanzato	Scrivere e valutare prompt efficaci per modelli generativi
5	LLM via API (multi-vendor)	Uso pratico di modelli LLM via API, autenticazione, deployment
6	Come costruire un RAG	Pipeline end-to-end per Retrieval-Augmented Generation
7	Tool-calling & Agent design	Progettare agenti AI che usano strumenti esterni
8	Hackathon: Agentic RAG	Challenge pratica: chatbot agentic RAG in team

STRUTTURA DELLE GIORNATE – PROGRAMMA BREVE

Tutte le giornate sono di 8 ore (9:00-17:00), con 1 ora di pausa suddivisa (mezz'ora pranzo, due pause da 15 min durante la mattina e il pomeriggio).

La progettazione sintetica delle giornate:

Giorno	Tema	Breve descrizione
9	Hackathon: Rapid Prototyping	Da prototipo a web-app con Streamlit e GitHub
10	AI Productivity Tools	Workflow con IDE AI-powered, automazione e refactoring assistito
11	Docker & HF Spaces Deploy	Deployment di app GenAI containerizzate o su HuggingFace Spaces
12	AI Act & ISO 42001 Compliance	Fondamenti di compliance e governance AI
13	Knowledge Base & Graph Data Science	Introduzione a Knowledge Graph e query con Neo4j
14	Model evaluation & osservabilità	Metriche avanzate, explainability, strumenti di valutazione
15	AI bias, fairness ed etica applicata	Analisi dei rischi, metriche e mitigazione dei bias
16-17	Project Work & Challenge finale	Lavoro a gruppi, POC/POD, presentazione e votazione progetti

METODOLOGIA DEL CORSO

1. Approccio introduttivo ma avanzato

Il corso è introduttivo nei concetti base dell'AI applicata allo sviluppo, ma affronta anche tecnologie, modelli e soluzioni avanzate per garantire un apprendimento completo.

2. Linguaggio adattato

Il linguaggio utilizzato è chiaro e adattato agli studenti, con spiegazioni dettagliate dei termini tecnici per favorirne la comprensione e l'apprendimento graduale.

3. Esercizi pratici

Gli esercizi pratici sono interamente svolti online tramite piattaforme come Google Colab o notebook Python, eliminando la necessità di installare software sul proprio computer.

4. Supporto interattivo

È possibile porre domande in qualsiasi momento durante le lezioni o successivamente via email per garantire una piena comprensione del materiale trattato.

NOTA

Il corso segue un **approccio laboratoriale**: ogni giornata combina sessioni teoriche chiare e concrete con molte attività pratiche supervisionate, per sviluppare *competenze reali* immediatamente applicabili.

I partecipanti lavoreranno spesso in gruppo, useranno notebook in Colab e versioneranno codice su GitHub, vivendo una vera simulazione del lavoro in azienda AI.

Nessun prerequisito avanzato richiesto: si partirà dagli strumenti e flussi fondamentali, con una crescita graduale verso le tecniche più attuali e richieste dal mercato.

ORARIO TIPICO DELLE GIORNATE

Orario	Attività	Dettaglio
09:00 – 09:30	Teoria introduttiva	Concetti chiave, schema della giornata
09:30 – 10:30	Live coding + esercizio guidato	Esempio pratico, notebook Colab
10:30 – 10:45	<i>Pausa breve</i>	
10:45 – 11:30	Approfondimento teorico	Tecniche, best practice
11:30 – 12:30	Esercizio hands-on individuale	Sviluppo o completamento di codice
12:30 – 13:00	Discussione soluzioni + Q&A	Condivisione e correzione
13:00 – 14:00	<i>Pausa pranzo</i>	
13:30 – 14:15	Teoria avanzata / nuovi tools	Nuovi strumenti, pattern, demo
14:15 – 15:30	Esercizio a gruppi / challenge	Lavoro di squadra su task reale
15:30 – 15:45	<i>Pausa breve</i>	
15:45 – 16:30	Sommario teorico e pratico	
16:30 – 17:00	Discussioni, feedback	Riepilogo, best practice, domande aperte

DOMANDE?

Cominciamo!

OBIETTIVI DELLA GIORNATA

Obiettivi della giornata

- Presentazione dei modelli LLM.
- Capire come utilizzare modelli LLM (GPT, Llama, ecc.) tramite API, conoscendo le differenze tra fornitori (Azure, Hugging Face, ecc.).
- Saper configurare endpoint API per task diversi (chat, completions, embeddings), comprendendo parametri, limiti e flussi di autenticazione.
- Gestire chiavi, credenziali, tier di servizio e regioni, imparando a usare rate-limit, retry e streaming dove necessario.
- Impostare e testare chiamate batch e in streaming per casi d'uso reali.
- Analizzare e confrontare la facilità d'uso, i costi e le performance tra fornitori API principali (Azure OpenAI, Hugging Face).
- Costruire codici pratici in cui chiamare API di LLM per generazione testo, Q&A, embeddings e altri task, usando chiavi live in sicurezza.
- Prepararsi a implementare LLM nei flussi aziendali: conoscenza di limiti di sicurezza, privacy e best practice nell'uso di API.
- Saper diagnosticare e risolvere errori comuni nelle chiamate API.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Introduzione a Natural Language Processing (NLP)

Cosa fa l'NLP?

Il Natural Language Processing (NLP) si occupa di comprendere, generare e manipolare il linguaggio naturale. È applicato in traduzione automatica, analisi del sentiment, estrazione di informazioni, chatbot e molto altro. Utilizza tecniche di machine learning e deep learning per analizzare testo e parlato.

Come arriviamo all'NLP?

L'NLP si sviluppa a partire dai concetti fondamentali del deep learning, come reti convoluzionali (CNN) per analisi dei pattern e reti ricorrenti (RNN, LSTM) per dati sequenziali. Gli approcci moderni includono trasformazioni avanzate, come embedding per rappresentazioni numeriche del linguaggio, e l'uso dell'attenzione per identificare relazioni chiave nei dati.

Verso i Large Language Models (LLM)

I LLM, come ad es. Llama, BERT e GPT, sono basati su architetture avanzate come i Transformers. Essi utilizzano miliardi di parametri per elaborare grandi quantità di dati e risolvere problemi complessi di linguaggio. Il futuro degli LLM include modelli più efficienti, personalizzati e capaci di risolvere compiti complessi in tempo reale, come agenti.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Utilizzi tipici del Natural Language Processing (NLP)

Classificazione del Testo

- **Categorizzazione** di documenti o messaggi in base a temi o argomenti.
- Esempio: Filtrare email come spam o non spam.

Ricerca di Informazioni (Information Retrieval)

- **Recupero di informazioni** pertinenti da un database o da documenti testuali.
- Esempio: Trovare elementi specifici in un documento.

Riconoscimento di Entità Nominate (NER)

- **Estrazione di entità** significative come nomi, luoghi, date o organizzazioni da un testo.
- Esempio: Estrarre "Profice" come azienda e "Mantova" come luogo in un articolo.

Correzione Grammaticale e Ortografica

- **Identificazione e correzione di errori** nel testo.
- Esempio: Strumenti come Grammarly.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Utilizzi tipici del Natural Language Processing (NLP)

Analisi del Sentiment

- Identificazione del tono emotivo in testi come recensioni, post sui social media o feedback dei clienti.
- Esempio: Determinare se un commento è positivo, negativo o neutro.

Sommario Automatico

- Creazione di versioni brevi e concise di documenti più lunghi.
- Esempio: Generare un riassunto di un articolo di notizie.

Traduzione Automatica

- Conversione di testo da una lingua all'altra utilizzando modelli NLP.
- Esempio: Tradurre dall'italiano all'inglese.

