UNIVERSITÀ
DI PAVIA

UNIVERSITA' DEGLI STUDI DI PAVIA

DIPARTIMENTO DI STUDI UMANISTICI
CORSO DI LAUREA MAGISTRALE IN LINGUISTICA TEORICA,
APPLICATA E DELLE LINGUE MODERNE

Large Language Models per l'annotazione di synset: uno studio esplorativo sul Latin WordNet

RELATORE
Prof.ssa Chiara Zanchi

CORRELATORE
Prof.ssa Claudia Roberta Combei

Tesi di Laurea Magistrale di
Daniela Santoro
Matricola n. 506285

Anno accademico 2023/2024

Indice

Introduzione

La creazione e l'arricchimento di risorse lessicali per le lingue antiche rappresenta una sfida significativa nell'ambito della linguistica computazionale e della linguistica storica (Biagetti et al., 2021). In particolare, il Latin WordNet (LWN) (Minozzi et al., 2009), una risorsa lessicale strutturata che organizza il significato delle parole latine in una rete di relazioni semantiche, richiede un considerevole lavoro di annotazione manuale per il suo sviluppo e ampliamento. Questa tesi esplora le potenzialità degli Large Language Models (LLM) come strumenti di supporto per l'arricchimento semi-automatico del LWN, con particolare attenzione alla generazione di sinonimi e al popolamento dei synset.

Il LWN, sviluppato inizialmente nel 2004 seguendo l'*Expand Method* (Vossen, 2002), rappresenta un esempio significativo di adattamento del modello WordNet alle lingue antiche. La metodologia iniziale si è basata sulla traduzione automatica di dati dall'inglese e dall'italiano attraverso il MultiWordNet, utilizzando dizionari bilingui. Questo approccio ha portato alla creazione di un database iniziale di 9.378 lemmi e 8.973 synset. Tuttavia, questo metodo ha evidenziato alcune criticità, in particolare una eccessiva dipendenza dalle strutture semantiche dell'inglese e dell'italiano, che ha portato in alcuni casi all'inclusione di significati anacronistici e imprecisi, specialmente nell'ambito della terminologia tecnica. Successivamente, il progetto ha attraversato diverse fasi di evoluzione e affinamento: un importante contributo è stato fornito dal lavoro di Franzini et al. (2019), che ha proposto una revisione del LWN attraverso la rimozione manuale dei termini moderni e l'integrazione dei significati mancanti. Un ulteriore sforzo di espansione è stato avviato nel contesto del progetto LiLa (Passarotti et al., 2019; Mambrini et al., 2021), che mira alla costruzione di una Knowledge Base di risorse linguistiche interconnesse per il latino utilizzando gli standard Linked Open Data. La porzione annotata e ripulita del LWN comprende attualmente 18.227 synset associati a 10.449 lemmi. Il lavoro sul Latin WordNet continua nel quadro del progetto *Linked WordNets for Ancient Indo-European Languages*, il cui obiettivo è estendere e armonizzare tre WordNet per latino, greco antico e sanscrito. Parallelamente, presso l'Università di Exeter, il LWN è stato ulteriormente espanso fino a 70.000 lemmi

utilizzando un metodo di gloss-ranking per l'assegnazione dei synset, che attribuisce un peso maggiore agli equivalenti di traduzione che ricorrono più frequentemente nelle glosse dei dizionari di riferimento, riducendo così l'impatto degli *outlier*.

In questo contesto, il presente lavoro si propone di investigare l'utilizzo dei LLM per supportare e velocizzare il processo di annotazione manual, mantenendo un approccio *human-in-the-loop* che combini l'efficienza degli strumenti computazionali con l'esperienza di un annotatore umano. La ricerca si concentra specificamente sul confronto tra diversi approcci metodologici – *zero-shot, few-shot e fine-tuning* – valutandone l'efficacia nella generazione di potenziali sinonimi latini. L'esperimento prevede l'utilizzo del modello Mistral-7B, selezionato per il suo ottimale bilanciamento tra prestazioni ed efficienza. L'esperimento è strutturato in tre fasi principali di complessità crescente: l'implementazione di approcci *zero-shot* e *few-shot*, lo sviluppo di una baseline in inglese per la validazione metodologica, e infine l'applicazione di tecniche di *fine-tuning* attraverso *Low-Rank Adaptation* (LoRA) (Hu et al., 2021). Per quest'ultima fase, è particolarmente significativa la scelta di utilizzare come dati di training il LWN stesso, creando così un ciclo di feedback dove il modello, inizialmente addestrato su dati strutturati provenienti dal LWN, viene successivamente utilizzato per generare nuovi dati della stessa natura. Questa metodologia non solo esplora il potenziale dei LLM nell'arricchimento di risorse linguistiche, ma esamina anche la possibilità di un'interazione bidirezionale tra modelli linguistici e database lessicali.

La tesi si articola in tre capitoli. Il primo capitolo fornisce il background teorico necessario, discutendo la struttura e l'evoluzione del Latin WordNet, nonché i principi fondamentali dei LLM e le loro applicazioni nella linguistica computazionale. Il secondo capitolo descrive dettagliatamente la metodologia, presentando il modello utilizzato, il dataset, le tecniche di *prompt engineering* e il processo di *fine-tuning*. Il terzo capitolo è dedicato all'analisi dei risultati, offrendo sia una valutazione quantitativa attraverso metriche standard sia un'analisi qualitativa delle generazioni prodotte dal modello. Le conclusioni discutono le implicazioni dei risultati ottenuti e suggeriscono possibili direzioni per ricerche future, con particolare attenzione al miglioramento delle performance attraverso l'incorporazione di informazioni etimologiche, l'uso di dati cross-linguistici e l'implementazione di tecniche di grounding più sofisticate.

1. WordNet e LLM

WordNet è una risorsa semantico-lessicale sviluppata inizialmente per l'inglese all'Università di Princeton (Fellbaum, 1998; Miller and Fellbaum, 2007; Miller et al., 1990). Successivamente, il modello è stato adattato per creare risorse simili in numerose altre lingue (Vossen 1998), incluse lingue antiche come il latino, il greco antico, il sanscrito e l'inglese antico (Biagetti et al., 2021; Khan et al., 2023; Bizzoni et al., 2014; Minozzi, 2017; Franzini et al., 2019; Mambrini et al., 2021; Boschetti, 2019; Zanchi et al., 2021; Hellwig, 2017; Khan et al., 2022). Organizzato in "synset", gruppi di sinonimi, WordNet permette una rappresentazione strutturata delle connessioni semantiche tra essi. La struttura semantica organizzata di WordNet è fondamentale nell'elaborazione del linguaggio naturale per diverse ragioni chiave. Offre una rappresentazione strutturata del significato che facilita la disambiguazione delle parole, l'inferenza semantica e l'analisi della similitudine concettuale. Questa rete di relazioni semantiche supporta varie applicazioni, dalla traduzione automatica alla generazione di testo coerente, passando per la *query expansion* e la *sentiment analysis*. Inoltre, fornisce una base standardizzata per la costruzione di ontologie e la normalizzazione del lessico, rendendo WordNet uno strumento versatile e potente per compiti che richiedono una comprensione approfondita del significato e delle relazioni tra le parole. Sebbene dunque, questa struttura semantica, sia ampiamente utilizzata in linguistica computazionale, presenta limitazioni in termini di copertura e aggiornamento. L'integrazione con i modelli linguistici di grandi dimensioni (in inglese: *Large Language Models*, abbreviato LLM) potrebbe offrire opportunità significative per ampliare e arricchire WordNet, sfruttando la vasta conoscenza linguistica implicita nei LLM per ampliare il repertorio lessicale e le relazioni semantiche di WordNet.

Parallelamente, i LLM hanno rivoluzionato la generazione automatica del linguaggio naturale, dimostrando una notevole capacità di creare testo simile al linguaggio umano. Basati prevalentemente su architetture *Transformer*, come GPT e BERT, questi modelli sono addestrati su vasti corpora di testo strutturato e non strutturato. Tuttavia, nonostante le loro capacità avanzate, i LLM presentano limitazioni notevoli, in quanto mostrano eccellenti performance principalmente in compiti relativi alla generazione di linguaggio naturale ma non ne raggiungono una vera e propria comprensione.

L'obiettivo di questa ricerca è volto a esaminare le prospettive dell'interazione tra LLM e WordNet per il futuro dell'elaborazione del linguaggio naturale. Se da un lato i LLM possono beneficiare delle strutture semantiche di WordNet per migliorare la comprensione e la generazione di testo – utilizzando gli stessi dati presenti e annotati in WordNet per l'addestramento –, i synset di WordNet possono essere arricchiti attraverso l'analisi dei grandi volumi di dati testuali processati dai LLM, favorendo l'automatizzazione e alleggerendo il lavoro del task di annotazione di synset stessi.

1.1. WordNet

WordNet è un *database* semantico-lessicale sviluppato in primo luogo per la lingua inglese da George Armitage Miller e Fellbaum presso l'Università di Princeton. L'organizzazione del lessico si avvale di raggruppamenti di termini con significato affine chiamati "synset", contrazione di "*synonymic set*", che organizzano le parole secondo i concetti che rappresentano piuttosto che in ordine alfabetico (Miller, 1995). WordNet si basa sul concetto di sinonimia contestuale: due parole sono considerate sinonimi in WordNet se possono essere intercambiabili in almeno un contesto senza alterare significativamente il significato complessivo della frase. Le parole in un synset non devono essere sinonimi perfetti in ogni contesto, è sufficiente che condividano un significato di base simile. Questa struttura del lessico è fondamentale per applicazioni di linguistica computazionale ed elaborazione automatica del linguaggio naturale, poiché modella le connessioni semantiche in maniera funzionale per il processamento algoritmico.

I synset sono le fondamenta della costruzione di WordNet e rappresentano gruppi di sinonimi che condividono un significato comune ciascuno etichettato con una definizione che delinea il concetto sottostante. La costruzione di tali synset segue principi psicolinguistici e computazionali che mirano a riflettere la struttura della memoria lessicale umana (Fellbaum, 1998). Ogni synset è interconnesso attraverso diverse relazioni semantiche, tra cui iperonimia, iponimia, meronimia e antonimia. All'interno del synset, quando una parola ha molteplici significati o sfumature di significato, queste vengono elencate e spiegate individualmente all'interno del synset; ogni numero corrisponde a un senso distinto della parola, e ciascuno è accompagnato da una definizione che ne chiarisce il significato specifico. Questa numerazione e

definizione dei sensi all'interno dei synset aiuta a disambiguare i diversi usi di una parola e a mappare precisamente le relazioni semantiche con altri synset.

Ad esempio, il synset per la parola *car* include termini come *automobile*, *machine* e *motorcar*, tutti raggruppati sotto la stessa definizione: «*A motor vehicle with four wheels; usually propelled by an internal combustion engine*». Questo synset è collegato a vari altri concetti tramite relazioni semantiche: *car* è un iponimo di *vehicle*, il quale a sua volta è un iperonimo, ed è collegato a termini come *convertible* e *sedan* tramite relazioni di meronimia.

