Un Dizionario Concettuale Ottimizzato per la Lingua Italiana: Approccio Modulare e Tecniche dal Gaming per l'AI

Samuele Scuglia

18 maggio 2025

Sommario

Presentiamo una pipeline automatica ispirata all'ingegneria dei videogiochi per la costruzione di un dizionario concettuale italiano compatto, privo di ridondanze e ottimizzato per applicazioni in intelligenza artificiale e NLP. Mostriamo le tecniche di parsing, pulizia e consolidamento che portano il lessico italiano da centinaia di migliaia di forme a circa 32.000 concetti semantici unici, e discutiamo le applicazioni concrete per l'AI, la traduzione e la ricerca semantica.

Indice

1 Introduzione

L'elaborazione del linguaggio naturale (NLP) e le applicazioni di intelligenza artificiale linguistica si sono storicamente basate su grandi risorse lessicali come WordNet e Wiktionary. Questi database organizzano il lessico in reti di sinonimi e relazioni semantiche, ma presentano spesso ridondanze, cluster sovrapposti e una gestione subottimale dei sinonimi, causando inefficienza sia nell'analisi semantica che nel training di modelli AI.

In questo lavoro propongo un nuovo approccio alla costruzione di un dizionario concettuale italiano: una pipeline completamente automatica, ispirata alle strategie di data management usate nell'industria dei videogiochi per gestire big data modulari (es. chunking, salvataggi atomici, JSON indexing). Grazie a questa pipeline, l'intero lessico italiano viene ridotto a circa 32.000 concetti unici, ciascuno rappresentato solo da sinonimi "perfetti" — ovvero gruppi privi di rumore semantico.

Questa metodologia, applicabile a qualunque lingua e dominio, garantisce accesso istantaneo ai dati, massima granularità concettuale e un'infrastruttura finalmente adatta sia per il training di LLM italiani che per applicazioni pratiche (ricerca semantica, traduzione automatica, analisi del testo). Il lavoro, inoltre, rappresenta un invito a contaminare la linguistica computazionale con le buone pratiche nate nell'industria gaming: modularità, scalabilità e patchabilità, per superare i limiti dei dizionari legacy e aprire la strada a una nuova generazione di AI "native italiane".

2 Background

2.1 Dizionari computazionali: WordNet, Wiktionary e limiti strutturali

WordNet è stato il primo database lessicale di riferimento per la linguistica computazionale, organizzando il lessico inglese in gruppi di sinonimi (synset) e mappando tra loro relazioni semantiche quali iperonimia, iponimia e antonimia. Tuttavia, la struttura stessa di WordNet – seppur efficace per l'inglese standard – presenta limiti nel momento in cui si tenta di applicarla ad altre lingue o a task NLP avanzati: duplicazione di concetti, sinonimi "sporchi", cluster sovrapposti, eccessiva granularità.

Wiktionary, il dizionario collaborativo open-source, offre una copertura molto più ampia e flessibile: ogni lemma può contenere gruppi di sinonimi, varianti, significati storici e tecnicismi. Ma anche qui la struttura "umana" genera rumore, duplicati e relazioni semantiche non sempre rigorose.

Open Multilingual WordNet, BabelNet e risorse simili hanno tentato l'estensione multilingue, ma restano ancorati a una logica di clusterizzazione statica, poco adatta a tokenizzazione concettuale per modelli AI realmente interpretabili e compatti.

2.2 Big Data e modularità nel gaming: una lezione ignorata

L'industria dei videogiochi – soprattutto titoli open world e piattaforme modding come Minecraft, Skyrim, GTA, The Witcher – ha dovuto affrontare (e risolvere) il problema della gestione di decine o centinaia di gigabyte di dati modulari, patchabili, chunkabili e facilmente aggiornabili.

Grazie a tecniche come il salvataggio a chunk, il caricamento dinamico on-demand, il data indexing su formati serializzati (JSON, NBT, YAML), e la separazione netta tra logica, asset e metadati, oggi i motori di gioco possono offrire esperienze "infinite" senza mai collassare per limiti di performance o incompatibilità tra versioni.

Queste stesse strategie – basate su modularità, accesso atomico e processi a layer – possono essere applicate con enorme efficacia anche al data engineering per l'AI e la linguistica computazionale, come dimostrato in questo lavoro.