Generazione del Linguaggio Naturale (NLG)

- Creazione automatica di testo comprensibile e coerente per chatbot, assistenti vocali e sistemi di risposta automatica.
- Esempio: Rispondere a una domanda in un chatbot.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

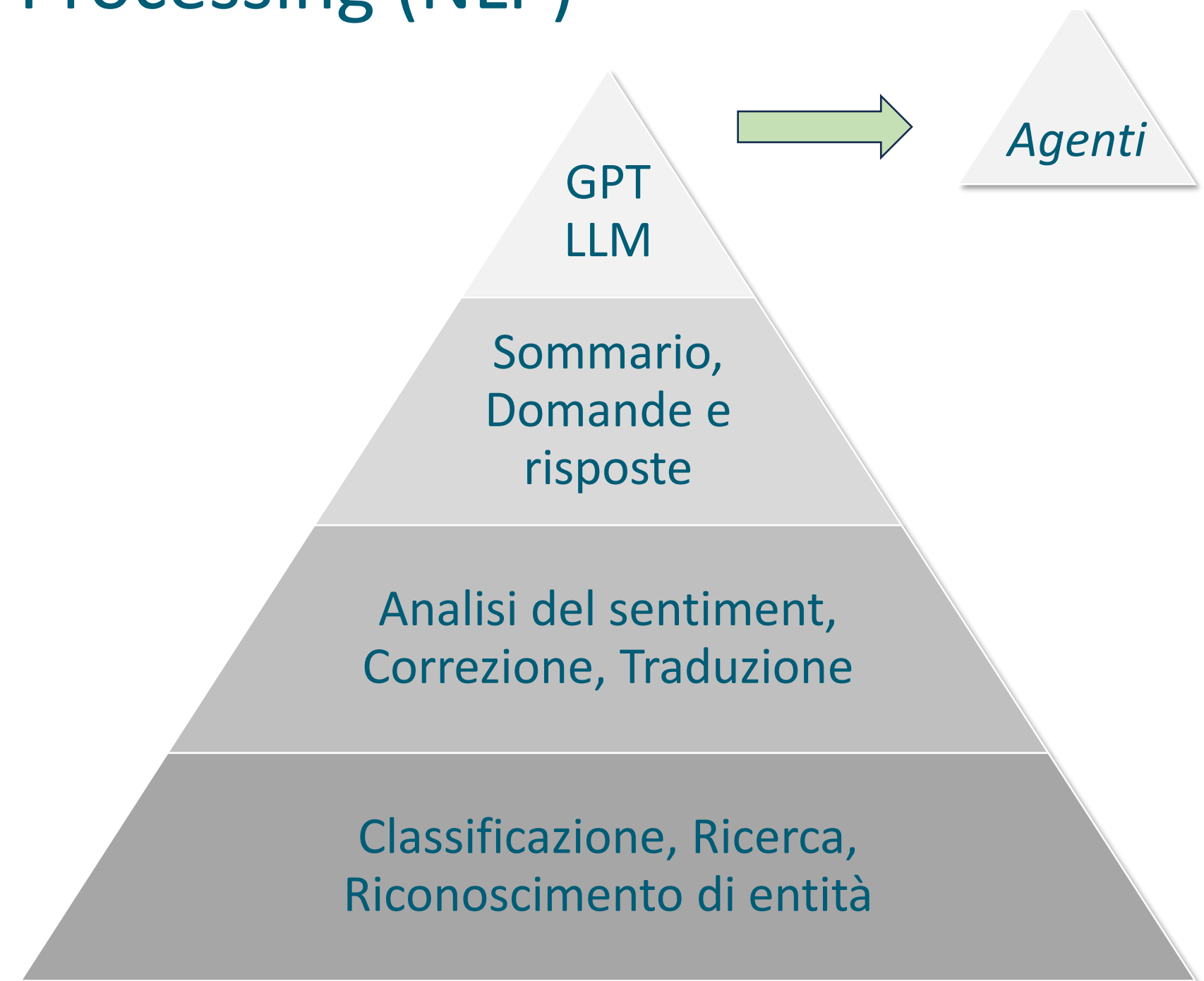
Utilizzi tipici del Natural Language Processing (NLP)

Domande e Risposte (Question Answering)

- Rispondere a domande specifiche basandosi su un contesto o una base di conoscenza.
- Esempio: Rispondere alla domanda "Chi è il CEO di Tesla?" leggendo da un articolo.

Chatbot e Assistenti Virtuali

- Automazione di conversazioni per servizio clienti, supporto tecnico o assistenza personale.
- Esempio: Siri, Alexa, ChatGPT.



ANALISI DOCUMENTALE E PROTEZIONE DATI

SmartDocs Srl – Analisi documentale e protezione dati

Scenario:

SmartDocs Srl, media azienda europea, deve gestire email e documenti contenenti dati sensibili di clienti (es: IBAN, codice fiscale, indirizzi, nomi, numeri di telefono).

L'azienda vuole automatizzare:

Estrazione di entità e dati sensibili (NER, pattern matching)

Riepilogo automatico e risposta alle richieste clienti

Tutelare la privacy (alcuni dati NON devono mai lasciare il server locale)

Minimizzare i costi cloud e garantire risposte rapide

ANALISI DOCUMENTALE E PROTEZIONE DATI

Useremo due modelli LLM per la soluzione aziendale

Architettura semplificata

1. Modello locale open source (es: TinyLLaMA)

1. Viene eseguito localmente, direttamente sul vostro computer o su server aziendali.
2. Si occupa delle operazioni più “sensibili”, come l'estrazione di dati personali e la protezione della privacy (Named Entity Recognition e masking dei dati).
3. È veloce, economico, e mantiene i dati riservati all'interno dell'azienda.

2. Modello cloud avanzato (es: Azure OpenAI GPT-3.5/4)

1. Viene utilizzato tramite API esterne, in cloud.
2. Si occupa di attività più complesse come il riepilogo automatico, l'analisi semantica e la generazione di risposte ai clienti.
3. Permette di gestire testi lunghi e offre maggiore potenza di calcolo e qualità delle risposte.

ANALISI DOCUMENTALE E PROTEZIONE DATI

L'obiettivo:

Sfruttare i **punti di forza di entrambi i modelli**:

- La privacy e la velocità del modello locale
- La potenza e la flessibilità del modello cloud

Garantire che i **dati più delicati non escano dall'azienda**, mentre sfruttiamo le migliori tecnologie disponibili per la produttività e l'automazione.

Distribuzione dei lavori

- Tutti lavorano sullo **stesso problema reale** ma con strumenti diversi.

Lavoro locale autonomo

- privacy, sicurezza, regex/NER, masking.

Lavoro cloud con l'aiuto dell'istruttore

- prompt, parametri, API, gestione delle risposte.

Altri componenti futuri per i giorni successivi

- RAG, contestualizzazione, configurazione, deployment ecc.
- La **collaborazione** tra i modelli con una architettura AI.

RUOLI DEI DUE MODELLI

1. Modello locale (TinyLLaMA o simili) — “Difensore/controllore”

Viene **eseguito localmente**.

Obiettivi:

- **NER (Named Entity Recognition)**: trova nomi, indirizzi, IBAN, codice fiscale, numeri, ecc.
- **Pattern Detection**: segnala se sono presenti dati sensibili o “red flag”.
- **RAG** (Retrieval Augmented Generation) opzionale: usa una knowledge base aziendale locale per arricchire le risposte.
- **Controlla l’output** prima che venga inviato al modello cloud, mascherando o anonimizzando i dati sensibili (es. sostituendo “Mario Rossi” con “[NOME]”).

2. Modello cloud (Azure GPT-3.5, GPT-4, ecc) — “Analista/colloquio”

Viene usato tramite API cloud, guidato passo-passo.