Un synset, o insieme di sinonimi, dunque è un gruppo di termini che veicolano lo stesso concetto. La definizione associata al synset, sebbene utile per la comprensione umana, è secondaria rispetto al gruppo di parole stesso, che costituisce il vero cuore del synset. Pertanto, una singola parola può apparire in più synset se polisemica. L'organizzazione in synset offre un vantaggio significativo nella strutturazione del lessico: invece di creare collegamenti diretti tra ogni parola e tutti i suoi possibili sinonimi, il che porterebbe a una rete estremamente complessa e ridondante, WordNet raggruppa i sinonimi in un unico "nodo" concettuale. Questo approccio semplifica notevolmente la struttura della rete semantica, rendendo più efficiente sia la sua costruzione che il suo utilizzo in applicazioni di elaborazione del linguaggio.

Le variazioni sinonimiche possono anche essere influenzate da fattori diafasici, diastratici o diatopici, che modificano l'uso delle parole a seconda del contesto sociale, stilistico o regionale. Secondo Fellbaum (1998), i synset costituiscono molto più che una semplice "rete di parole": essi rappresentano il fulcro di *thesauri*, dizionari e ontologie. La sinonimia è considerata la pietra angolare di WordNet, l'iperonimia ne costituisce l'asse portante e la meronimia il legante essenziale (*ibidem*). Tuttavia, queste relazioni non si stabiliscono primariamente tra i synset stessi, ma sono relazioni lessico-semantiche, con il synset che rappresenta un concetto piuttosto che una forma lessicale.

Uno sviluppo significativo nel campo dei wordnet è stato il progetto *EuroWordNet* (1996-1999), una rete di wordnet per diverse lingue europee che include wordnet per l'olandese, l'italiano, lo spagnolo, il tedesco, il francese, il ceco e l'estone. *EuroWordNet* è stato finanziato dalla Comunità Europea, il che ha permesso di estendere l'approccio di WordNet a un contesto multilingue, connettendo i vari database

lessicali attraverso un Indice Interlingua, che collega ogni synset a un'ontologia superiore condivisa, rendendo possibile l'interoperabilità tra le diverse lingue (Vossen, 1998).

Ogni WordNet costituisce un sistema autonomo di organizzazione lessicale della lingua. Questo indice, da un lato, facilita le connessioni tra le parole di una lingua e i loro corrispondenti in altre lingue; dall'altro, fornisce l'accesso a un'ontologia che include 63 distinzioni semantiche, offrendo una struttura semantica comune per tutte le lingue, pur mantenendo le specificità di ciascun WordNet. Il database si rivela utile non solo per il recupero di informazioni in una singola lingua ma anche per ricerche che attraversano diverse lingue, come dimostrato dall'esperienza degli utenti coinvolti nel progetto. La sfida principale nella costruzione dei synset per diverse lingue sta nel bilanciare la specificità linguistica e culturale con la necessità di mantenere un certo grado di uniformità e coerenza attraverso le lingue.[1] Il progetto *EuroWordNet* ha dimostrato come le strutture lessicali possano essere adattate a contesti linguistici diversi mantenendo un framework comune che facilita la comparazione e l'analisi translinguistica (Vossen, 1998).

Negli anni sono stati dunque sviluppati diversi *WordNet* per lingue moderne ma anche per lingue antiche (Bizzoni et al., 2014; Minozzi, 2017; Franzini et al., 2019; Mambrini et al., 2021; Boschetti, 2019; Zanchi et al., 2021).

Ad esempio, *Linked WordNets for Ancient Indo-European Language* è un progetto di recente sviluppo il cui obiettivo primario è applicare il modello di WordNet alle lingue antiche, mirando a costruire reti semantiche per le lingue indoeuropee antiche – sanscrito, greco antico e latino – e armonizzare e rifinire questi tre WordNets rendendoli interoperabili. I WordNets hanno il potenziale di permettere confronti semantici interlinguistici utilizzando lo stesso insieme di synset del WordNet originale.

Nei paragrafi seguenti approfondiremo la costruzione del *Princeton WordNet*, lo sviluppo del progetto *Linked WordNets for Ancient Indo-European Language* e l'attuale stato dell'arte relativo alle applicazioni dei WordNet e relativi modelli computazionali.

[1] https://www.ilc.cnr.it/progetti/eurowordnet-2/

1.1.1. Princeton WordNet

Come discusso nella Sezione 1.1, il *Princeton WordNet*[2] è un database lessicale elettronico per la lingua inglese, sviluppato a partire dal 1986 presso l'Università di Princeton. Ideato da George A. Miller, psicolinguista ispirato dagli esperimenti di Intelligenza Artificiale mirati a comprendere la memoria semantica umana (Collins & Quillian, 1969), il progetto si basa sull'idea che i parlanti conoscano decine di migliaia di parole e concetti. Pertanto, risulta ragionevole ipotizzare l'esistenza di meccanismi efficienti ed economici per l'archiviazione e l'accesso a tali informazioni (Miller, 1995; Fellbaum, 1998a).

WordNet è stato ideato da George Armitage Miller (1995) e successivamente ampliato da Christiane Fellbaum (Fellbaum, 1998), fino ad essere rilasciato nella sua forma definitiva come Princeton WordNet (PWN) 3.0 nel 2006. Da allora, è stata rilasciata solo una versione di manutenzione (3.1) nel 2011, che ha ridotto il numero di parole coperte. Tuttavia, l'interesse e l'uso dei WordNet sono cresciuti a livello globale, con molti progetti che hanno creato nuovi WordNet per lingue diverse dall'inglese e aggiunto nuovi dati, come informazioni relative al sentiment (Esuli & Sebastiani, 2006) o enciclopediche (Navigli & Ponzetto, 2012) e l'aggiunta di pronomi ed esclamativi, non inclusi nel PWN (Da Costa & Bond, 2016).

Il PWN – come anche la maggior parte degli "altri" WN – copre solo quattro parti del discorso: nomi, verbi, aggettivi e avverbi. I nomi sono organizzati in una gerarchia in cui ogni termine è un iponimo di una singola *entity*. I verbi sono anch'essi raggruppati in gerarchie, sebbene non esista un concetto sovraordinato per i verbi e la loro struttura risulti pertanto più disconnessa. Per gli aggettivi, la struttura si basa generalmente su un modello "a bilanciere" (*dumbell-model*), dove gli aggettivi sono raggruppati in coppie di antonimi, come "caldo"–"freddo", e aggettivi satellite collegati all'estremità del bilanciere. Gli avverbi, invece, presentano una struttura meno definita e molti synset di avverbi non hanno connessioni nel grafo (Fellbaum, 1998).

Come si è detto nella sezione 1.1, il cuore dell'architettura WordNet è il synset, il cui concetto centrale è la sinonimia. Miller et al. (1993) basano la loro definizione di sinonimia sulla prospettiva di Leibniz, secondo cui la sostituzione di una parola in una frase con un sinonimo non cambia il valore di verità della frase stessa nei suoi usi.

[2] Da qui in poi si troverà citato anche solo come "WordNet" o WN.

Tuttavia, tale definizione limita severamente il numero di coppie sinonimiche in qualsiasi lingua naturale. Per questo motivo, gli autori hanno proposto un criterio più flessibile per identificare i sinonimi: è sufficiente che le condizioni di verità siano preservate solo in alcuni contesti o usi (Miller et al., 1993).

La sua struttura a grafo rende WordNet altamente adatto alle applicazioni di elaborazione del linguaggio naturale (NLP), che ha sviluppato vari metodi per sfruttare questa caratteristica. Ad esempio, la similarità tra parole può essere calcolata semplicemente contando il numero di archi necessari per connettere due parole (Wu & Palmer, 1994), e metodi più sofisticati sono stati costruiti su questo principio (Lin & Sandkuhl, 2008). Inoltre, rimane la risorsa più utilizzata per la disambiguazione del senso delle parole (WSD), tra i task principali della NLP (Navigli, Jurgens & Vannella, 2013).

Nonostante le critiche riguardanti la presenza di errori e distinzioni di senso troppo dettagliate o troppo generiche (Hovy et al., 2006; McCrae & Prangnawarat, 2016), WN continua a essere una risorsa fondamentale nel campo della linguistica computazionale e anche nei recenti sviluppi dell'intelligenza artificiale generativa. La sua struttura a grafo continua a essere esplorata e sfruttata per sviluppare reti neurali e *embedding-model* (Kutuzov et al., 2018; Rothe & Schütze, 2015).

A tal proposito, gli *embedding* sono rappresentazioni numeriche di parole che mirano a mappare le relazioni semantiche e sintattiche tra le parole in un contesto linguistico. Questi vettori multidimensionali collocano parole con significati simili in posizioni vicine nello spazio vettoriale, facilitando il rilevamento di similarità e relazioni tra esse. Gli *embedding* si basano sulla *Distributional Hypothesis*, secondo cui le parole con significati simili tendono a comparire in contesti simili (Mikolov et al., 2013; Pennington et al., 2014). Due approcci principali sono utilizzati per generare *embedding*: quelli basati sulla frequenza e quelli basati sulla previsione. Gli *embedding* basati sulla frequenza, come TF-IDF, considerano la frequenza delle parole nei documenti per determinare la rilevanza di un termine rispetto ad un documento o ad una collezione di documenti. In contrasto, gli *embedding* basati sulla previsione, come *Word2Vec* e *GloVe*, mappano relazioni semantiche e contestuali predicendo parole mancanti in base alle parole circostanti (Mikolov et al., 2013; Pennington et al., 2014). *Word2Vec*, sviluppato da Mikolov et al. nel 2013, è uno dei modelli di *embedding* più

noti e ha due varianti principali: *Continuous Bag of Words* (CBOW) e *Continuous Skip-gram*. CBOW prevede una parola target basata sulle parole circostanti, mentre Skip-gram fa il contrario, predicendo parole circostanti date una parola target (Mikolov et al., 2013). Gli *embedding* delle parole sono fondamentali per numerose applicazioni di NLP. Grazie alla loro capacità di generalizzare anche su parole sconosciute o rare, gli embeddings migliorano significativamente le prestazioni dei modelli di NLP (Pennington et al., 2014; Conneau et al., 2018).

1.1.1.1. Parole, significati e relazioni

Un dizionario tradizionale organizza le informazioni lessicali seguendo metodi convenzionali: le parole sono disposte in ordine alfabetico per facilitarne la ricerca, e i significati di ciascuna parola sono elencati generalmente partendo dall'uso più frequente o dal significato principale. In WordNet, invece, le informazioni sono organizzate per significato e parti del discorso, legate tra loro da diversi tipi di relazioni. WordNet distingue tra *Word Form* – la forma scritta della parola, i lemmi – e *Word Meaning* – il significato espresso dalla parola stessa. Il dizionario di WordNet è diviso in quattro morfosintattiche: sostantivi, verbi, aggettivi e avverbi, ognuna delle quali è, a sua volta, raggruppata in synset. Ogni insieme di sinonimi rappresenta un particolare concetto ed è legato ad altri synset tramite relazioni semantiche. Le relazioni fra singoli lemmi o synset avviene, invece, tramite relazioni lessicali (Fellbaum, 1998).

La classificazione delle parole in WordNet inizia dunque con le relazioni tra lemma e significato. Possiamo rappresentare questa teoria con una Matrice Lessicale, in cui troviamo i significati delle parole nelle righe (*Word meanings*) e i lemmi nelle colonne (*Word forms*), come possiamo vedere in Tabella 1.