3 Metodologia

3.1 Pipeline a layer: dal dump Wiktionary al dizionario concettuale

La pipeline proposta si compone di una serie di task indipendenti e sequenziali, ispirati all'approccio "a layer" tipico dell'ingegneria dati nei videogiochi. Ogni layer svolge una funzione precisa (pulizia, normalizzazione, clustering, consolidamento), riducendo la complessità computazionale e facilitando debugging e ottimizzazione.

3.2 Parsing e pulizia iniziale

Il primo step consiste nell'estrazione sistematica dei gruppi di sinonimi dal dump XML di Wiktionary italiano. Un parser automatico, scritto in Python, isola le sezioni di interesse (tipicamente marcate da `{{-sin-}}` e delimitatori strutturali) e le converte in una lista JSON, dove ogni entry rappresenta un gruppo di potenziali sinonimi.

Segue una fase di pulizia:

- Rimozione di caratteri speciali, varianti ortografiche, note inutili e tag HTML/Wiki.
- Normalizzazione di maiuscole/minuscole, uniformazione delle desinenze e delle forme flesse.
- Eliminazione di duplicati intra-gruppo e tra gruppi.

3.3 Accoppiamento e consolidamento: sinonimi perfetti

Il cuore della metodologia è il consolidamento intelligente dei gruppi di sinonimi. Per ogni gruppo JSON, si cerca la massima sovrapposizione con altri gruppi: tutti i gruppi che differiscono solo per una parola vengono unificati in un unico "concetto" se la differenza non implica una perdita semantica evidente (criterio basato su matching a insiemi e verifica manuale di outlier).

Questa procedura, ripetuta su tutto il dump, riduce drasticamente la ridondanza: da oltre 100.000 cluster iniziali si scende a circa 32.000 concetti unici, ciascuno rappresentato da un insieme di sinonimi "puri" e completamente intercambiabili.

3.4 Serializzazione finale e accesso istantaneo

Il dizionario così ottenuto viene serializzato in JSON, permettendo lookup istantanei e manipolazione su milioni di entry anche senza l'uso di database pesanti. Questa scelta di formato garantisce massima portabilità e riusabilità, sia per training di AI che per integrazione in motori di ricerca semantici o traduttori concettuali.

3.5 Integrazione con Morph-it

Successivamente alla creazione dei gruppi di sinonimi, ogni lemma principale è stato arricchito con tutte le sue forme flesse, grazie all'integrazione automatica del lemmario **Morph-it**, risorsa morfologica open source per l'italiano. Attraverso un

processo di matching diretto tra lemma e forma, sono state aggiunte coniugazioni verbali, declinazioni, plurali e varianti grammaticali che permettono una copertura completa delle forme utilizzabili nel linguaggio naturale.

Questo passaggio ha trasformato ciascun concetto da semplice rappresentazione semantica a struttura completa, pronta per task di lemmatizzazione inversa, tokenizzazione concettuale o generazione linguistica a partire da simboli astratti. Il dizionario risultante è così non solo semanticamente disambiguato, ma anche morfologicamente esaustivo.

3.6 Nota sugli script

Per completezza, si segnala che gli script originali utilizzati per la parserizzazione e il consolidamento sono stati sviluppati come prototipi temporanei e non sono più disponibili, in quanto eliminati dopo la generazione del dataset finale. Tuttavia, l'intero workflow può essere facilmente replicato grazie alla semplicità degli algoritmi adottati (parsing lineare, operazioni su insiemi, deduplicazione) e ai dettagli forniti in questo lavoro. Un esempio di pseudocodice è fornito in appendice per facilitare eventuali riproduzioni o miglioramenti futuri.

4 Risultati

4.1 Statistiche quantitative

L'applicazione della pipeline ha portato a una riduzione drastica della ridondanza semantica. A partire da oltre 100.000 gruppi di sinonimi estratti dal dump di Wiktionary italiano, il processo di consolidamento ha prodotto circa 32.000 concetti unici, ciascuno rappresentato da un insieme di sinonimi "perfetti".

- **Numero di parole uniche (lemmi):** $\sim$ 120.000
- **Gruppi iniziali (pre-pulizia):** $\sim$ 100.000
- **Gruppi finali (post-merge):** 32.000
- **Sinonimi medi per gruppo:** 3.2
- **Tempo di parsing e consolidamento:** $\sim$ 15 minuti su hardware consumer

4.2 Confronto con WordNet e altre risorse

A titolo di confronto, WordNet (versione 3.0, inglese) contiene circa 117.000 synset, spesso sovrapposti e non sempre composti da sinonimi realmente perfetti. Il dizionario qui presentato offre una copertura semantica totale della lingua italiana con una granularità notevolmente superiore, grazie all'eliminazione sistematica dei duplicati e all'uso di gruppi di sinonimi "puri".