Obiettivi:

- Riceve documenti **già “ripuliti”** o parzialmente anonimizzati dal modello locale.
- Esegue:
 - Riepilogo (summary)
 - Analisi semantica
 - Generazione di risposte per i clienti
- Gestisce solo dati che non violano la privacy.

ESEMPIO DI PIPELINE COLLABORATIVA

Input:

L'utente carica un documento/email.

Passo 1 (locale):

Il modello locale fa NER, segnala dati sensibili e li anonimizza (es: "Mario Rossi" → "[NOME]", "IT60X0542811101000000123456" → "[IBAN]").

Passo 2 (controllo):

L'output ripulito viene controllato/validato (gli esperti possono anche vedere che i dati sono davvero rimossi).

Passo 3 (cloud):

Il testo anonimizzato viene inviato all'API Azure che fa il riepilogo, classifica la richiesta e prepara una risposta.

Passo 4 (output):

L'output finale può essere ricomposto localmente, reinserendo alcune entità dove permesso, oppure consegnato così.

SCEGLIERE IL PROVIDER E PROCURARE IL MODELLO

Per usare i LLM via API o localmente, dobbiamo **scegliere un provider** e individuare il modello adatto al nostro task (es: NER, generazione testo).

Provider principali e come scegliere

Azure OpenAI

Cos'è:

Servizio cloud di Microsoft, offre modelli GPT (es. GPT-3.5, GPT-4) via API.

Tiers (Livelli di servizio):

- **Trial/Free:** Crediti gratuiti iniziali per test e sviluppo.
- **Pay-as-you-go:** Paghi solo per quanto usi, adatto a uso flessibile.
- **Enterprise:** Per aziende con bisogni avanzati, offre SLA e supporto.

Come funziona:

Il modello resta in cloud, si accede tramite endpoint API, gestione centralizzata di sicurezza e billing.

SCEGLIERE IL PROVIDER E PROCURARE IL MODELLO

Hugging Face (HF)

Cos'è:

Community open source e servizio cloud con migliaia di modelli AI pronti, scaricabili o usabili tramite API.

Types (Categorie di modelli):

- **Text generation** (generazione testo)
- **NER** (Named Entity Recognition)
- **Summarization** (riepilogo)
- **Translation** (traduzione)
- **Classification** (classificazione)
- **Embeddings** (vettorizzazione)

Come funziona:

Si può:

- Scaricare e usare modelli **in locale** (privacy totale, serve hardware)
- Usare **endpoint cloud** (gratuito con limiti, poi piani a consumo)

SCEGLIERE IL PROVIDER E PROCURARE IL MODELLO

In sintesi (per qualsiasi progetto):

(Dopo aver analizzato il progetto, individuato scopi, disegnato architettura, ecc.) – fatto già per l'esercizio.

- Conosci i vari modelli
- Benchmark su dataset
- Scegli il provider (Azure, Hugging Face, ecc.)
- Cerca il modello giusto per il tuo compito (es: NER, text generation)
- Controlla il tier (Azure) o il type (HF) per capire costi, modalità di accesso e limiti.

CONCETTI UTILI PER L'IMPLEMENTAZIONE

Authentication

Cos'è:

Processo per identificarsi e autorizzare l'accesso alle API (token, chiave API).

Nota:

Ogni chiamata API necessita della chiave corretta; non condividere pubblicamente!

Rate Limit

Definizione:

Numero massimo di richieste API consentite in un intervallo di tempo (es: 60/min).

Superamento:

Porta a errori temporanei (es: 429 Too Many Requests).

Retry e Back-off

Retry:

Tentare nuovamente la richiesta dopo un errore temporaneo.

Back-off:

Aumentare progressivamente l'intervallo tra i retry (es: 1s, 2s, 4s...) per evitare "flooding" dell'API.

CONCETTI UTILI PER L'IMPLEMENTAZIONE

Endpoints

Definizione:

URL specifico per inviare richieste API al servizio desiderato (es: /v1/chat/completions).

Varia per modello e provider.

Endpoint Types

Esempi:

- Chat (conversazione)
- Completion (completamento testo)
- Embedding (vettorizzazione)

Autoscaling

Cos'è:

Capacità del provider cloud di allocare più risorse automaticamente in base al carico.

Vantaggio:

Performance costante anche con molti utenti.

CONCETTI UTILI PER L'IMPLEMENTAZIONE

Batch vs Streaming Completions

Batch:

Invio e ricezione di molte richieste tutte insieme; risposta finale unica.

Streaming:

Ricezione della risposta “a pezzi” man mano che viene generata (utile per risposte lunghe / conversazione).

Model Types

Esempi:

- **Text-generation:** GPT, LLaMA, Falcon, TinyLLaMA. Mistral.
- **NER:** modelli per Named Entity Recognition.
- **Classification, Summarization, Translation:** altri task AI.

Pipelines (HF Transformers)

Cos'è:

Metodo rapido per usare modelli preaddestrati con una sola funzione (es: `pipeline("ner")`, `pipeline("text-generation")`).

Vantaggi:

Semplifica l'uso, anche per principianti.

PERCORSO SU AZURE

Registrazione su Azure

1. Vai su <https://azure.microsoft.com/free/>
2. Clicca su **“Inizia gratis”**.
3. Registrati con la tua email personale e 2FA.
 - Servirà C.F. e **una carta di credito** per la verifica (**nessun addebito reale**).
 - Inserisci le informazioni richieste (nome, indirizzo, ecc.).
4. Attiva il tuo account Azure: avrai **\$200 di credito gratis** per 30 giorni.

Build in the cloud with an Azure account

Get started creating, deploying, and managing applications—across multiple clouds, on-premises, and at the edge—with scalable and cost-efficient Azure services.

Try Azure for free

Pay as you go

Welcome to Azure!

Don't have a subscription? Check out the following options.



Start with an Azure free trial

Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

Start

PERCORSO SU AZURE

Creazione della risorsa Azure OpenAI

1. Dopo il login, entra nel [portale Azure](#).
2. Clicca su **“Crea una risorsa”** (in alto a sinistra).
3. Cerca **“Azure OpenAI”** e seleziona la risorsa corrispondente.
4. Clicca su **“Crea”**.
 - Scegli **abbonamento** (Subscription) e **gruppo di risorse** (Resource Group) (puoi crearne uno nuovo).
 - Dai un nome alla risorsa.
 - Scegli la regione (consigliata: West Europe, North Europe).
 - Clicca su **“Rivedi e crea”**, poi ancora su **“Crea”**.
5. Attendi la creazione (1–2 minuti).

✓ Your deployment is complete

[Go to resource](#)

Azure services



Create a resource



Azure OpenAI

Create

Project Details

Subscription * ⓘ

Azure subscription 1

Resource group * ⓘ

[Create new](#)



Basics



Network



Tags



Review + submit



Configure network security for your Azure AI services resource.

Type *



All networks, including the internet, can access this resource.

PERCORSO SU AZURE

Ottenere il token API (“chiave”)

1. Vai alla **risorsa Azure OpenAI** che hai appena creato.
2. Nel menu a sinistra, in Gestione delle risorse clicca su **“Chiavi e endpoint” (Keys and Endpoint)**.
3. Troverai due “Key” (API key): **copia una** di queste chiavi — ti servirà per autenticare le chiamate API.

Ottenere l’Endpoint

- Nella stessa pagina di **Keys and Endpoint** vedrai il tuo **Endpoint** personale, che inizia per `https://...openai.azure.com/`
- **Copia e incolla** questo endpoint: serve nel codice per collegarti.



Azure OpenAI



T-AI1



Resource Management



Keys and Endpoint

KEY 1

.....