Word Meanings	Word forms				
	F_1	F_2	F_3	$F_{...}$	F_n
M_1	V(1,1)	V(2,1)			
M_2		V(2,2)			
$M_{...}$					
M_m					V(m,n)

Tabella 1. Matrice Lessicale

Ad esempio, la colonna per *bank* può avere significati come "*financial institution*", "*the side of a river*" quando è un sostantivo, o "*to tilt*" e "*to rel*" quando è un verbo. Un valore non nullo nella matrice indica che la forma F esprime il significato M; valori non nulli nella stessa riga rappresentano sinonimi, mentre valori non nulli nella stessa colonna indicano polisemia. La matrice lessicale può essere rappresentata da due blocchi con frecce bidirezionali, denominati *Word Meaning* e *Word Form*. Queste frecce indicano che si può partire da un significato per trovarne la forma appropriata o viceversa.

WordNet è dunque organizzato in relazioni semantiche, che coinvolgono i significati (rappresentati nei *synset*), e relazioni lessicali, che stabiliscono le relazioni tra i lemmi. La semantica lessicale di stampo (neo)strutturalista parte dall'idea che una parola associ la sua espressione (forma) ai concetti che esprime. Riprendendo Tabella 1, sinonimia e polisemia sono aspetti complementari della Matrice Lessicale: le relazioni sono "*molti-a-molti*", rispecchiando i processi mentali di comprensione e produzione linguistica; questo significa che una parola può avere diversi significati (polisemia) e diversi significati possono essere espressi da parole differenti (sinonimia).

La costruzione della base di dati di WordNet ha dovuto tenere in considerazione due teorie principali: la teoria costruttiva e la teoria differenziale. La teoria costruttiva sostiene che la costruzione accurata di un concetto richieda un numero sufficiente di informazioni per caratterizzarlo e distinguerlo da altri concetti lessicali, fornendone una definizione precisa (Miller, 1995). La teoria differenziale, invece, afferma che la rappresentazione di un concetto può essere fatta solo con elementi che lo distinguano da altri (Fellbaum, 1998). Riprendiamo l'esempio precedente per chiarire meglio la questione: la parola *bank* può avere i seguenti significati: (a) istituto finanziario che offre servizi come depositi e prestiti; (b) la riva di un fiume. Nella teoria costruttiva, per differenziare i due significati è necessario fornire un numero sufficiente di informazioni per distinguerli chiaramente. Questo può includere dettagli specifici e definizioni precise. Ad esempio, per il primo significato, "*bank*" può essere definita come "un istituto finanziario che raccoglie depositi, concede prestiti e fornisce altri servizi finanziari". Per il secondo significato, "*bank*" può essere definita come "la riva o la sponda di un fiume o di un corso d'acqua". Nella teoria differenziale, invece, basta fornire una lista di forme che esprimano i vari significati. Il significato M può essere

espresso con una lista di forme (F1, F2, ...). In questo modo, per ogni significato avremo una lista di forme fra di loro in relazione di sinonimia, il nostro synset. Ritornando all'esempio di prima, per distinguere i significati della parola "*bank*", sarebbe sufficiente citarne due sinonimi: "*financial institution*" per il primo e "*riverbank*" per il secondo.

Nell'implementazione di queste teorie, WordNet organizza i concetti lessicali utilizzando una varietà di relazioni semantiche, di cui la sinonimia rappresenta una componente fondamentale ma non esclusiva, permettendo di rappresentare le connessioni semantiche tra le parole in modo ricco e dettagliato. Come anticipato nella Sezione 1.1.1, le relazioni principali implementate in WordNet includono:

a. Sinonimia
b. Antonimia
c. Iponimia/Iperonimia
d. Meronimia/Olonimia

La definizione tradizionale di sinonimia, basata sul pensiero di Leibniz e ripresa da Fellbaum (1998), descrive una relazione tra parole in cui la sostituzione di una con l'altra in una frase non altera sostanzialmente il significato complessivo dell'enunciato. Secondo una definizione più flessibile, due concetti sono sinonimi in un contesto linguistico se la sostituzione di un concetto con l'altro nel contesto non ne altera il valore di verità. Ad esempio, la parola "*car*" può essere sostituita con "*automobile*" senza alterare il significato della frase "*I drive a car*". Oltre a questa definizione basata sulla conservazione del significato, esiste un'interpretazione più ampia che considera i sinonimi come parole con una similarità semantica sufficiente a renderle intercambiabili in determinati contesti, pur riconoscendo che la sinonimia perfetta è rara nel linguaggio naturale.

Mentre la sinonimia rappresenta una relazione di equivalenza semantica, l'antonimia esprime una relazione di opposizione, sebbene sia ugualmente complessa da definire con precisione. L'antonimo di una parola *x* viene spesso definito come *not-x* ma questa definizione si rivela spesso insufficiente o imprecisa. Ad esempio, "*hot*" e "*cold*" sono antonimi, sebbene non tutto ciò che non è "*hot*" sia necessariamente "*cold*". Cruse (1986) offre un'analisi più approfondita del''antonimia, distinguendo diversi tipi di relazioni antonimiche:

a. *Complementary antonyms*: coppie di termini che dividono un dominio in due parti mutuamente esclusive, come "vivo/morto" o "presente/assente".
b. *Gradable antonyms*[3]: coppie che rappresentano estremi opposti di una scala continua, come "caldo/freddo" o "grande/piccolo". Questi permettono gradi intermedi e forme comparative.
c. *Directional antonyms*: coppie che indicano movimenti o cambiamenti in direzioni opposte, come "salire/scendere" o "entrare/uscire".
d. *Converses*: coppie di termini che descrivono la stessa relazione da prospettive opposte, come "comprare/vendere" o "marito/moglie".

Cruse sottolinea che l'antonimia non è semplicemente una relazione di opposizione binaria, ma un fenomeno complesso che coinvolge vari tipi di contrasto semantico. Questa categorizzazione più dettagliata aiuta a comprendere perché la semplice definizione di *not*-x spesso non rappresenta adeguatamente la natura dell'antonimia nel linguaggio naturale.

Oltre a sinonimia e antonimia, altre relazioni semantiche cruciali nella struttura di WordNet sono l'iponimia e iperonimia, che collegano concetti attraverso una gerarchia semantica. Un concetto rappresentato dal synset {x1, x2, x3, ...} è un iponimo del concetto rappresentato dal synset {y1, y2, ...} se "un x è un tipo di y" è una frase accettabile (Miller, 1995). Ad esempio, "*sparrow*" è un iponimo di "*bird*".

Complementare a queste relazioni gerarchiche, la meronimia esprime il concetto di "parte di". Un concetto {x1, x2, ...} è un meronimo di un concetto {y1, y2, ...} se "x è parte di y" è una frase accettabile (Cruse, 1986). Ad esempio, "wheel" è un meronimo di "car". Le relazioni di meronimia sono spesso complesse e possono includere varie sottocategorie come componente/oggetto o porzione/intero.

Infine, per completare la rete di relazioni semantiche, WordNet incorpora anche relazioni morfologiche, che collegano forme diverse della stessa parola. Ad esempio, la parola "*trees*" è collegata morfologicamente al singolare "*tree*". Durante la costruzione di WordNet, è stato necessario implementare la gestione delle relazioni morfologiche per evitare ridondanze e gestire le numerose forme irregolari di nomi e verbi (Fellbaum, 1998).

[3] In semantica lessicale *contraries* (Lyons, 1977. Semantics. Cambridge University Press.).

1.1.1.2. Parti del discorso

In passato, abbiamo analizzato la struttura di WordNet, focalizzandoci su come questo database lessicale organizza il significato delle parole. In particolare, abbiamo esplorato le diverse relazioni semantiche e lessicali che WordNet utilizza per collegare i concetti. Queste relazioni formano la base della struttura di WordNet, permettendo di navigare tra i significati delle parole e comprendere le loro connessioni semantiche, andando a formare la base della struttura complessa e dinamica di WordNet, che – come si è detto – tiene conto delle basi gettate dalla teoria costruttiva e dalla teoria differenziale.

I sostantivi in WordNet sono organizzati come gerarchie di specializzazione (iperonimi/iponimi), dove ogni concetto eredita caratteristiche dal suo sovraordinato, permettendo una chiara classificazione e distinzione tra i termini. La definizione comune di un nome in genere include un termine più generale che lo descrive (iperonimo) e un elenco di caratteristiche che lo distinguono da altri. Questa relazione di iponimia introduce il concetto di ereditarietà, dove un figlio eredita dal padre tutte le caratteristiche, aggiungendone altre che lo qualificano e lo distinguono dagli altri figli. Ad esempio, nel synset {quercia}, la quercia è un iponimo di {albero}, ereditando le caratteristiche generali degli alberi ma distinguendosi per attributi specifici come la robustezza del legno e la forma delle foglie. La definizione di "quercia" potrebbe essere: "Un grande albero deciduo con legno duro e foglie lobate, noto per la sua longevità e per la produzione di ghiande".

Gli aggettivi sono invece rappresentati come iperspazi n-dimensionali, riflettendo la complessità delle loro relazioni semantiche. WordNet suddivide gli aggettivi in descrittivi, *reference-modifying* e relazionali, ognuno con caratteristiche uniche che ne determinano l'uso e la funzione linguistica:

a. Aggettivi descrittivi: Specificano il valore di una caratteristica associata a un nome, come "pesante" in "un pacco pesante", dando un valore alla caratteristica "peso" del pacco. Questa classificazione non utilizza relazioni di iponimia/iperonimia, ma può essere visualizzata come uno spazio dimensionale piuttosto che un albero gerarchico. La relazione fondamentale tra aggettivi è l'antonimia.

b. Aggettivi *Reference-Modifying*: Questa categoria è poco popolata e include aggettivi come "vecchio" in "un mio vecchio amico", che può riferirsi sia all'età dell'amico (*referent*) sia alla durata dell'amicizia (*reference*).
c. Aggettivi Relazionali: Derivano da nomi, come "atomico" da "atomo" o "musicale" da "musica". Questi aggettivi mantengono un collegamento al nome di origine e non sono legati da relazioni di antonimia.

Dunque, gli aggettivi descrittivi, come nel synset {pesante}, specificano valori di attributi; gli aggettivi relazionali, come quindi "atomico" derivato da "atomo", mantengono un collegamento al nome di origine, arricchendo ulteriormente la rete semantica mentre gli aggettivi *reference modifyin*g modificano un nome in modo tale da cambiarne la referenza specifica, distinguendolo all'interno di un insieme.

I verbi, d'altra parte, sono strutturati attraverso la relazione di troponimia – relazione simile all'iponimia per i nomi – che identifica un verbo come una modalità specifica di un altro verbo più generale. Per esempio, nel synset {correre}, "correre" è un troponimo di "muoversi", specificando un modo particolare di movimento. I verbi in WordNet sono organizzati in uno schema gerarchico che include anche i *phrasal verbs*. Sono suddivisi in 15 file: 14 relativi a gruppi semantici che denotano azioni o eventi, e un file per i verbi che denotano stati. Le relazioni tra verbi si basano su implicazione e opposizione: secondo Cruse (1986), l'implicazione tra verbi rappresenta una relazione logica in cui l'azione di un verbo presuppone necessariamente l'azione di un altro. Ad esempio, il verbo "acquistare" implica "pagare", poiché l'acquisto non può avvenire senza il pagamento. Tale relazione è fondamentale per comprendere le sfumature semantiche e pragmatiche del linguaggio. D'altra parte, l'opposizione tra verbi, come evidenziato da Lyons (1977), riguarda verbi che esprimono azioni contrastanti o reciprocamente esclusive, come "accendere" e "spegnere". Questi verbi non solo definiscono azioni opposte, ma strutturano anche la nostra comprensione delle dinamiche operative e delle trasformazioni di stato negli eventi descritti.