4.3 Esempi pratici

Esempio 1 (gruppo consolidato):

```
["abbandonare", "lasciare", "desistere", "rinunciare"]
```

Esempio 2 (differenza rispetto a WordNet):
WordNet: *to abandon*: ["abandon", "forsake", "leave behind", ...] *to relinquish*: ["relinquish", "give up", "release", ...]
Nel dizionario concettuale:

```
["abbandonare", "lasciare", "rinunciare"]
```

4.4 Performance e scalabilità

L'intera pipeline – dalla parserizzazione del dump Wiktionary alla produzione del file

5 Applicazioni

Il dizionario concettuale ottimizzato offre una piattaforma versatile per una vasta gamma di applicazioni pratiche, sia nell'ambito dell'intelligenza artificiale che in altri settori legati al linguaggio.

5.1 Tokenizzazione concettuale per LLM e AI

La rappresentazione compatta e non ridondante dei concetti semantici permette di addestrare modelli linguistici di grandi dimensioni (LLM) utilizzando un vocabolario di "token concettuali" invece di semplici parole o caratteri. Ciò migliora sia la comprensione semantica che l'efficienza, facilitando anche il transfer learning tra lingue diverse o domini specialistici.

5.2 Motori di ricerca semantici

Integrando il dizionario in un motore di ricerca, è possibile superare la ricerca per parola-chiave a favore di una ricerca per concetto: l'utente trova informazioni anche usando sinonimi rari, tecnicismi o regionalismi, grazie all'unificazione semantica dei gruppi.

5.3 Traduzione concettuale multilingua

Grazie alla struttura "neutrale" del dizionario, è possibile mappare concetti italiani su quelli di altre lingue (ad es. usando mapping tra cluster di sinonimi perfetti in italiano e inglese/francese/spagnolo). Questo approccio riduce drasticamente la perdita di significato e le ambiguità tipiche della traduzione automatica basata su singole parole.

5.4 Parsing e analisi testuale

L'analisi di testi può essere effettuata su base concettuale, identificando automaticamente sinonimie, parafrasi e relazioni tra frasi anche molto diverse a livello lessicale ma equivalenti sul piano del significato.

5.5 Didattica e strumenti linguistici

Docenti e studenti possono sfruttare il dizionario per esercizi di comprensione, produzione di sinonimi, parafrasi, arricchimento lessicale e giochi linguistici (ad esempio, generazione automatica di quiz o test di "parola mancante" basati su cluster di sinonimi).

5.6 Versatilità per moduli AI specializzati

Grazie alla modularità, il dizionario può essere usato come base per moduli grammaticali (declinazione, accordo), frasificatori, generatori di testo controllato e AI

"spiegatrici" che possono ricostruire il percorso concettuale di una risposta o una traduzione.

5.7 Espandibilità: altre lingue e domini

Il workflow adottato, basato su parsing, pulizia e clustering, è replicabile facilmente su altri Wiktionary (francese, spagnolo, tedesco, ecc.) e adattabile a domini tecnici (giuridico, medico, informatico) semplicemente variando i criteri di accoppiamento e le fonti dei sinonimi.

6 Lezioni dal gaming: modularità, chunk e data engineering per l'AI

L'industria dei videogiochi ha anticipato di anni molte delle sfide che oggi affrontano l'intelligenza artificiale e la linguistica computazionale, sviluppando soluzioni concrete per la gestione di enormi quantità di dati in modo modulare, patchabile e scalabile.

6.1 Gestione dei big data: chunk, index e streaming

Nei titoli open world (ad es. Minecraft, The Witcher 3, GTA V), il mondo di gioco viene suddiviso in chunk: blocchi di dati caricati e scaricati dinamicamente in base alle necessità del giocatore. Questa tecnica, insieme a formati dati serializzati come JSON e NBT, consente a motori di gioco di gestire centinaia di gigabyte senza saturare la memoria o compromettere le performance.

6.2 Modularità e patchabilità

I motori di gioco moderni supportano mod, plugin e patch che possono modificare il comportamento del gioco senza riscrivere o corrompere il dataset di partenza. Questo è reso possibile da una struttura dati pensata per essere atomica e indipendente: ogni mod è un "layer" aggiuntivo che si può caricare, rimuovere o aggiornare senza rischiare crash o incompatibilità.