KEY 2

.....

Endpoint

`https://t-ai1.openai.azure.com/`

PERCORSO SU AZURE

Creare un Deployment Model

1. Nel menu della risorsa OpenAI, vai su “**Deployments**” (Deployment dei modelli) e **Deploy model**.
2. Clicca su “**Deploy model**”.
3. Scegli il modello (es: o4-mini).
4. Dai un nome unico al deployment (es: “chatgpt-demo”).
5. Conferma.

Il nome scelto sarà usato come engine nel codice Python!

Sei pronto!

- Ora hai:
 - **Token API** (la chiave)
 - **Endpoint**
 - **Nome del deployment**
- Puoi usarli subito nel codice Python (vedi risposta precedente).



Explore and deploy

Explore and deploy the generative AI models, craft unique prompts for your use cases, and fine-tune select models.

[Explore Azure AI Foundry portal](#)

+ Deploy model ▾

↻ Refresh

Deploy base model

Deploy fine-tuned model



o4-mini
Chat completion



Creating resource...

AI resource

PERCORSO SU AZURE

← o4-mini

Details

Metrics

Open in playground

Edit

Delete

Endpoint

Target URI

https://tcuoc-mc3vel13-eastus2.cogniti...

Key

.....

Language

Pyt...

SDK

Az...

Authentication type

Key Au...

Get Started

3. Run a basic code sample

This sample demonstrates a basic call to the chat completion API. The call is synchronous.

endpoint =

subscription_key =

```
raise self._make_status_error_from_response(err.response) from None
openai.NotFoundError: Error code: 404 - {'error': {'code': 'DeploymentNotFound', 'message': 'The API deployment for this resource does not exist. If you created the deployment within the last 5 minutes, please wait a moment and try again.'}}
```

PERCORSO SU AZURE

```
import openai

# Crea il client Azure OpenAI
client = openai.AzureOpenAI(
    api_key="LA_TUA_KEY", # <-- La tua chiave API di Azure OpenAI
    azure_endpoint="IL_TUO_ENDPOINT", # <-- Il tuo API endpoint di Azure OpenAI
    api_version="2024-12-01-preview", # <-- O la versione sul portale Azure
)

response = client.chat.completions.create(
    model="o4-mini", # <-- il nome esatto del deployment in Azure
    messages=[
        {"role": "system", "content": "Sei un assistente AI."},
        {"role": "user", "content": "Qual è la capitale dell'Italia?"}
    ],
    max_completion_tokens=256, # <-- Valore minimo per ottenere output
    temperature=1, # <-- Non puoi usare altro qui
)

print(response.choices[0].message.content)
```

```
>python G5E1_Azure_OpenAI.py
La capitale d'Italia è Roma.
```

PERCORSO SU HUGGING FACE

- **Registrati** gratis su Hugging Face.
- **Crea il token** (Settings > Access Tokens > New token > "Read" permission).
- **Copia il token** nel tuo codice.
- **Installa la libreria requests:**
`pip install requests`
- **Lancia il codice!** (Funziona anche su Colab)

DOMANDE?

PAUSA

DOMANDE?

Cominciamo il lavoro!

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Classificazione di testi

La classificazione del testo è il processo di categorizzazione di documenti, messaggi o frammenti di testo in base a temi, argomenti o classi predefinite. Questa tecnica è fondamentale in numerosi ambiti, come l'analisi delle opinioni, il filtraggio di contenuti indesiderati e l'organizzazione automatica delle informazioni.

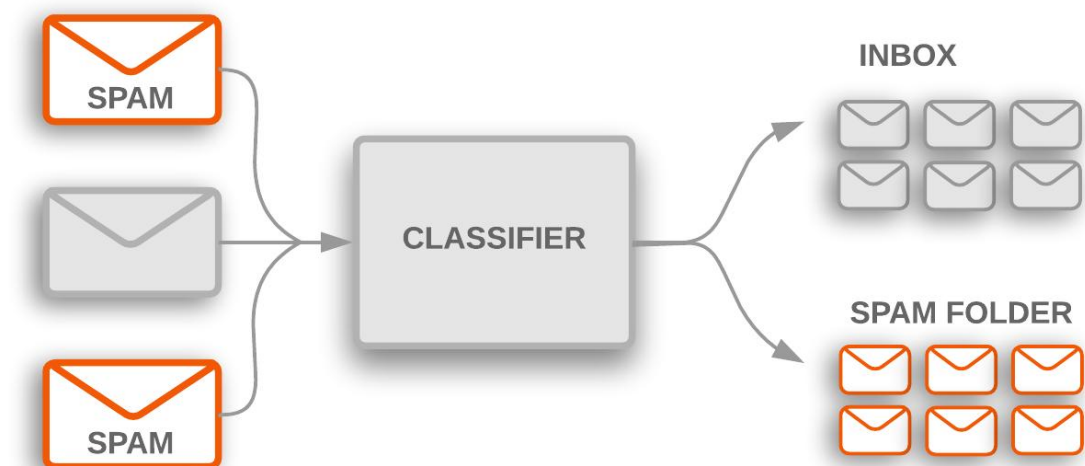
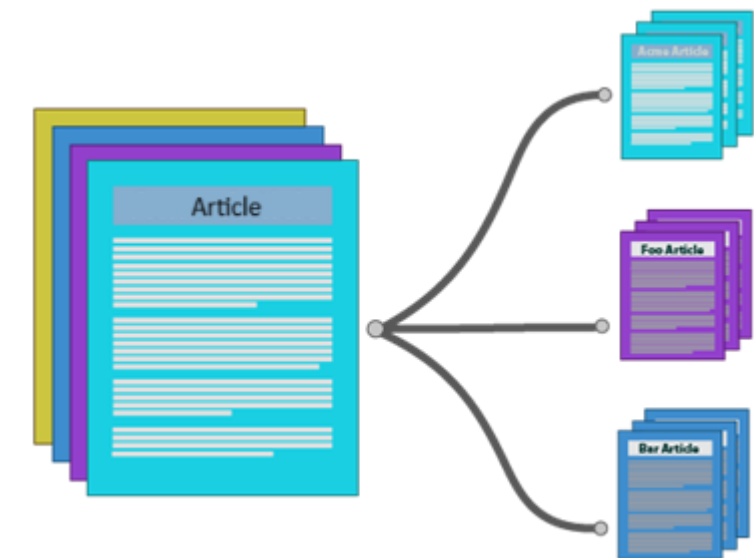
Esempi applicativi

Filtraggio Email: Classificare email come *spam* o *non spam* (ham).

Supporto clienti: Classificare messaggi in base al tipo di richiesta (tecnica, commerciale, amministrativa).

Analisi delle Opinioni:

Determinare se un messaggio è positivo, negativo o neutro.



INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Classificazione di testi

Come Funziona

1.Preprocessing:

- Rimuovere **stopword**, **punteggiatura** e **simboli**.
- Trasformare il testo in una **rappresentazione numerica** (es. embedding, Bag-of-Words o TF-IDF).

2.Modelli di Classificazione:

- Algoritmi tradizionali:
 - 1.Naive Bayes
 - 2.Support Vector Machines (SVM)
 - 3.Random Forest
- Modelli avanzati:
 - 1.Reti Neurali Artificiali (ANN)
 - 2.Modelli basati su Transformer (es. BERT, GPT).

3.Valutazione:

- Misurare precisione, richiamo, F1-score e accuratezza per valutare le prestazioni del modello.

Vantaggi

- **Scalabilità**: Gestisce grandi quantità di testo con alta velocità.
- **Flessibilità**: Applicabile a molti settori.
- **Personalizzazione**: Modelli personalizzati possono adattarsi a dati specifici.