L'informazione sui verbi è del tipo predicato-argomento, e per ogni verbo nel synset sono inseriti uno o più *frame*[4] che indicano i contesti sintattici in cui possono

[4] Con *frame* facciamo riferimento a strutture che descrivono il modo in cui i verbi e i loro dipendenti sintattici e partecipanti semantici vengono organizzati in una lingua. Secondo Fillmore (1982), i frame verbali possono essere descritti efficacemente attraverso la teoria della, *Frame Semantics*, che considera il significato delle parole e delle espressioni linguistiche come parte di strutture concettuali più ampie, o *frame*, che rappresentano situazioni tipiche o schemi di eventi. Per ulteriori approfondimenti, si possono

occorrere: in WN i frame verbali vengono utilizzati per categorizzare e descrivere i diversi sensi dei verbi, indicando le possibili relazioni tra il verbo e i suoi argomenti[5] (Fillmore, 1982; Baker et al.,1998). Un frame sintattico per "correre" potrebbe essere "Qualcuno corre [verso qualcosa]", come in "L'atleta corre verso il traguardo"; un verbo come "comprare" può avere diverse configurazioni argomentali, come "A compra B da C" (dove A è l'acquirente, B è l'oggetto acquistato e C è il venditore).

Completando il quadro, il trattamento degli avverbi in WordNet è stato oggetto di studio sin dai primi sviluppi del database lessicale. Come descritto in Fellbaum (1989), gli avverbi sono gestiti in modo diverso rispetto ai sostantivi, ai verbi e agli aggettivi, poiché spesso derivano da aggettivi e mantengono una stretta relazione semantica con essi. In WordNet, gli avverbi sono collegati alle loro basi aggettivali attraverso relazioni derivate, facilitando così l'accesso alle loro definizioni e alle loro proprietà semantiche: ad esempio, l'avverbio "velocemente" è direttamente collegato all'aggettivo "veloce". Oltre a queste relazioni derivazionali, WordNet include anche avverbi in synset che rappresentano concetti ad essi correlati, permettendo di individuare sfumature semantiche e di utilizzo contestuale. Per esempio, l'avverbio "felicemente" può essere collegato non solo all'aggettivo "felice", ma anche a synset contenenti termini come "gioiosamente" e "allegramente", che rappresentano concetti correlati ma con leggere differenze di significato.

Il trattamento degli avverbi in WordNet non si limita alle relazioni con gli aggettivi. Studi successivi, come quelli di Miller et al. (1990), hanno evidenziato l'importanza di considerare anche le relazioni semantiche tra avverbi e verbi. Ad esempio, l'avverbio "rapidamente" non solo deriva dall'aggettivo "rapido", ma è anche semanticamente correlato al verbo "correre", indicando il modo in cui l'azione è compiuta.

La struttura gerarchica dei sostantivi tramite iperonimi e iponimi, la categorizzazione dei verbi attraverso la troponimia, e la rappresentazione degli aggettivi

---

consultare le opere di Fillmore (1982), nonché i più recenti sviluppi nell'ambito del *FrameNet* e dei modelli di *embedding* semantici che integrano conoscenze lessicali e semantiche (Baker et al., 1998; Ruppenhofer et al., 2010).

[5] Un'importante risorsa in questo ambito è rappresentata dall'*Unified Verb Index* (UVI), che integra frame verbali provenienti da diverse risorse, offrendo una visione unificata e ampliata delle relazioni verbo-argomento (Schneider et al., 2014). Questa integrazione consente un'analisi più completa e accurata dei verbi, migliorando le applicazioni linguistiche sia teoriche che pratiche. (https://uvi.colorado.edu)

come iperspazi n-dimensionali mostrano l'approccio di WordNet alla rappresentazione complessa delle relazioni semantiche. Questi meccanismi permettono una rappresentazione più ricca e interconnessa del significato delle parole, migliorando così l'elaborazione e l'analisi linguistica. Nel prossimo capitolo, esploreremo in dettaglio la struttura di WordNet, esaminando come queste relazioni vengono implementate e organizzate all'interno del database lessicale.

1.1.1.3. Struttura del database

La struttura fisica di WordNet può essere suddivisa in diverse componenti principali: i *Source File*, il *Database*, il formato dei *Source File*, la sintassi delle parole nei *synset*, e i puntatori (Tengi, in Fellbaum, 1998).

I *Source File* sono file di testo che contengono la rappresentazione delle relazioni semantiche e lessicali tra le parole. Questi file sono il prodotto di una dettagliata analisi semantica e rappresentano il cuore del database WordNet. I *Source File* organizzano i dati lessicali in synset, raggruppati secondo la categoria grammaticale (nomi, verbi, aggettivi, avverbi) e ulteriori criteri semantici.

a. Nomi (*noun*): Divisi per campi semantici come oggetti, persone, animali, ecc.
b. Verbi (*verb*): Classificati in base al loro aspetto lessicale (azioni, eventi stati).
c. Aggettivi (*adj*): Suddivisi in aggettivi descrittivi, relazionali e *reference-modifying*.
d. Avverbi (*adv*): Tutti gli avverbi sono contenuti in un unico file.

Ogni file ha un nome del tipo `pos.suffix`, dove *pos* indica la parte del discorso e *suffix* organizza i file in gruppi specifici.

Nei *Source File*, ogni *synset* occupa una singola riga. La struttura di una riga segue una sintassi specifica:

```
{ parole puntatori (glossa) }
```

a. Parole: La lista delle parole che formano il synset.
b. Puntatori: Rappresentano le relazioni tra questo synset e altri synset.
c. Glossa: Una definizione o esempio che chiarisce il significato del synset.

Per i verbi, la struttura include anche i *frames*:

```
{ parole puntatori frame (glossa)}
```

I puntatori (*pointers*) sono utilizzati per rappresentare le relazioni tra synset oppure tra parole in synset diversi. Questi includono relazioni semantiche e lessicali come antonimia, iperonimia, iponimia, meronimia, ecc. La sintassi generale dei puntatori è:

```
[lex_filename : ]word[lex_id] , pointer_symbol
```

a. *lex_filename*: (Opzionale) Il nome del file sorgente contenente la parola puntata.
b. *word:* La parola di destinazione.
c. *lex_id*: (Opzionale) L'identificatore lessicale per distinguere significati diversi.
d. *pointer_symbol*: Il simbolo che rappresenta il tipo di relazione.

i.e. `{ canine, [ dog1, cat,! ] pooch, canid,@ }`

Questo esempio mostra come i puntatori in WordNet non solo collegano le parole, ma definiscono anche la natura semantica della relazione tra i concetti rappresentati dai synset ed è proprio attraverso i puntatori che WordNet costruisce la sua ricca rete di relazioni semantiche (Fellbaum, 1998; Miller, 1995). Esempi di puntatori per i nomi includono:

! Antonym: Indica una relazione di antonimia (opposto) tra due synset. Ad esempio, *happy!sad* mostra che "happy" è l'opposto di "sad".

@ Hypernym: Indica una relazione di iperonimia (categoria più generale). Ad esempio, *dog@animal* mostra che "dog" è un tipo di "animal".

~ Hyponym: Indica una relazione di iponimia (categoria più specifica). Ad esempio, *animal~dog* mostra che "dog" è un tipo di "animal".

%m Member meronym: Indica una relazione di parte (parte di un insieme). Ad esempio, *tree%mleaf* mostra che "leaf" è parte di "tree".

Oltre ai puntatori elencati precedentemente per i verbi possiamo trovare anche:

+ Derivationally related form: Indica una relazione derivazionale tra forme verbali. Ad esempio, *decide+decision* mostra che "decide" è correlato a "decision".

Dunque, in un file dei nomi, una riga potrebbe apparire così:

```
{ dog[1] @ canine[2] (a domesticated carnivorous mammal) }
```

Mentre per un verbo potremmo trovarci davanti a una dicitura di questo tipo:

```
{ run[1] ~ jog[2] frame (to move swiftly on foot) }
```

Per quanto riguarda invece la struttura del synset stesso, le parole al suo interno possono essere espresse in due forme principali:

1. Parola semplice: word [(marker)] [lex_id]

a. *word*: La parola stessa.
b. *marker*: Un'opzione per indicare la forma grammaticale.
c. *lex_id*: Un identificatore lessicale per distinguere significati diversi della stessa parola.

Ad esempio: `{bank(n)1, financial_institution(n)1, banking_concern(n)1, banking_company(n)1}` in cui "bank" è la *word,* "(n)" è il marker che indica che si tratta di un nome (*noun*), "1" è il *lex_id* che distingue questo significato di "bank" da altri (i.e. "bank" nel senso di "riva del fiume")

2. Parola con Puntatori: word [(marker)] [lex_id], pointers
    a. Include puntatori (*pointers*) che specificano le relazioni con altri synset.

Ad esempio: `{run(v)1, jog(v)1, lope(v)1, @2000001, ~2000002, +3000001}`, dove – come nell'esempio precedente – "run", "jog", "lope" sono le *word*, "(v)" è il marker che indica la loro POS, mentre "1" è il loro *lex_id*; i numeri preceduti da simboli (@, ~, +) sono i pointers che indicano relazioni con altri synset, ad esempio @2000001 potrebbe indicare un iperonimo (es. "move").

Il *database* WordNet è infine il risultato della trasformazione dei *source file* in una struttura di dati più facilmente interrogabile. Questo processo di trasformazione è gestito da una funzione chiamata "*Grinder*", che verifica l'integrità sintattica dei file sorgente e risolve i puntatori tra i synset. Il database risultante permette agli utenti di effettuare ricerche efficienti e ottenere rapidamente le informazioni desiderate (Tengi, in Fellbaum 1998).

La struttura di WordNet, basata su relazioni semantiche e lessicali offre dunque un modello complesso e dinamico per rappresentare le informazioni lessicali, a differenza dei dizionari tradizionali che organizzano le parole in ordine alfabetico. Attraverso l'uso di synset e relazioni tra synset, WordNet consente una comprensione approfondita delle parole non solo in base alla loro forma e significato, ma anche tramite le loro interrelazioni.

1.1.2. Applicazioni di WordNet

L'origine di WordNet risale dunque all'intenzione di creare un modello lessico-concettuale e un database che consistesse sia di unità lessicali che delle relazioni tra queste unità, strutturate in una rete semantica relazionale, mirando a combinare i

vantaggi dei dizionari elettronici con quelli dei *thesauri* online. WordNet è stato progettato per coprire la maggior parte dei sostantivi, aggettivi, verbi e avverbi della lingua inglese, in cui – come si è ampiamente discusso (cfr. Sezione 1.1.1.2) – ogni concetto è rappresentato da un synset, che raggruppa parole con significati simili. Lo scopo principale di WN dunque è fornire una rete semantica ricca e ben strutturata, che faciliti l'elaborazione automatica del linguaggio naturale.