6.3 Applicazione all'ingegneria dati linguistica

L'approccio a chunk e layer, così efficace nei giochi, è perfetto anche per la costruzione e gestione di dizionari concettuali e dataset linguistici di grandi dimensioni. Serializzare i gruppi di sinonimi in JSON, indicizzare ogni concetto, e suddividere la pipeline in step modulari consente:

- Accesso istantaneo a qualsiasi concetto o gruppo senza dover processare l'intero corpus ogni volta.
- Patch e correzioni incrementali, senza necessità di "ripartire da zero".
- Espandibilità a nuovi domini, lingue o tecnologie semplicemente aggiungendo nuovi layer o moduli.

- Portabilità totale: lo stesso dizionario può essere caricato in un'AI, in un motore di ricerca, in un'app didattica o in un tool di analisi linguistica.

6.4 Verso una linguistica "engineered"

Questo lavoro mostra come le tecniche nate per risolvere problemi concreti nei videogiochi possano ispirare una nuova generazione di risorse linguistiche, più flessibili, aggiornabili e compatibili con le esigenze delle AI moderne. Se la comunità NLP adotterà queste strategie "da gaming", potrà finalmente superare molti dei limiti attuali di scalabilità, modularità e interoperabilità dei dizionari e dataset linguistici.

7 Modularità e composizione nell'Intelligenza Artificiale: una nuova architettura ispirata al gaming

In questo lavoro viene proposta un'architettura modulare per l'intelligenza artificiale linguistica, ispirata alle logiche di layering e composizione dei dati tipiche dei motori grafici nei videogiochi, ma qui applicata in modo innovativo alla pipeline semantico-grammaticale.

7.1 Separazione tra semantica e grammatica

A differenza degli approcci convenzionali, la pipeline sviluppata prevede una separazione esplicita tra il modulo semantico (che gestisce la rappresentazione astratta dei concetti tramite gruppi di sinonimi perfetti) e il modulo grammaticale (che si occupa di trasformare la struttura concettuale in output linguistico naturale attraverso declinazioni, accordi, sintassi e stile).

Questa suddivisione, pur ispirandosi ai principi di modularità noti nell'ingegneria del software e del gaming, rappresenta un'applicazione originale al contesto dell'AI linguistica italiana, con vantaggi in termini di trasparenza, scalabilità e controllo.

7.2 Composizione a layer e analogia con il rendering grafico

Come nel rendering grafico, dove la scena finale nasce dalla composizione di molteplici layer (mesh, texture, illuminazione, effetti), così nell'architettura qui presentata il linguaggio generato è il risultato della combinazione flessibile di moduli specializzati, ciascuno responsabile di una funzione ben definita.

7.3 Benefici della modularità

L'adozione di questa struttura rende il sistema:

- Più semplice da aggiornare o patchare (basta modificare un modulo senza riscrivere tutto).
- Facilmente estendibile a nuovi domini, lingue o stili.
- Più trasparente nel processo di generazione, grazie alla tracciabilità dei passaggi dal concetto alla realizzazione linguistica.

7.4 Prospettive e adozione futura

Questa architettura apre nuove prospettive per la progettazione di IA linguistiche "componibili" e adattabili, e si auspica che possa stimolare la sperimentazione di soluzioni analoghe anche in altri ambiti e lingue.

8 Stato attuale dell'IA: training, risultati preliminari e rilascio futuro

Attualmente, l'intelligenza artificiale basata sul dizionario concettuale descritto in questo lavoro è in fase di training su un dataset privato, non pubblicabile per motivi di licensing e privacy. Questa fase è fondamentale per testare l'effettiva fattibilità tecnica e semantica dell'approccio adottato prima di rendere disponibili risorse e modelli pubblici addestrati esclusivamente su dati open.

8.1 Dati preliminari sul training

Il modello, con una dimensione di circa 234 milioni di parametri e un vocabolario concettuale di 31.665 token, è stato inizializzato e avviato su hardware consumer (GPU Nvidia RTX 3070). L'addestramento avviene in mixed precision (float16), utilizzando l'ottimizzatore AdamW e configurazioni standard di deep learning, per garantire massima compatibilità e replicabilità.