Limiti

- Richiede un **dataset** di addestramento **di alta qualità**.
- Può essere **sensibile a variazioni linguistiche**, ambiguità e sarcasmo.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Ricerca di testi simili

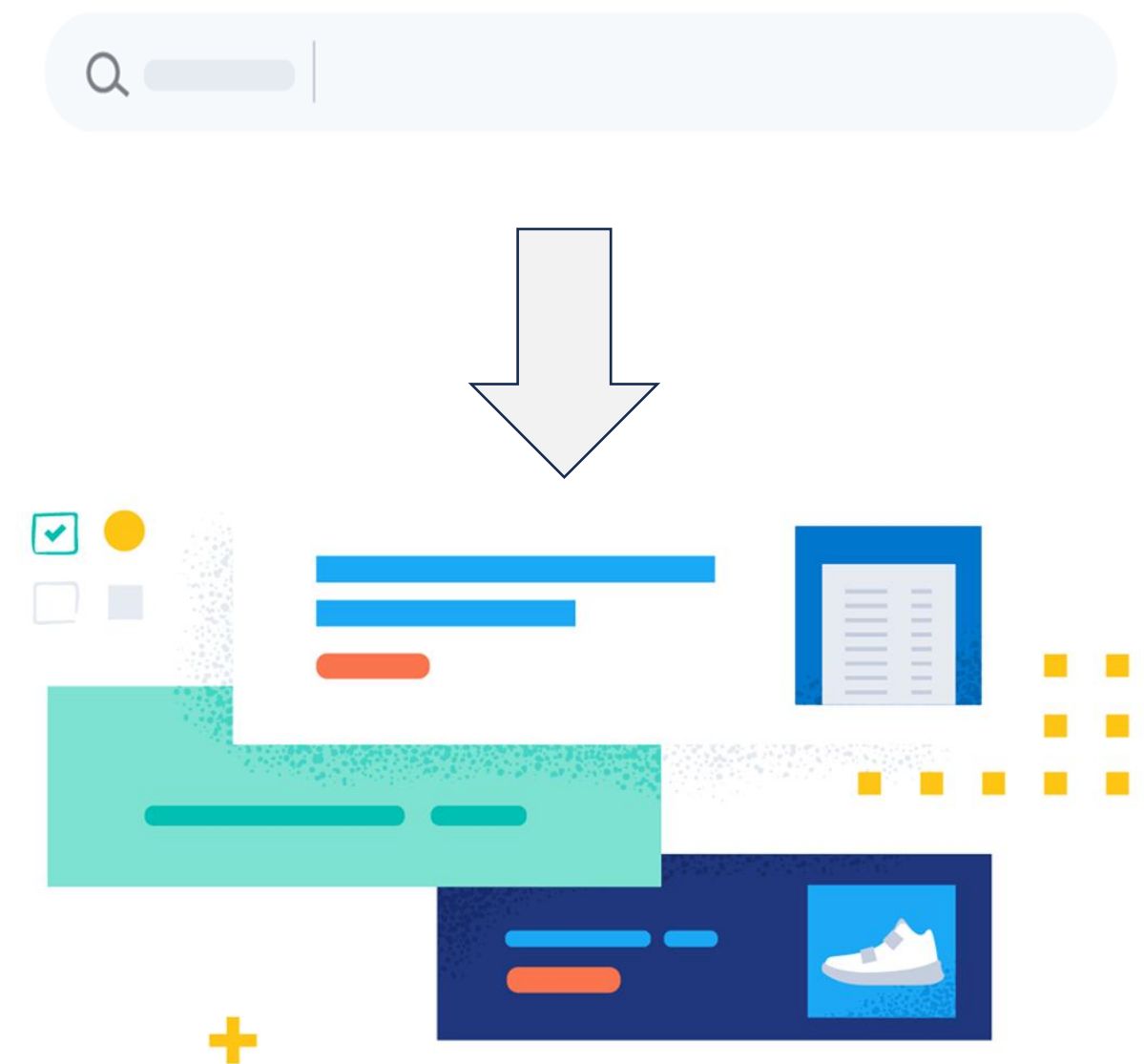
La ricerca di testo simile consiste nell'**identificare** testi che sono **semanticamente simili** a un dato tema o query. **Non si limita al matching esatto**, ma considera il significato e il contesto delle parole, permettendo di trovare testi anche con **termini correlati** o sinonimi.

Esempi applicativi

Motori di ricerca: Fornire risultati pertinenti anche con query parzialmente corrispondenti.

Raccomandazioni: Suggestire contenuti simili ad un testo tra i documenti aziendali.

Analisi dei documenti: Rilevare contenuti duplicati o simili in grandi archivi testuali.



INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Ricerca di testi simili

Come Funziona

1. Preprocessing del testo:

- Normalizzazione: Conversione in minuscolo, rimozione di stopwords e punteggiatura.
- Tokenizzazione: Suddivisione del testo in parole o frasi.

2. Rappresentazione semantica:

- TF-IDF: Valuta l'importanza delle parole in un testo rispetto a un corpus.
- Word Embeddings: Rappresenta parole in uno spazio vettoriale considerando il contesto (es. Word2Vec, GloVe).
- Sentence Embeddings: Rappresenta frasi o documenti interi in forma vettoriale (es. Sentence-BERT).

3. Misura della similarità:

- Calcolo della similarità tra i vettori:
 - Similarità Coseno: Valuta quanto due vettori puntano nella stessa direzione.
 - Distanza Euclidea: Misura la distanza tra i punti nello spazio vettoriale.

4. Classificazione e filtraggio:

- Impostare una soglia per identificare i risultati come "simili" o meno.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