In effetti, WN è una delle risorse lessicali più utilizzate nel campo della linguistica computazionale e della NLP (Miller, 1995; Fellbaum, 1998; Pedersen, Patwardhan, & Michelizzi, 2004; Navigli, 2009; Bond & Foster, 2013; Morato et al., 2004). Nel corso degli anni, WN è stato ampiamente utilizzato in una varietà di applicazioni, grazie alla sua accessibilità gratuita e al codice aperto ben documentato[6]. Si noterà che l'uso principale di questo strumento è stato nell'area della disambiguazione di senso (WSD) (Navigli, 2009), che tuttavia non è l'unica delle sue applicazioni in ambito accademico e extra-accademico.

1.1.2.1. Word Sense Disambiguation

La disambiguazione del senso delle parole (Word Sense Disambiguation, WSD) è un task fondamentale nella NLP, nonché uno dei task classici della linguistica computazionale (AlMousa et al., 2022). La WSD consiste nel determinare quale senso di una parola viene utilizzato in un dato contesto, problema che sorge a causa della natura polisemica delle parole. La sfida della WSD è dunque cruciale per migliorare le prestazioni di varie applicazioni NLP come la traduzione automatica, il recupero di informazioni (in inglese: Information Retrieval, abbreviato IR) e l'etichettatura dei ruoli semantici (in inglese: Semantic role labeling, abbreviato SRL). Le metodologie per la WSD possono essere classificate in tre macro-categorie: approcci supervisionati, non supervisionati e *knowledge-based*.

I metodi di apprendimento supervisionato si basano su corpora annotati in cui i sensi delle parole sono pre-etichettati. Questi metodi addestrano classificatori per prevedere il senso corretto di altre parole basandosi su caratteristiche contestuali. Gli approcci supervisionati noti includono l'uso di algoritmi di apprendimento automatico come alberi decisionali, macchine a vettori di supporto (SVM) e reti neurali. Nonostante

[6] Open-English WN, fork del PWN: https://github.com/globalwordnet/english-wordnet

la loro efficacia, questi metodi sono limitati dalla disponibilità di grandi corpora annotati per senso, che sono costosi e richiedono molto tempo per essere prodotti (Agirre & Edmonds, 2007).

I metodi non supervisionati non richiedono corpora annotati; invece, si basano su dati linguistici grezzi. Tecniche come il *clustering*, dove le parole sono raggruppate in base alla somiglianza contestuale, e l'uso di statistiche di co-occorrenza sono comuni nella WSD non supervisionata. I metodi distribuzionali, che analizzano i modelli di co-occorrenza delle parole, sono particolarmente diffusi. Ad esempio, Schütze (1998) ha introdotto un metodo di disambiguazione del senso delle parole utilizzando *contextual embedding* (cfr. 1.1.1) derivati da grandi corpora.

Tra varie le categorie di WSD, gli approcci supervisionati e *knowledge-based* sono i più promettenti (Jurafsky e Martin, 2018). Tuttavia, a causa del numero limitato di dataset annotati per senso, questi sistemi incontrano difficoltà nell'eccellere e dimostrare miglioramenti significativi rispetto ad altri sistemi. D'altra parte, i sistemi basati sulla conoscenza non richiedono un dataset di addestramento poiché si basano su ampi dizionari o *knowledge-graphs* (KG). Inoltre, i sistemi basati sulla conoscenza offrono tipicamente un elevato grado di trasparenza nel loro processo decisionale, facilitando la comprensione e l'interpretazione dei loro risultati da parte degli utenti o degli sviluppatori. Questo è dovuto al fatto che tali sistemi operano seguendo regole e relazioni esplicite, rendendo più agevole tracciare e spiegare il percorso logico che porta a una determinata conclusione. Con l'avanzamento dei Linked Open Data (LOD) e dei KG specifici del dominio, questi sistemi hanno un potenziale maggiore per superare altri approcci grazie alla disponibilità di una gamma più ampia di dati. I metodi basati sulla conoscenza, dunque, utilizzano risorse lessicali esterne come dizionari, thesauri e, soprattutto, reti semantiche come WordNet e sono sfruttate, ad esempio, nell'algoritmo di Lesk, che misura la sovrapposizione tra il contesto della parola e le definizioni dei suoi sensi in un dizionario (Lesk, 1986). Questo algoritmo esegue un compito di WSD assegnando un peso a ciascun senso della parola ambigua basato sulla sua similarità semantica con altri termini all'interno della frase, del documento, o di entrambi. Il senso con il peso più alto viene selezionato come il senso corretto.

Riprendendo gli *embedding*, giá citati in 1.1.1 quando parliamo di similarità semantica dobbiamo ricordare che le sue metriche possono essere suddivise in quattro

principali categorie in base al loro approccio: *path*, *feature*, Contenuto Informativo (IC) e *hybrid* (AlMousa et al., 2021).

Le misure basate sul *path* (percorso) calcolano il numero di *edges* nel percorso più breve tra i concetti nello spazio vettoriale e sfruttano direttamente le relazioni gerarchiche in WordNet. Più lungo è il percorso tra due concetti, meno sono simili, e viceversa (Zhou et al., 2008; Zhu et al., 2016). Tra i primi approcci ricordiamo ad esempio, la misura proposta da Leacock & Chodorow (1998) che utilizza il percorso più breve tra due concetti in WordNet, normalizzandolo per la profondità della tassonomia; altro approccio notevole è quello di Wu & Palmer (1994), che considera non solo la distanza tra i concetti, ma anche la loro profondità nella gerarchia di WordNet e il loro antenato comune più specifico (Least Common Subsumer, LCS). Nel campo delle misure *path-based*, più recentemente, Wang et al. (2020) hanno proposto un metodo che combina la struttura gerarchica di WordNet con tecniche di deep learning: "*HyperVector*", utilizza *embedding* iperbolici per rappresentare la gerarchia di WordNet, permettendo una migliore rappresentazione delle relazioni gerarchiche rispetto ai metodi tradizionali basati su spazi euclidei. Le metriche *feature-based* rappresentano un concetto come un vettore di caratteristiche costruito a partire dai suoi attributi (Sanchez et al., 2012; Zhu et al., 2017). Le glosse (definizioni) in WordNet sono spesso utilizzate come fonte di caratteristiche, così come le varie relazioni semantiche (come sinonimia, antonimia, meronimia) che possono essere trattate come attributi dei concetti. Uno tra i primi approcci influenti in questa categoria è il modello di Tversky (1977), che propone una misura di similarità basata sulle caratteristiche comuni e distintive dei concetti, modello che può essere implementato efficacemente utilizzando le ricche informazioni contenute in WordNet. Metodi recenti in questa categoria hanno sfruttato modelli di reti neurali per incorporare vettori di caratteristiche che rappresentano non solo le entità e le relazioni ma l'intero grafo di conoscenza (KG): questi metodi sono noti come *Knowledge Graph Embedding* (KGE) (Camacho-Collados et al., 2016; Yu et al., 2019). Pilehvar e Camacho-Collados (2019) hanno invece introdotto nel contesto delle metriche *feature-based* "WiC" (Word-in-Context), un benchmark che sfrutta WordNet per valutare la capacità dei modelli di linguaggio di distinguere diversi significati di una parola in base al contesto. Questo lavoro ha stimolato lo sviluppo di nuovi approcci che

combinano le informazioni strutturali di WordNet con rappresentazioni contestuali derivate da modelli linguistici pre-addestrati.

Le metriche basate sul Contenuto Informativo (IC) si dividono in due categorie: estrinseche e intrinseche. Le misure IC estrinseche utilizzano dati esterni per calcolare l'informazione associata a ciascun concetto. Specificamente, queste misure si basano su un corpus di testo esterno e su modelli statistici per determinare quanto sia informativo un concetto all'interno di quel corpus (Pedersen et al., 2004). In questo approccio, la frequenza con cui un concetto appare nel corpus viene utilizzata per stimare il suo contenuto informativo: concetti più rari sono considerati più informativi. Un esempio significativo è la misura introdotta da Resnik (1995), che utilizza WordNet per definire i concetti e le loro relazioni gerarchiche, combinandoli con statistiche derivate da corpora esterni. Più recentemente, Lastra-Díaz et al. (2019) hanno proposto una nuova famiglia di misure IC intrinseche basate su WordNet. Il loro approccio, denominato "WNet", introduce una normalizzazione che tiene conto della profondità e della densità locale della tassonomia di WordNet, offrendo prestazioni competitive rispetto alle misure IC estrinseche tradizionali. Le misure IC intrinseche, invece, sono basate sulla struttura del KG, ovvero l'informazione del concetto risiede all'interno della struttura topologica del KG. Un esempio notevole è il lavoro di Seco et al. (2004), che propone una misura di IC intrinseco basata sulla struttura gerarchica di WordNet, considerando il numero di iponimi di un concetto come indicatore del suo contenuto informativo, ma vari attributi strutturali sono stati utilizzati come indicatori dell'informazione contenuta in ciascun concetto (Sebti et al., 2008). Rimanendo nel campo dei Knowledge Graph Embeddings, Colla et al. (2020) hanno proposto "ConSeC", un metodo che combina embedding contestuali con la struttura di WordNet per la WSD. Il loro approccio utilizza un grafo di conoscenza costruito dinamicamente basato su WordNet, integrandolo con informazioni contestuali estratte da modelli di linguaggio pre-addestrati come BERT. Infine, le misure di similarità ibride combinano due o più degli approcci sopra citati (Sebti et al., 2008). Dunque relativamente agli approcci ibridi, Mancini et al. (2017) hanno presentato "SW2V" (Sense2Vec), un metodo che unisce informazioni da WordNet e da corpora di testo non annotati per produrre embeddings sensibili al senso. Questo approccio sfrutta la struttura di WordNet per guidare l'apprendimento di rappresentazioni vettoriali che mappano specifici sensi delle parole. In tutti questi

metodi, le relazioni tra concetti sono il fulcro principale, sia direttamente nella struttura del grafo (per le misure basate sul percorso e IC intrinseche), sia indirettamente attraverso la rappresentazione vettoriale o statistica dei concetti (per le misure feature-based e IC estrinseche).

Tuttavia, per la maggior parte di questi metodi, come evidenzia AlMousi et al. (2020), la similarità è valutata esclusivamente sulla base delle relazioni gerarchiche (cioè iponimo/iperonimo), con l'eccezione di pochi metodi che sfruttano un numero limitato di relazioni non tassonomiche, come meronimia/olonomia e antonimia, per calcolare varie misure di correlazione: nonostante in WN siano incluse diverse categorie di relazioni non tassonomiche, queste non sono ancora state appieno sfruttate dai ricercatori per migliorare le misure di similarità e correlazione semantica (*ibidem*).