Di seguito, alcuni dati estratti dalle prime fasi di training:

- **Token per iterazione:** 245,760
- **Numero totale di parametri:** 233.79 milioni
- **Vocabolario concettuale:** 31.665 token
- **Step iniziali di training:**
  - `step 0:` train loss 10.60, val loss 10.61
  - `step 50:` train loss 10.33, val loss 10.33
  - `step 80:` train loss 10.00, val loss 10.00
  - **MFU:** 1.46% (efficienza ottima, modello altamente gestibile)
- **Ottimizzatore:** AdamW (fused)
- **Backend:** nccl, mixed precision attiva

8.2 Prospettive e rilascio pubblico

Non appena completato il training e validata la qualità del modello su dati privati, verrà avviata la fase di addestramento su dataset interamente pubblico, per consentire la massima trasparenza e verificabilità da parte della comunità.

L'obiettivo è mettere a disposizione un modello AI "nativo italiano", pienamente documentato, testato e pronto per l'integrazione in tool open source e applicazioni concrete. Gli aggiornamenti saranno pubblicati in tempo reale, sia tramite il repository che nei preprint di questo paper.

8.3 Implicazioni pratiche

I dati raccolti finora confermano la fattibilità di addestrare modelli di ampia scala su base concettuale e con risorse hardware accessibili, aprendo la strada a una vera democratizzazione della NLP italiana di nuova generazione.

8.4 Obiettivo di leggerezza e accessibilità

Uno degli obiettivi principali del progetto era dimostrare che, grazie a una tokenizzazione realmente concettuale e alla rimozione sistematica della ridondanza lessicale, è possibile addestrare e utilizzare un modello di ampia scala (circa 240 milioni di parametri) mantenendo al contempo una leggerezza computazionale non comune per modelli di questa taglia.

I dati di utilizzo GPU confermano il risultato: l'addestramento del modello, anche su hardware consumer (es. RTX 3070), sfrutta solo l'1.5% delle risorse GPU disponibili durante le prime fasi di training. Questa efficienza è ottenuta sia per la compattezza del vocabolario concettuale sia per l'assenza di duplicazioni e "rumore" nel dataset, che normalmente causano overhead e inefficienza nei LLM tradizionali.

Questa scelta architetturale rende il modello facilmente deployabile non solo su GPU dedicate, ma anche su iGPU e CPU moderne. Si apre così la possibilità di una democratizzazione concreta dell'intelligenza artificiale linguistica: modelli di grandi dimensioni, potenti e semantici, ma utilizzabili da chiunque, ovunque, anche senza infrastrutture costose o cloud a pagamento.

Il risultato pratico è la possibilità di integrare AI avanzata in software desktop, applicazioni offline, dispositivi embedded e ambienti a risorse limitate, senza rinunciare alla qualità della generazione o alla granularità semantica del modello.

8.5 Osservazioni sui trend delle loss e tuning del training

L'esperienza raccolta nei training precedenti ha mostrato che, con questa architettura e tokenizzazione, sia la training loss che la validation loss calano drasticamente (di circa 2 punti ogni 500 cicli) già dalle prime fasi di addestramento. Questo comportamento, poco comune in molti LLM, ha suggerito la necessità di una regolazione molto fine di batch size, decay del learning rate e frequenza di valutazione dei checkpoint.

Per questo motivo il settaggio adottato prevede controlli molto ravvicinati sulle metriche di loss (ogni 100 step), consentendo aggiustamenti rapidi dei parametri in base al comportamento delle curve. Questo approccio garantisce sia la massima efficienza nell'apprendimento che la prevenzione di fenomeni di overfitting precoce, sfruttando al meglio la compattezza e la pulizia del dataset concettuale.

Tale tuning empirico dimostra la flessibilità della pipeline e l'importanza di un monitoraggio costante nei primi stadi del training, soprattutto quando si opera su dati strutturati in modo innovativo come quelli prodotti dal presente lavoro. In particolare, l'osservazione della rapida discesa delle curve di loss ha portato alla scelta di un valore di decay del learning rate relativamente alto (0.99), più conservativo rispetto agli standard comuni. Questa impostazione rallenta intenzionalmente la diminuzione del learning rate, permettendo al modello di continuare ad apprendere efficacemente anche dopo i primi "cali" iniziali, senza perdere la capacità di generalizzare e adattarsi ai dati nei cicli successivi.

Il valore di decay 0.99, insieme al monitoraggio ravvicinato delle metriche, si è rivelato ottimale nel mantenere un equilibrio tra velocità di apprendimento e stabilità delle performance sul validation set.

8.6 Rilascio del vocabolario: meta.pkl già disponibile

Per facilitare la sperimentazione e permettere alla comunità di valutare e integrare da subito il vocabolario concettuale sviluppato, viene allegato a questo lavoro (o reso disponibile tramite repository) il file `meta.pkl`, contenente l'intero mapping dei token concettuali utilizzati per la tokenizzazione.