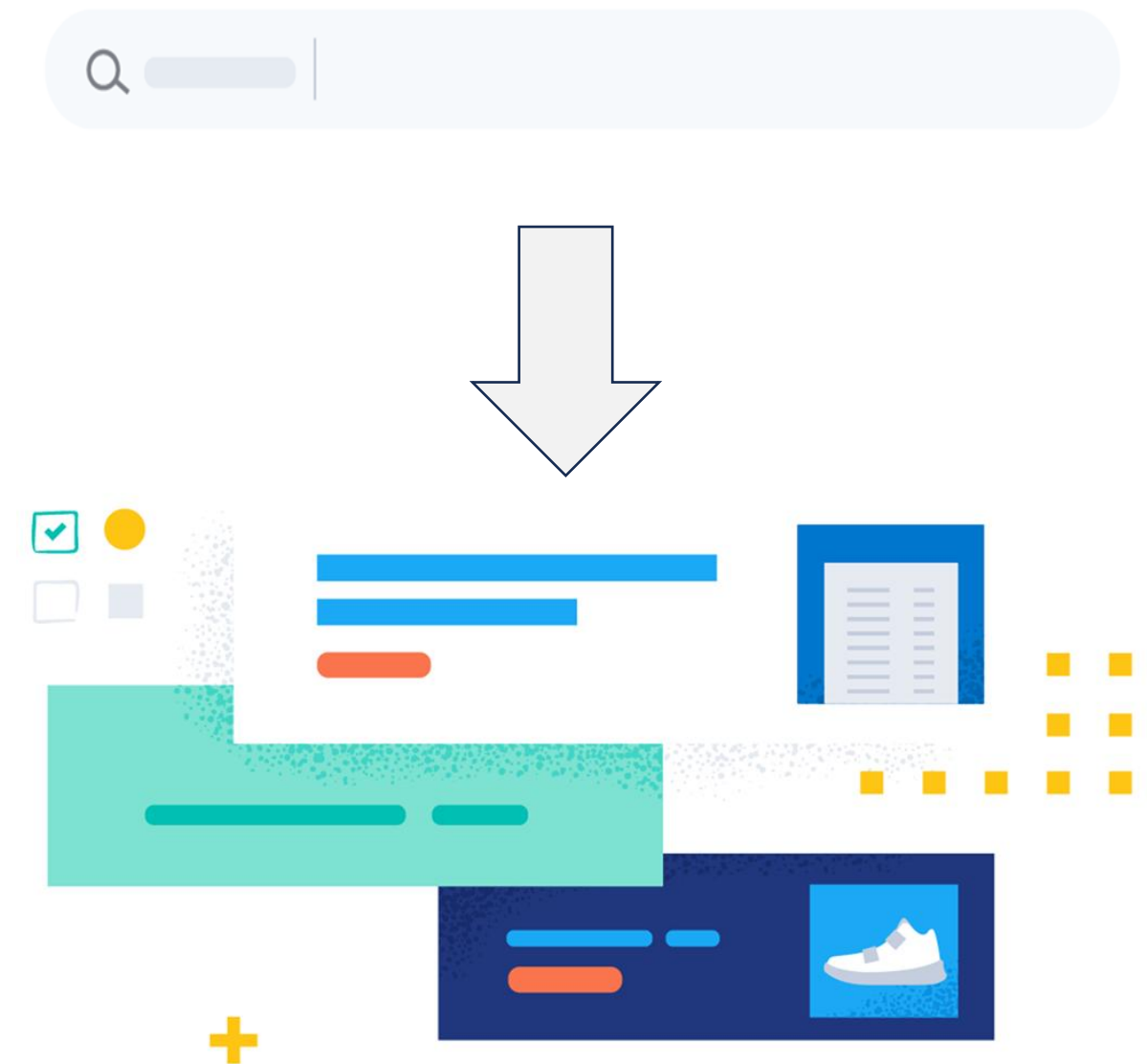
Ricerca di testi simili

Vantaggi

- **Ricerca più intelligente:** Trova risultati pertinenti anche senza corrispondenza esatta.
- **Maggiore precisione:** Consente di considerare il significato delle parole e non solo il loro matching.
- **Adattabilità:** Funziona bene con diverse lingue e domini.

Limiti

- **Dipendenza dal modello:** Le performance dipendono dalla qualità della rappresentazione semantica.
- **Ambiguità semantica:** Sinonimi o significati multipli possono confondere il modello.
- **Computazione intensiva:** L'elaborazione di grandi dataset richiede risorse.



INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Riconoscimento di Entità Nominate (NER)

Il riconoscimento di entità nominate (NER - Named Entity Recognition) è una tecnica di elaborazione del linguaggio naturale (NLP) usata per individuare e classificare automaticamente entità significative in un testo. Le entità possono includere nomi di persone, luoghi, organizzazioni, date, quantità, e altre categorie.

Come funziona

1.Preprocessing del Testo:

- **Normalizzazione:** Conversione in minuscolo, rimozione di stopwords.
- **Tokenizzazione:** Suddivisione del testo in parole o frasi.

2.Tecniche di NER:

- **Regole basate su pattern:**
 - Identificazione di entità usando espressioni regolari.
 - Esempio: Una data può essere trovata con il pattern `\d{2}/\d{2}/\d{4}`.
- **Modelli supervisionati:**
 - Algoritmi di machine learning addestrati su dataset etichettati (es. CRF, SVM).
- **Modelli basati su Deep Learning:**
 - Utilizzo di reti neurali come LSTM, GRU, o Transformer (es. BERT, GPT).
- **Dizionari e Lessici:**
 - Riconoscimento basato su liste predefinite di nomi, luoghi, ecc

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Riconoscimento di Entità Nominate (NER)

3. Classificazione delle Entità:

- Assegna a ciascun token o frase un'etichetta, come:
 - PERSON: Nome di persona.
 - ORG: Organizzazione.
 - LOC: Luogo geografico.
 - DATE: Data.
 - MISC: Entità varie non categorizzabili.

4. Validazione e pulizia:

- Rimuovere falsi positivi o risultati duplicati.

Risultato Atteso

Input: "Meeting con Giovanni a Sala 2 il 12 marzo."

Output: Giovanni: PERSON
Sala 2: LOC
il 12 marzo: DATE

Vantaggi

- **Automatizza la lettura del testo:** Rende i dati non strutturati più utilizzabili.
- **Riduce il lavoro manuale:** Identifica rapidamente entità in grandi dataset.
- **Alta adattabilità:** Può essere personalizzato per specifici domini.

Limiti

- **Dipendenza dal Dominio:** Richiede addestramento su dati specifici per ottenere buone performance.
- **Ambiguità:** Errori possono derivare da contesti poco chiari o ambiguità linguistica.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Correzione grammaticale e ortografica

La correzione grammaticale e ortografica è una tecnica di elaborazione del linguaggio naturale (NLP) utilizzata per individuare e correggere automaticamente errori di scrittura, ortografia e grammatica nei testi. Questo migliora la qualità del contenuto scritto e facilita la comprensione.

Come funziona

1. Preprocessing del testo:

- **Tokenizzazione:** Il testo è suddiviso in parole, frasi o paragrafi.
- **Normalizzazione:** Conversione in minuscolo, rimozione di caratteri non necessari.

2. Tecniche di correzione:

- **Dizionari Ortografici:**
 - Confronto delle parole con un dizionario predefinito.
 - Correzione suggerita basata sulle somiglianze.
- **Modelli linguistici statistici:** Riconoscono errori grammaticali basandosi sulla probabilità delle combinazioni di parole.
- **Modelli basati su Deep Learning:** Utilizzano reti neurali per comprendere il contesto e proporre correzioni (es. BERT, GPT).
- **Modelli di traduzione automatica:** Riformulano intere frasi correggendone struttura e grammatica.
- **Modelli di rete Bayesiana:** Identificano errori comuni e suggeriscono correzioni probabili.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Correzione grammaticale e ortografica

3. Classificazione degli errori:

- **Ortografia:** Parole mal scritte o inesistenti.
- **Grammatica:** Errori di accordo, tempi verbali o ordine delle parole.
- **Stile:** Frasi ridondanti o poco chiare.

4. Validazione e Feedback:

- Mostrare il testo originale con le correzioni suggerite.
- Permettere all'utente di accettare o rifiutare le modifiche.

Vantaggi

- **Accessibilità:** Migliora la qualità della scrittura, soprattutto per i non madrelingua.
- **Efficienza:** Riduce il tempo necessario per revisionare documenti.
- **Migliora la comunicazione:** Testi più chiari e grammaticalmente corretti.

Limiti

- **Dipendenza dal contesto:** Errori grammaticali contestuali possono non essere rilevati.
- **Falsi positivi:** Correzioni errate possono alterare il significato originale del testo.
- **Supporto linguistico limitato:** Alcune lingue o dialetti potrebbero non essere completamente supportati.

DOMANDE?

PAUSA

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Analisi del sentiment

L'analisi del sentiment è una tecnica di elaborazione del linguaggio naturale (NLP) che **identifica e classifica le emozioni o le opinioni** espresse in un testo. Viene utilizzata per determinare se il contenuto è **positivo, negativo o neutro**.

Come funziona

1. Preprocessing del testo:

- Rimuovere stop words, caratteri speciali e normalizzare il testo.

2. Tecniche di analisi:

- **Rule-based (Basato su regole):**

- Utilizza dizionari di parole positive e negative per calcolare il sentiment.
- Esempio: "buono", "fantastico" (+), "terribile", "scadente" (-).

- **Machine Learning:**

- Modelli supervisionati (es. **Naive Bayes, Random Forest**).
- Richiede un **dataset etichettato** per addestrare il modello.