Oltre alla misura della similarità semantica, altri due importanti elementi – ricorrenti nella letteratura sulla WSD – sono la *word sense heuristics* (WSH) e l'analisi del *document context* (Navigli, 2009; Raganato et al., 2017; Chaplot e Salakhutdinov, 2018). La WSH è espressa dalla distribuzione di frequenza dei sensi della parola basata sul loro utilizzo nel dataset di addestramento (come, ad esempio, SemCor e OMSTI[7]), mentre il *document context* fornisce una parola ambigua con un contesto globale che consente la selezione del senso appropriato; entrambi costituiscono la base di nuovo algoritmo di WSD basato sulla conoscenza, denominato *Sequential Contextual Similarity Matrix Multiplication* (SCSMM) (AlMousa et al., 2022). L'algoritmo proposto segue il processo di disambiguazione del cervello umano sfruttando il contesto locale basato sul senso all'interno della frase, la conoscenza precedente dell'uso del

[7] *SemCor* (Miller et al., 1994) è un corpus annotato manualmente derivato dal Brown corpus (aggiungi riferimento). Il dataset è annotato con tag di parte del discorso (POS), lemmi e sensi delle parole basati sul grafo di conoscenza (KG) di WordNet. *SemCor* è composto da 352 documenti: 186 documenti includono tag per tutte le categorie di POS (sostantivi, verbi, aggettivi e avverbi), mentre i restanti 166 contengono tag solo per i verbi. Il numero totale di annotazioni di senso in tutti i documenti è 226.040. A nostra conoscenza, *SemCor* è il più grande corpus annotato manualmente con i sensi di WordNet ed è il corpus principale utilizzato in letteratura per addestrare sistemi di WSD supervisionati (Agirre et al., 2010; Zhong e Ng, 2010); *OMSTI* (Taghipour e Ng, 2015) è un corpus annotato automaticamente con i sensi di WordNet 3.0. Come suggerisce il nome, contiene un milione di istanze annotate per senso. Per taggare automaticamente i sensi, *OMSTI* ha utilizzato un corpus parallelo inglese-cinese con un approccio WSD basato sull'allineamento (Chan e Ng, 2015). *OMSTI* ha già dimostrato il suo potenziale come corpus di addestramento migliorando le prestazioni dei sistemi supervisionati (Taghipour e Ng, 2015; Iacobacci et al., 2016).

termine, e il contesto globale basato sul senso del documento, che sono rappresentati rispettivamente dalla similarità semantica tra i termini, dalle euristiche della frequenza dei termini e dal contesto del documento (*ibidem*).

Altri metodi più avanzati incorporano misure di somiglianza semantica basate sulla struttura di WordNet per migliorare le prestazioni della disambiguazione (Mihalcea & Moldovan, 2001). Utilizzando la rete semantica di WordNet, il WSD *system* sfrutta le relazioni tra i sinonimi (dunque i synset) e le gerarchie semantiche per identificare con precisione il significato corretto di una parola (*ibidem*). Questo processo si basa su tecniche come la selezione delle coppie sostantivo-verbo all'interno di una frase e l'uso di glossari per individuare i concetti predominanti. Algoritmi avanzati, come le reti bayesiane e i modelli statistici, sono spesso integrati con WordNet per raffinare ulteriormente i risultati del WSD *task*, rendendo le interfacce di ricerca più intelligenti e le risposte alle *query* più rilevanti e precise. La capacità di WordNet di rappresentare riccamente le connessioni semantiche tra le parole lo rende uno strumento ideale per affrontare le sfide della disambiguazione semantica in vari domini applicativi (Agirre & Edmonds, 2007).

Bisogna ricordare, a questo punto, come la WSD sia una componente piuttosto rilevante nella NLP, che trova il suo impiego in una varietà di task e ne consente un significativo miglioramento. Comprendere il significato preciso delle parole in un determinato contesto ci permette infatti di ottenere risultati più accurati e pertinenti in una varietà di campi. Di seguito, esamineremo brevemente come la WSD abbia un ruolo cruciale nell'efficacia di sistemi di recupero di informazioni, traduzione automatica, etichettatura dei ruoli semantici, categorizzazione dei testi, mappatura delle ontologie e profilazione degli utenti.

Nel recupero di informazioni (*Information Retrieval,* IR), migliora infatti l'accuratezza dei motori di ricerca assicurando che le query siano abbinate ai documenti che riflettono il significato inteso dei termini di ricerca. Sanderson (1994) ha dimostrato che incorporare WSD nei sistemi IR può migliorare significativamente le prestazioni del recupero riducendo l'ambiguità. Invece, i sistemi di traduzione automatica beneficiano della WSD traducendo accuratamente le parole polisemiche in base al contesto. A tal proposito, Carpuat e Wu (2007) hanno mostrato come la WSD delle frasi può migliorare la qualità dei sistemi di traduzione automatica statistica fornendo una migliore

comprensione contestuale. Grazie alla WSD, dunque, i traduttori automatici possono scegliere il significato corretto di una parola, riducendo errori e ambiguità nella traduzione.

L'etichettatura dei ruoli semantici (SRL), che comporta l'assegnazione di ruoli alle parole in una frase (ad esempio, agente, paziente, strumento, ecc.), può essere migliorata dalla WSD: determinando il senso corretto di una parola, i sistemi SRL possono assegnare più accuratamente i ruoli semantici, migliorando così la comprensione complessiva della struttura della frase (Palmer et al., 2010). La WSD è anche fondamentale nella categorizzazione dei testi assicurando che i documenti siano classificati in base al senso corretto delle parole chiave. Liu et al. (2007) hanno proposto un metodo in cui ogni parola chiave in un documento viene disambiguata prima di essere utilizzata per la classificazione, portando a risultati di categorizzazione più accurati. Nella mappatura delle ontologie, la WSD aiuta nell'allineamento dei concetti di diverse ontologie disambiguando i termini che in esse occorrono. Schadd e Ross (2007) hanno introdotto un metodo di disambiguazione dei concetti che migliora il processo di allineamento identificando i sensi corretti dei termini in diverse ontologie. La WSD può, infine, migliorare la profilazione e la personalizzazione degli utenti interpretando accuratamente le loro *query* e preferenze di contenuto. Disambiguando i termini nelle *query* degli utenti, i sistemi possono fornire raccomandazioni più rilevanti e contenuti personalizzati (Baeza-Yates & Ribeiro-Neto, 1999).

1.1.2.2. Altre applicazioni di WordNet

Come discusso nella sezione 1.1.2.1 relativamente alle applicazioni della WSD, possiamo immaginare come WN sia dunque ampiamente utilizzato per migliorare la ricerca e il recupero delle informazioni. Integrando le relazioni semantiche e i sinonimi, i motori di ricerca possono espandere le *query* degli utenti con termini correlati, aumentando la pertinenza dei risultati recuperati. Questo approccio non solo migliora la precisione, ma consente anche di recuperare documenti che potrebbero non contenere esattamente i termini della *query* originale, ma che sono comunque rilevanti (Fellbaum, 1998; Agirre & Edmonds, 2007).

Più nello specifico, WN ha trovato inoltre applicazione anche in ambito di *Query Expansion* (QE), un sotto-task votato a migliorare la performance dei sistemi di

IR. È noto infatti che diversi modelli di IR sono stati sviluppati per migliorare le prestazioni dei sistemi di *retrieval* (Manning et al., 2008). Tuttavia, questi metodi soffrono del comune problema di non soddisfare pienamente le esigenze degli utenti: un'indagine sulla letteratura indica che la lunghezza media delle query degli utenti sul web è di circa due o tre parole[8]. Questo tipo di informazione non è sufficiente affinché un sistema di recupero preveda esattamente quali informazioni l'utente stia cercando (Robertson & Jones, 1976). Per ridurre il divario informativo tra l'utente e il sistema di recupero, sono stati introdotti diversi metodi: la QE è uno di questi. I metodi di espansione delle query aggiungono termini supplementari alla query originale, in modo che i documenti rilevanti per la query originale, ma non recuperati inizialmente, vengano recuperati dopo l'espansione, mentre i documenti irrilevanti vengano eliminati dai set di recupero iniziali (*ibidem*).

I metodi di QE si suddividono principalmente in metodi globali e locali. I metodi globali estraggono i termini di espansione dal set di documenti inizialmente recuperati. I metodi locali si dividono ulteriormente in due categorie: *feedback* di *pseudo-relevance* e *relevance* feedback (Shukla et al., 2021). Nel primo metodo, i primi "k" documenti dal set di documenti inizialmente recuperati sono considerati rilevanti. I termini di espansione vengono quindi estratti da questi primi "k" documenti utilizzando un algoritmo specifico e poi aggiunti per recuperare il set finale di documenti. I metodi globali, invece, estraggono i termini di espansione da dataset esterni, come WordNet, per aggiungere termini semanticamente correlati alla *query* originale. Questo processo può includere sinonimi, iperonimi e altri termini collegati che arricchiscono il contesto della query, permettendo al sistema di recupero di identificare documenti rilevanti che altrimenti potrebbero essere ignorati. Ad esempio, se un utente cerca "aereo", la QE potrebbe includere termini come "velivolo" e "aeroplano" grazie a WordNet, migliorando la precisione e il richiamo del sistema di IR (Manning et al., 2008). Un approccio innovativo descritto in letteratura prevede la combinazione di metodi che sfruttano WordNet con altre tecniche, come quelle basate sulla distribuzione e sull'associazione dei termini. Questi metodi combinati si sono dimostrati più efficaci rispetto all'uso isolato di singole tecniche. Per esempio, Držík & Šteflovič (2023) hanno proposto di utilizzare WordNet insieme a metodi basati di *data augmentation* (come

[8] https://www.statista.com/statistics/269740/number-of-search-terms-in-internet-research-in-the-us/

Random Insertion, *Random Swap*, *Random Deletion*) per selezionare e pesare i termini di espansione, ottenendo così un miglioramento nelle performance di recupero su vari dataset standard .

In questo contesto, Renner, Denis e Gilleron (2023) hanno sviluppato un metodo non supervisionato sorprendentemente efficace per la previsione del *Graded Lexical Entailment* (GLE) basato esclusivamente su WordNet. Il GLE misura il grado di inclusione semantica tra due termini. A differenza della semplice relazione di iperonimia/iponimia, che è binaria (un termine è o non è un iperonimo di un altro), il GLE introduce una dimensione graduale, permettendo di quantificare quanto un termine è considerato un iperonimo di un altro su una scala continua. Questo è particolarmente utile nei sistemi di recupero delle informazioni, dove il GLE può migliorare la QE fornendo termini correlati che migliorano la precisione dei risultati di ricerca.

Il metodo di Renner et al. (2023) si ispira al lavoro di Resnik (1995) e modella il GLE come la somma di due punteggi: un punteggio simmetrico di similarità semantica e un punteggio asimmetrico di *specificity loss*. La funzione di *specificity loss* è fondamentale per mappare la natura asimmetrica dell'*entailment* lessicale. Nei sistemi gerarchici come WordNet, i concetti più specifici (iponimi) contengono più informazioni rispetto ai concetti più generali (iperonimi) (Snow et al., 2006; Budanitsky & Hirst, 2006). Quando si passa da un concetto specifico a uno generale, come ad esempio da "cane" a "animale", si perde informazione specifica. La *specificity loss* quantifica questa perdita come la differenza tra il contenuto informativo del concetto specifico e quello del concetto generale. Questa perdita di specificità è asimmetrica; la perdita da "cane" a "animale" è maggiore rispetto alla "perdita" da "animale" a "cane", che sarebbe in realtà un guadagno di specificità. Un'alta *specificity loss* indica una forte relazione di *entailment* dal concetto specifico a quello generale, mentre una bassa *specificity loss* suggerisce una maggiore similarità in termini di specificità (Renner et al., 2023). Entrambi i punteggi sfruttano la struttura gerarchica dei synset di WordNet. Il metodo include anche un semplice meccanismo di disambiguazione per gestire la polisemia nei termini. Nonostante la sua semplicità, questo approccio ha raggiunto performance superiori allo stato dell'arte, con un coefficiente di correlazione di Spearman (ρ) di 0.75 sul dataset *HyperLex*, superando tutti i metodi precedenti, inclusi

quelli basati su *word embedding* specializzati che utilizzano WordNet come supervisione debole (*ibidem*).