Questo file, compatibile con il framework NanoGPT e simili, permette a chi possiede competenze in data engineering e AI di utilizzare immediatamente il vocabolario concettuale italiano in pipeline di training, analisi o sviluppo di modelli personalizzati, anche in assenza del modello AI completo.

L'accesso anticipato al meta consente di sperimentare, validare e proporre miglioramenti o varianti al dizionario, favorendo una collaborazione aperta e l'evoluzione continua della risorsa.

Oltre al file `meta.pkl`, viene fornito (o sarà pubblicato a breve) anche un file `json` che consente di tradurre ogni token concettuale nel relativo gruppo di sinonimi perfetti. Questa mappatura umanamente leggibile permette di:

- Sfruttare il vocabolario in qualunque pipeline, anche senza strumenti PyTorch.
- Integrare facilmente i concetti in motori di ricerca, tool didattici, software di analisi o traduzione.
- Consentire a ricercatori, linguisti e sviluppatori di esplorare e proporre miglioramenti o correzioni direttamente sul mapping.

Questa scelta favorisce la massima accessibilità e interoperabilità della risorsa, incentivando l'adozione e la collaborazione all'interno della comunità linguistica e AI.

8.7 Mappatura completa: sinonimi e forme flesse

Per massimizzare la portabilità e l'utilità del vocabolario concettuale, viene allegato (o reso disponibile online) un archivio `concetti_completi.zip` contenente un file JSON strutturato che, per ogni token concettuale:

- Elenca tutti i sinonimi perfetti (gruppo semantico unificato).
- Include tutte le forme flesse (coniugazioni verbali, plurali, varianti grammaticali, ecc.), permettendo la riconduzione di qualsiasi input naturale al concetto di base.

Questo traduttore rappresenta uno strumento chiave sia per sviluppatori AI sia per linguisti computazionali, consentendo la conversione automatica di testo naturale in rappresentazione concettuale e viceversa. La struttura JSON assicura inoltre la compatibilità con una vasta gamma di linguaggi di programmazione e tool di data science.

L'archivio viene aggiornato regolarmente con nuove revisioni e feedback della community.

Esempio di struttura JSON per un concetto:

```
{
  "CONC_00123": {
```

```
    "sinonimi": ["correre", "sprintare", "scattare"],
    "forme": ["corro", "corri", "corre", "corriamo",
    "correte", "corrono", "correvo", ...]
  },
  ...
}
```

9 Conclusioni

Il dizionario concettuale presentato in questo lavoro costituisce un passo avanti importante per la linguistica computazionale italiana e per lo sviluppo di intelligenze artificiali realmente "native" e trasparenti. Grazie a una pipeline di parsing e consolidamento ispirata alle tecniche del data engineering gaming, abbiamo ottenuto una risorsa compatta, facilmente accessibile e priva di ridondanze, capace di rappresentare l'intero lessico italiano con circa 32.000 concetti atomici.

La scelta di privilegiare la granularità semantica, l'assenza di ambiguità e la massima modularità ha permesso di ottenere non solo un dizionario più "umano" e interpretabile, ma anche la base per una nuova generazione di AI linguistiche: più leggere, trasparenti, componibili e pronte per essere democratizzate.

Pur riconoscendo la possibilità di alcune imprecisioni residue e la necessità di ulteriori revisioni manuali, la pipeline adottata ha dimostrato di essere ripetibile, scalabile e già fruibile sia per linguisti che per sviluppatori AI. Con la pubblicazione del meta e del traduttore JSON (sinonimi + forme flesse), la comunità ha a disposizione strumenti pratici e open source per esplorare, migliorare e integrare la risorsa in qualunque pipeline di NLP, traduzione, didattica o analisi.

I primi risultati dei training su dataset privati confermano la fattibilità e la leggerezza computazionale di un modello di grandi dimensioni basato su questo approccio. L'impegno ora è completare la validazione su dati pubblici e rendere disponibili modelli AI pre-addestrati per la massima trasparenza e diffusione.

Questo lavoro intende essere un punto di partenza, più che un punto d'arrivo: l'invito è rivolto a tutta la comunità linguistica, informatica e AI a contribuire, estendere e criticare la risorsa, affinché l'italiano abbia finalmente infrastrutture semantiche degne della sua ricchezza— e l'intelligenza artificiale possa, finalmente, parlare e ragionare davvero nella nostra lingua.

Licenza