- **Deep Learning:**

- Utilizza reti neurali come **LSTM, RNN o Transformer** per analizzare il contesto e il tono.

- **Approcci ibridi:**

- Combina regole e modelli di apprendimento per migliorare l'accuratezza.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Analisi del sentiment

3. Output:

- Classificazione Binaria: **Positivo/Negativo**.
- Classificazione Multiclasse:
Positivo, Negativo, Neutro.
- Punteggio Continuo:
Da -1 (n Negativo) a +1 (positivo).

Strumenti utilizzati

- **TextBlob**: Libreria semplice per analisi base del sentiment.
- **VADER** (Valence Aware Dictionary and sEntiment Reasoner): Ideale per testi brevi come tweet.
- **Transformers** (BERT, GPT): Analisi del sentiment avanzata con contesto.

Vantaggi

- **Automazione**: Analizza grandi quantità di dati in poco tempo.
- **Insight emotivi**: Identifica tendenze e reazioni pubbliche.
- **Personalizzazione**: Migliora l'esperienza utente basandosi sui feedback.

Limiti

- **Ambiguità del linguaggio**: Ironia e sarcasmo possono essere difficili da rilevare.
- **Dipendenza dal contesto**: Il sentiment può variare in base alla situazione.
- **Bias nei dataset**: I modelli possono riflettere pregiudizi nei dati di addestramento.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Sommario automatico

Il sommario automatico è una tecnica di elaborazione del linguaggio naturale (NLP) utilizzata per **estrarre o generare una versione ridotta di un testo**, mantenendo le informazioni chiave. Può essere **estrattivo** (seleziona frasi dal testo originale) o **astrattivo** (genera nuove frasi).

Come funziona

1. Preprocessing del testo:

- Pulizia del testo (rimozione di simboli, numeri non necessari, ecc.).
- **Tokenizzazione** e analisi grammaticale.

2. Tipi di sommario:

•Estrattivo:

- **Identifica le frasi più importanti** basandosi su punteggi di rilevanza.
- Utilizza metodi come **TF-IDF** o **grafi** (es. algoritmo TextRank).

•Astrattivo:

- **Genera una sintesi** interpretando il significato del testo.
- Utilizza modelli di **deep learning** (es. LSTM, Transformer).

3. Output:

- Un **sommario sintetico** che mantiene il contesto e le informazioni chiave.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Sommario automatico

Vantaggi

- **Efficienza:** Riduce significativamente il tempo di lettura.
- **Personalizzazione:** I sommari possono essere adattati al tipo di utente.
- **Scalabilità:** Ideale per elaborare grandi quantità di testo.

Limiti

- **Perdita di dettagli:** Alcune informazioni rilevanti possono essere omesse.
- **Qualità variabile:** I sommari astrattivi possono contenere errori o non mantenere il contesto.
- **Bias del modello:** I risultati possono riflettere pregiudizi nei dati di addestramento.

Strumenti utilizzati

1.NLTK + TextRank:

- Algoritmi basati su **grafi** per estrarre frasi importanti.

2.BART e T5:

- Modelli di **deep learning** per il sommario astrattivo.

3.Gensim:

- Libreria Python per il **sommario estrattivo** semplice.

4.Hugging Face Transformers:

- Sommari astrattivi avanzati con modelli come Pegasus.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Traduzione automatica

La traduzione automatica è il processo di **trasformazione di un testo scritto** in una lingua (lingua di origine) in un testo equivalente in **un'altra lingua** (lingua di destinazione). Questo è **uno dei compiti fondamentali** del Natural Language Processing (NLP) ed è ampiamente utilizzato in applicazioni come traduttori online, assistenti virtuali e software multilingue.

Come funziona

1. Pre-processing del testo

- **Tokenizzazione**: Il testo viene **suddiviso in unità più piccole**, come parole o frasi.
- **Normalizzazione**: Rimozione di caratteri speciali, conversione a lettere minuscole, espansione delle abbreviazioni, ecc.

- **Analisi morfologica**: Identificazione di radici, suffissi e prefissi.

2. Conversione linguistica

- **Riconoscimento grammaticale**: Analisi della struttura grammaticale del testo di origine.
- **Mappatura semantica**: Traduzione dei significati delle parole o frasi in una rappresentazione intermedia (modelli rule-based o embedding neurali).

3. Generazione del testo tradotto

- **Ricostruzione grammaticale**: Applicazione delle regole della lingua di destinazione.
- **Adattamento del contesto**: Scelta delle traduzioni più appropriate in base al significato e al contesto.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Traduzione automatica

4. Post-processing

- **Correzione grammaticale**: Verifica della correttezza sintattica e grammaticale.
- **Adattamento culturale**: Modifiche per rispettare le differenze culturali tra le lingue.

Metodi principali

1. Rule-based Machine Translation (RBMT)

- Utilizza **regole grammaticali predefinite** e **dizionari bilingue**.
- Funziona bene per linguaggi con strutture rigide, ma richiede molta manutenzione manuale.
- Esempio: Traduttori pionieristici come SYSTRAN.

2. Statistical Machine Translation (SMT)

- Basato su probabilità calcolate da corpora di traduzioni parallele.
- Analizza le frequenze con cui certe parole o frasi si traducono in altre.
- Esempio: Versioni precedenti di Google Translate.

3. Neural Machine Translation (NMT)

- Basato su **reti neurali profonde** (spesso con architetture come i trasformatori).
- Crea **rappresentazioni vettoriali (embedding)** del testo e genera traduzioni fluide e contestuali.
- Utilizza modelli **seq2seq** con meccanismi di **attenzione** per "focalizzarsi" su parti rilevanti del testo.
- Esempio: Google Translate, DeepL.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Traduzione automatica

Vantaggi

- **Accessibilità globale:** Permette alle persone di comunicare indipendentemente dalla lingua.
- **Efficienza:** Traduzioni rapide rispetto a quelle manuali.
- **Supporto multilingue:** Utilizzato in servizi clienti e app internazionali.

Limitazioni

- **Qualità variabile** a seconda della complessità del testo e delle lingue.
- Non sempre capace di **catturare il contesto** culturale o le sfumature.
- I traduttori basati su regole o statistici possono risultare **meno accurati** rispetto ai modelli neurali.

Sfide

- **Ambiguità:** Parole con **significati multipli** possono essere tradotte in modo errato.
- **Strutture grammaticali diverse:** Lingue come il giapponese o l'ungherese possono presentare difficoltà.
- **Cultura e Contesto:** Adattare la traduzione alle differenze culturali è complesso.

Prospettive future

- Maggiore comprensione semantica e contestuale.
- Traduzione vocale in tempo reale.
- **Sistemi più leggeri** per dispositivi mobili e connettività limitata.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Tecniche usate in NLP

Word Embeddings:

Rappresentazioni vettoriali delle parole che catturano relazioni semantiche e sintattiche.

Esempi:

- **Word2Vec**: Modello che mappa le parole in uno spazio continuo, rappresentando parole con contesti simili con vettori vicini.
- **GloVe**: Modello che cattura statistiche globali del corpus per generare embedding.
- **FastText**: Espande Word2Vec includendo sub-parole, utile per lingue con morfologia complessa.

Ruolo: Migliora le rappresentazioni rispetto ai metodi basati su TF-IDF o Bag of Words.

Seq2Seq (Sequence-to-Sequence):

Architettura per convertire una sequenza in un'altra (e.g., traduzione, riassunto).

Componente chiave: Encoder-decoder.

- **Encoder**: Crea una rappresentazione compatta della sequenza in input.
- **Decoder**: Genera la sequenza di output dalla rappresentazione dell'encoder.

Aggiunta del meccanismo di attenzione per gestire sequenze lunghe.