Ritornando a quanto detto nella Sezione 1.2.1.1., WordNet è fondamentale per la costruzione di ontologie e basi di conoscenza, poiché fornisce una struttura lessicale ricca e dettagliata. Le ontologie create con l'ausilio di WordNet sono utilizzate in vari campi, come la biomedicina, per organizzare e rappresentare la conoscenza in modo sistematico. Le relazioni semantiche codificate in WordNet facilitano l'integrazione e l'interoperabilità tra diverse basi di conoscenza, migliorando la gestione e l'accesso alle informazioni (Fellbaum, 1998; Morato et al., 2004).

Oltre alla disambiguazione del senso delle parole (WSD) e quanto ne consegue, WordNet trova dunque applicazione in numerosi altri ambiti della linguistica computazionale e dell'elaborazione del linguaggio naturale, dal miglioramento dei sistemi di Named Entity Recognition (Bunescu & Pasca 2006; Nadeau & Sekine, 2007; Magnini et al., 2002); all'uso nella Sentiment Analysis (Pang & Lee, 2008; Baccianella & Sebastiani, 2010) e non solo.

Dunque, WordNet è utilizzato – come si anticipava – nel riconoscimento delle entità nominate (Named *Entity Recognition* o semplicemente NER), dove aiuta a identificare e classificare correttamente i nomi di persone, organizzazioni, luoghi e altre entità all'interno dei testi. Utilizzando le relazioni semantiche e i sinonimi presenti in WordNet, i sistemi NER possono migliorare la loro accuratezza nella categorizzazione delle entità, riducendo gli errori dovuti all'ambiguità linguistica (Morato et al., 2004). Nell'analisi del *sentiment*, WordNet viene impiegato per valutare il tono emotivo dei testi. Le relazioni tra sinonimi e antonimi in WordNet permettono di identificare meglio le parole che esprimono sentimenti positivi o negativi. Questo approccio aiuta a comprendere meglio le opinioni e le emozioni espresse nei social media, nelle recensioni dei clienti e in altri contesti testuali (*ibidem*).

Inoltre, WN è impiegato nei sistemi di *question-answering*, al fine di puntare a una migliore comprensione delle domande poste dagli utenti e per trovare risposte pertinenti. Grazie alla sua struttura gerarchica e alle relazioni tra i synset, WordNet aiuta i sistemi QA a interpretare correttamente le domande, permettendo di individuare le risposte più appropriate nei database di informazioni. Questa capacità di comprensione semantica approfondita rende i sistemi QA più efficaci e affidabili (Morato et al., 2004*)*.

Sempre relativamente al campo del QA, l'*answer selection* (AS) è un sub-task del *document-based question answering* (DQA).

Il compito di AS si differenzia dal *Machine Reading Comprehension* (MRC) utilizzato in dataset come SQuAD e MS-MARCO (Li & Wu, 2020), in cui le risposte sono tipicamente brevi segmenti di testo estratti direttamente dal documento (Rajpurkar et al., 2016; Nguyen et al., 2016). L'AS, d'altra parte, mira a identificare un'intera frase come risposta all'interno di un documento, configurandosi come un problema di classificazione data la necessità di una comprensione più ampia del contesto e la capacità di valutare la rilevanza di un'intera frase rispetto alla domanda posta (Wang & Jiang, 2017). Il processo tipicamente include l'estrazione delle frasi candidate dal documento, la loro rappresentazione in un formato adatto al modello, la classificazione di ciascuna frase come "risposta" o "non risposta", e infine la selezione della frase con il punteggio più alto come risposta finale (Tan et al., 2016). Questo approccio permette al modello di considerare il contesto completo di ciascuna frase, valutandone la pertinenza complessiva rispetto alla domanda. L'utilizzo di frasi complete come risposte offre vantaggi significativi, tra cui una maggiore coerenza e leggibilità, la possibilità di sottolineare sfumature e dettagli, e una potenziale riduzione di ambiguità. Tuttavia, questo approccio può anche comportare sfide, come il rischio di includere informazioni irrilevanti o una maggiore complessità computazionale (Yih et al., 2013). Per migliorare la rappresentazione delle parole nel processo di AS, si propone un approccio che va oltre l'uso diretto degli embedding o la loro semplice concatenazione con gli embedding dei caratteri, sfruttando le informazioni semantiche presenti in WordNet per arricchire gli embedding di parole e caratteri con informazioni semantiche aggiuntive (Li & Wu, 2020). In particolare, utilizza i *synset* per arricchire la rappresentazione delle parafrasi, aiutando a distinguere le risposte candidate in uno spazio semantico latente. Contemporaneamente, impiega gli iperonimi per individuare conoscenze utili al ragionamento. Il processo inizia mappando ogni parola in uno spazio vettoriale, incorporando queste informazioni semantiche aggiuntive per una rappresentazione più ricca e contestualizzata. Questo approccio mira a migliorare la precisione e la rilevanza delle risposte selezionate, sfruttando le relazioni semantiche tra le parole per una comprensione più profonda del testo (*ibidem*).

Bisogna inoltre aggiungere che, nell'ambito di relazioni tra parole, un particolare problema riscontrato nei modelli di *embedding* è proprio il riconoscimento e la predizione dell'iperonimo (Pinter e Eisenstein, 2018; Bernier-Colborne e Barriere, 2018; Nickel e Kiela, 2018). Cho et al. (2020) hanno sviluppato il modello *hypo2path*, capace di ottenere performance rilevanti nel task di previsione degli iperonimi. Gli autori trattano questo problema come un compito di generazione di sequenze, dove le sequenze rappresentano i percorsi tassonomici in WordNet. Utilizzando modelli *encoder-decoder*[9]*,* hanno dimostrato che addestrare il modello a generare percorsi tassonomici migliora le prestazioni rispetto alla previsione diretta degli iperonimi. In questo approccio, WordNet fornisce i percorsi tassonomici che rappresentano le relazioni iperonimo-iponimo. Durante l'addestramento, al modello vengono fornite coppie di termini e i loro percorsi tassonomici corrispondenti, permettendo al modello di apprendere le relazioni semantiche e tassonomiche tra i termini. L'encoder codifica il termine di input in una rappresentazione vettoriale, mentre il decoder genera il percorso tassonomico associato a tale termine. Questa metodologia ha permesso al modello *hypo2path* di superare i benchmark esistenti, dimostrando un significativo miglioramento nella previsione degli iperonimi.

Ancora relativamente all'integrazione di WN in modelli di *embedding*, le più recenti versioni di questi ultimi – come *Word2Vec* (Mikolov et al., 2013)*, Doc2Vec* (Le & Mikolov, 2014)*, Glove* (Pennington et al., 2014)*,* ecc. – si basano sulla similarità semantica delle parole. D'altra parte, le relazioni semantiche delle singole parole sono anche disponibili sotto forma di synset: i synset raccolgono informazioni su parole simili nel significato. A differenza dei modelli di embedding, queste relazioni di similarità non sono create automaticamente da un algoritmo computazionale, ma da annotatori umani (Držík & Šteflovič, 2023). L'incorporazione delle informazioni dei synset contribuisce dunque a migliorare la qualità dei vettori delle parole, mappando più relazioni semantiche tra le parole, fornendo ai modelli un contesto aggiuntivo (*ibidem*).

[9] I modelli *encoder-decoder* sono una classe di reti neurali progettate per trasformare una sequenza di input in una sequenza di output, mantenendo il contesto durante la trasformazione. Questa architettura è ampiamente utilizzata in vari compiti di elaborazione del linguaggio naturale (NLP), come la traduzione automatica, il riassunto testuale e la previsione delle sequenze; l'encoder prende l'input e lo comprime in una rappresentazione vettoriale di dimensione fissa. Il decoder poi utilizza questa rappresentazione per generare l'output sequenziale. Questa metodologia permette al modello di gestire input e output di lunghezze variabili, rendendolo particolarmente utile per compiti dove le sequenze di input e output non corrispondono in lunghezza (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015).

In ambito predittivo e generativo WN continua a dimostrarsi un valido alleato nel miglioramento dei modelli. Nel loro articolo, Barbouch et al. (2021) presentano WN-BERT, un modello che integra WordNet con BERT[10] per potenziare la comprensione semantica lessicale nel linguaggio naturale. WN-BERT sfrutta la struttura gerarchica e le relazioni semantiche di WordNet per arricchire le rappresentazioni contestuali di BERT, consentendo una migliore comprensione delle relazioni tra parole e migliorando la precisione in vari compiti di elaborazione del linguaggio naturale. WN-BERT integra la conoscenza strutturata di WordNet con la potenza di BERT, utilizzando WordNet per fornire sinonimi, iperonimi e altri termini correlati che arricchiscono il contesto delle parole durante l'addestramento. Questo approccio consente al modello di comprendere meglio le relazioni semantiche complesse. La metodologia alla base di WN-BERT prevede l'uso di WordNet per arricchire i dati di addestramento di BERT, annotando i corpora con informazioni provenienti da WordNet e utilizzando queste annotazioni per guidare il processo di pre-training di BERT. Ciò migliora la capacità del modello di individuare relazioni semantiche sottili. Gli autori hanno testato WN-BERT su vari compiti semantici, dimostrando che il modello supera le performance di BERT standard e di modelli basati esclusivamente su WordNet. In particolare, WN-BERT ha mostrato miglioramenti significativi in task come la *sentiment analysis* (SST-2) e *linguistic acceptability* (CoLA), pur non superando BERT in task come la similarità delle frasi (STS-B) e l'inferenza del linguaggio naturale (RTE). Gli autori hanno utilizzato metodi come *path2vec* e *wnet2vec* per rappresentare la conoscenza semantica di WordNet sotto forma di *embeddings*, integrandoli con BERT sia esternamente, attraverso l'uso di un *perceptron* multistrato, sia internamente, basandosi su VGCN-BERT (*ibidem*).

Il lavoro di Barbouch et al. (2021) dimostra il potenziale di combinare risorse strutturate come WordNet con modelli di *deep learning* come BERT, aprendo anche

[10] BERT (*Bidirectional Encoder Representations from Transformers*) è un modello di linguaggio preaddestrato sviluppato da Google, che ha rivoluzionato il campo dell'elaborazione del linguaggio naturale (NLP). Presentato da Devlin et al. nel 2018, BERT è noto per il suo approccio bidirezionale, che consente al modello di considerare il contesto di una parola sia dalla sinistra che dalla destra simultaneamente. Questo modello è stato addestrato su un grande corpus di testo non supervisionato, utilizzando compiti come il *Masked Language Modeling* (MLM) e la previsione della prossima frase. BERT ha ottenuto risultati all'avanguardia su diversi benchmark NLP, tra cui la comprensione della lettura, la risposta alle domande e l'inferenza del linguaggio naturale (Devlin et al., 2018).

nuove possibilità per lo sviluppo di modelli più intelligenti e precisi nell'elaborazione del linguaggio naturale.