Esempi: Utilizzato per traduzioni, chatbot, e text-to-speech.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Tecniche usate in NLP

L'Embedding

L'embedding è una tecnica che mappa parole, frasi o interi documenti in rappresentazioni vettoriali numeriche in uno spazio continuo a bassa dimensionalità. Questo permette ai modelli di lavorare con testo, immagini o dati categoriali trasformandoli in input numerici.

La sua importanza in NLP

- **Cattura relazioni semantiche:** Parole con significati simili sono rappresentate con vettori vicini nello spazio.
- **Esempio:** re - uomo + donna \approx regina.
- **Riduce la dimensionalità:** Converte dati sparsi come Bag of Words in vettori densi e compatti.
- **Migliora le performance dei modelli:** Fornisce input più informativi e adatti a modelli avanzati.

Applicazioni

- **Ricerca e recupero:** Trovare documenti simili in grandi dataset.
- **Classificazione del testo:** E.g., categorizzare email o recensioni.
- **Riconoscimento di Entità Nominate (NER):** Miglioramento delle prestazioni nei modelli di riconoscimento.
- **Analisi del Sentiment:** Comprendere opinioni espresse in testo.
- **Sistemi di raccomandazione:** Trovare contenuti correlati basati su preferenze.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Sistemi di Raccomandazione

I sistemi di raccomandazione sono algoritmi progettati per suggerire contenuti o prodotti agli utenti, basandosi su vari tipi di dati raccolti, come preferenze, comportamento passato o caratteristiche del prodotto.

Benefici

- **Personalizzazione**: Esperienze utente su misura.
- **Engagement**: Maggiore tempo trascorso sulle piattaforme.
- **Aumento delle vendite**: Incremento del cross-selling e up-selling.

Un sistema di raccomandazione ben progettato migliora l'esperienza dell'utente e crea valore per le aziende.

Esempi di applicazione

- **E-commerce**: Suggerimenti di prodotti simili o complementari (es. Amazon).
- **Streaming**: Raccomandazioni di film, serie TV o musica (es. Netflix, Spotify).
- **Social Media**: Consigli su chi seguire o contenuti da visualizzare (es. Instagram, YouTube).
- **E-learning**: Suggerimenti di corsi o materiali di studio personalizzati (es. Udemy, edx, Coursera).

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

Sistemi di Raccomandazione

Tipologie

1. Collaborative Filtering (Filtraggio Collaborativo)

- Basato sulle interazioni degli utenti con i prodotti.
- **Esempio:** "Utenti simili a te hanno acquistato anche..."
- Funziona bene con grandi quantità di dati.
- Problema del "cold start" (difficoltà iniziali per nuovi utenti o prodotti).

2. Content-Based Filtering (Filtraggio Basato sui Contenuti)

- Analizza le caratteristiche dei prodotti per raccomandarne di simili.
- **Esempio:** "Hai guardato un film d'azione, ti potrebbero piacere questi altri film d'azione."
- Non richiede dati sugli altri utenti.
- Limita la scoperta di contenuti nuovi o diversi.

3. Hybrid Systems (Sistemi Ibridi)

- Combinano filtraggio collaborativo e basato sui contenuti.
- **Esempio:** Netflix utilizza entrambi i metodi per raccomandare film e serie TV.
- Maggiore accuratezza e capacità di superare i limiti delle singole tecniche.

INTRODUZIONE A NATURAL LANGUAGE PROCESSING

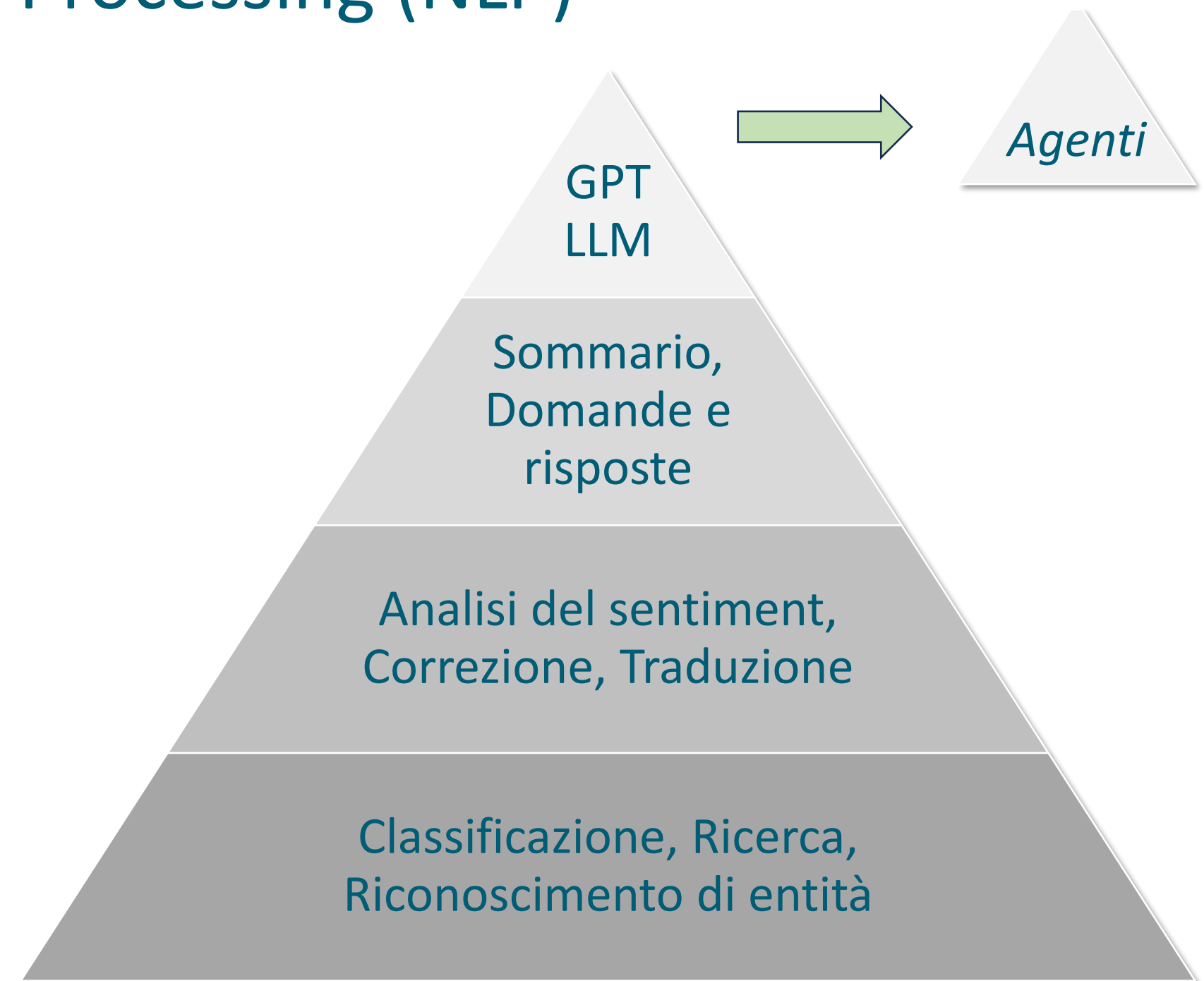
Utilizzi tipici del Natural Language Processing (NLP)

Domande e Risposte (Question Answering)

- Rispondere a domande specifiche basandosi su un contesto o una base di conoscenza.
- Esempio: Rispondere alla domanda "Chi è il CEO di Tesla?" leggendo da un articolo.

Chatbot e Assistenti Virtuali

- Automazione di conversazioni per servizio clienti, supporto tecnico o assistenza personale.
- Esempio: Siri, Alexa, ChatGPT.



GRAZIE PER L'ATTENZIONE