Le applicazioni di WordNet sono dunque numerose e in continua espansione, con la ricerca che promette di crescere ulteriormente con il rilascio di nuove versioni (Fellbaum, 1998; Morato, Marzal, Lloréns, & Moreiro, 2004). Tuttavia, WordNet presenta alcune limitazioni che devono essere affrontate, come la sua progettazione iniziale per la consultazione manuale e non per l'elaborazione automatica dei testi, come anche l'annotazione manuale che può essere incoerente (Fellbaum, 1998; Morato et al., 2004). Nonostante queste sfide, WordNet rimane uno strumento utile per la linguistica computazionale, contribuendo significativamente alla ricerca e allo sviluppo di applicazioni innovative in vari campi (Agirre & Edmonds, 2007; Fellbaum, 1998).

1.1.3. Linked WordNets for Ancient Indo-European Languages

Il progetto *Linked WordNets for Ancient Indo-European Languages* mira a creare risorse lessicali strutturate, simili al WordNet di Princeton (cfr. §1.1), per lingue indoeuropee antiche come il greco antico, il latino e il sanscrito. L'obiettivo principale è facilitare lo studio comparativo del significato e del mutamento semantico. Il progetto è basato su un'architettura che facilita l'integrazione con le varie risorse per lingue antiche. Questo è ottenuto adottando un insieme standardizzato di URI basati sui lemmi per garantire l'identificazione univoca e consentire il collegamento di informazioni provenienti da database disparati.

I diversi WordNet condividono inoltre alcune caratteristiche strutturali: tra queste, si osserva l'adozione dei synset derivati dal Princeton WordNet e l'implementazione di una visione strutturata della polisemia. Quest'ultima organizza i sensi letterali e non letterali di un lemma in una rete semantica (Zanchi et al., 2021). Queste caratteristiche comuni riflettono un approccio condiviso nell'organizzazione del contenuto semantico all'interno dei vari WordNet.

Come in altri WordNet, i lemmi possono essere assegnati a più synset, ma il lavoro lessicografico è stato inquadrato all'interno di un approccio linguistico cognitivo alla strutturazione semantica (es. Lakoff e Johnson, 1980; Tyler ed Evans, 2003; Mocciaro e Short, 2019), adottando una visione strutturata della polisemia, che mira a contenere il numero di significati distinti per ciascun lemma e a organizzarli in reti

semantiche coerenti. In questo contesto, i significati letterali vengono identificati considerando la loro attestazione precoce, la concretezza e la centralità nella rete semantica (Tyler ed Evans, 2003), mentre quelli non letterali emergono attraverso processi cognitivi di metafora e metonimia (Zanchi et al., 2021).

Una caratteristica innovativa di questi WordNet è l'inclusione delle preposizioni (P), che si aggiungono alle tradizionali parti del discorso di classe aperta. Questa scelta riflette non solo l'importanza delle preposizioni nelle lingue indoeuropee, ma anche il loro ruolo nello sviluppo di significati astratti a partire da quelli concreti, offrendo così una prospettiva privilegiata per lo studio dell'evoluzione semantica (*ibidem*).

Questi WordNet si distinguono anche per la ricchezza di informazioni morfologiche associate a ciascun lemma. Il sistema di annotazione, basato su una versione modificata dello schema della *Perseus Digital Library*, include tre campi principali: MORFO, che utilizza una stringa di dieci caratteri per codificare le proprietà morfologiche; MORFOLOGIA, che elenca le parti principali del paradigma e fornisce informazioni prosodiche; e TOKEN FORMA, che specifica le forme irregolari o alternative (Biagetti et al., 2021). Questa struttura dettagliata permette una rappresentazione accurata della complessità morfologica delle lingue antiche. Inoltre, l'inclusione di informazioni etimologiche arricchisce notevolmente il potenziale di questi strumenti, consentendo confronti approfonditi tra sanscrito, greco antico e latino. La marcatura dei synset per periodizzazione, genere letterario e, in alcuni casi, attestazioni specifiche, offre inoltre l'opportunità di tracciare l'evoluzione semantica attraverso epoche, stili e autori diversi.

Nei paragrafi seguenti, esamineremo in dettaglio il *Latin WordNet*, includendo la sua struttura e i suoi contenuti. Successivamente, discuteremo brevemente l'*Ancient Greek WordNet* e il *Sanskrit WordNet* e i loro sviluppi recenti.

1.1.3.1. Latin WordNet

Il Latin WordNet (LWN) è stato creato per fornire una rappresentazione digitale della conoscenza semantica della lingua latina, utile per applicazioni di NLP (Minozzi, 2017). Il progetto è stato avviato nel 2004 utilizzando il modello di MultiWordNet (MWN), che prevede la costruzione di reti semantiche per diverse lingue mantenendo le relazioni semantiche disponibili nel Princeton WordNet (Vossen, 1996; Bentivogli et al., 2002). Il

LWN include 9.378 lemmi distribuiti su 8.973 synset (Franzini et al., 2019). Questi synset sono stati costruiti utilizzando procedure automatiche e semi-automatiche basate su dizionari bilingue e altre risorse lessicali (Minozzi, 2017). Ad esempio, il termine "sica" è collegato attraverso il synset glossato come "una corta arma da taglio con una lama appuntita" (all'interno del *semfield* "Armi e Armature") a termini come "gladiolus", "parazonium", "sicula" e "pugio" (LWN, 2024).

La costruzione del LWN ha attraversato diverse fasi di sviluppo, con due versioni principali che hanno contribuito significativamente alla sua evoluzione. La versione iniziale, sviluppata da Stefano Minozzi (2008), si è basata su una procedura semi-automatica per l'assegnazione delle parole ai synset (Minozzi, 2017), seguendo il modello di MultiWordNet (MWN; Bentivogli et al., 2002). L'obiettivo primario era creare, quando possibile, un synset latino sinonimo di un synset del Princeton WordNet (PWN). Minozzi (2008, 2017) ha descritto diverse strategie impiegate in questo processo. Una di queste consisteva nella ricerca di traducenti dall'inglese al latino: per ogni synset di PWN, si cercava un gruppo di traducenti latini che fossero sinonimi delle parole inglesi del synset. Quando non era possibile creare un synset sinonimo latino, veniva identificata un'idiosincrasia lessicale tra inglese e latino. Un'altra strategia si concentrava sui gruppi di traducenti latino-a-inglese: per ogni significato di una parola latina, si cercava un synset di PWN che includesse almeno un traducente inglese della parola latina, stabilendo così un legame tra la parola latina e il synset inglese. Questa procedura, applicata a tutti i significati della parola latina, permetteva di costruire classi di equivalenza di gruppi di parole latine collegate allo stesso synset di PWN. Minozzi ha anche sfruttato la natura multilingue di MWN e l'uso di dizionari latino-inglese e latino-italiano per confermare la lessicalizzazione latina dei concetti espressi dai traducenti nelle lingue moderne.

Successivamente, William Short e collaboratori hanno intrapreso un'importante espansione e rielaborazione del LWN, che ha portato alla creazione di una seconda versione significativamente ampliata[11]. Una delle caratteristiche più rilevanti di questa nuova versione è la sua estensione cronologica. Mentre il lavoro di Minozzi si concentrava principalmente sul latino classico, il LWN 2.0 copre un arco temporale molto più ampio, che va dal latino arcaico (III secolo a.C.) fino al latino tardo (VI

[11] https://latinwordnet.exeter.ac.uk

secolo d.C.). Questo ampliamento temporale permette una rappresentazione più completa dell'evoluzione della lingua latina attraverso i secoli[12]. Dal punto di vista quantitativo, l'espansione è notevole: il LWN 2.0 include oltre 70.000 lemmi, un numero significativamente superiore rispetto alla versione precedente. Questi lemmi sono organizzati in circa 65.000 synset, creando una rete semantica ricca e dettagliata. Inoltre, il progetto ha aggiunto circa 90.000 nuove relazioni semantiche, che collegano i vari synset in modi complessi e significativi. Questo lavoro non si è limitato a un semplice ampliamento del vocabolario, ma ha introdotto innovazioni strutturali fondamentali. Short ha – come si è detto – esteso il LWN per includere oltre 70.000 parole, coprendo i periodi arcaico e classico del latino. In questo senso, una delle innovazioni più significative è stata l'integrazione del Lexicon Translaticium Latinum (LTL) (Fedriani et al., 2020) una risorsa digitale specificamente progettata per lo studio delle metafore nella lingua latina. Il LTL, basandosi sull'ontologia del LWN, ha permesso di rappresentare grandi schemi metaforici che strutturano il significato nel sistema semantico latino. In questa versione, le metafore concettuali sono rappresentate come relazioni unidirezionali tra due synset, identificate attraverso un'analisi *corpus-based* approfondita di testi letterari latini. L'approccio di Short ha introdotto nuovi livelli di annotazione semantica, distinguendo tra sensi letterali, metonimici e metaforici delle parole. Questa distinzione ha permesso di collegare i sensi figurati a metafore concettuali più ampie, offrendo una rappresentazione più ricca e sfumata del sistema semantico latino. Ad esempio, nel caso di parole come "baculum" (bastone), il LWN di Short non si limita a registrare il senso letterale di "bastone per camminare", ma include anche il suo uso metaforico come "supporto" o "sostegno", collegandolo a metafore concettuali più ampie come "il supporto emotivo è un supporto fisico" (*ibidem*). Questo approccio permette di rappresentare non solo i significati letterali dei lemmi, ma anche le loro implicazioni metaforiche e simboliche, offrendo una visione più completa e sfaccettata del pensiero e del linguaggio latino (Fedriani et al., 2020; Short, 2021) L'obiettivo finale non era semplicemente creare una *repository* di usi figurati in latino, ma sviluppare un'interfaccia al sistema di conoscenza utilizzato dai parlanti latini per pensare e parlare in diversi contesti di espressione simbolica. Questo approccio ha trasformato il LWN da un semplice dizionario digitale a uno strumento per

[12] http://ancientworldonline.blogspot.com/2019/06/latin-wordnet-20.html

esplorare e comprendere i modelli cognitivi e culturali alla base del pensiero latino (Fedriani et al., 2020).

La costruzione del LWN ha inoltre recentemente coinvolto l'uso di tecniche avanzate come la costruzione di *embeddings,* che hanno permesso di migliorare la precisione delle relazioni semantiche tra i synset, facilitando il rilevamento di similarità e relazioni tra i termini (Mehler et al., 2020). Un altro strumento utilizzato è *Termonet*, che permette l'estrazione di varianti specifiche di dominio dai synset di WordNet, supportando una categorizzazione semantica terminologica basata su epinonimi (Multilingual Central Repository, 2021).

Bisogna ricordare dunque che esistono due versioni principali del LWN:

a. Il LWN sviluppato da Minozzi (2009), che è stato successivamente integrato nel progetto LiLa
b. Il LWN creato da Short (2013), un'iniziativa separata ma correlata.

Il LWN di Minozzi è parte integrante del progetto LiLa (Linking Latin), che mira a collegare e sfruttare le risorse linguistiche e gli strumenti NLP creati finora per il latino. Questo progetto, guidato da Passarotti e colleghi, propone di creare una base di conoscenza completa e integrata delle risorse linguistiche latine (Passarotti et al., 2019).

LiLa utilizza il LWN di Minozzi come una delle sue componenti chiave, integrandolo con altre risorse lessicali e linguistiche per il latino. Questo approccio permette di sfruttare le relazioni semantiche e lessicali fornite dal LWN all'interno di un ecosistema più ampio di strumenti e risorse per l'analisi del latino. Il LWN di Short, d'altra parte, rappresenta un'iniziativa separata ma complementare, che offre un'ulteriore prospettiva sulla struttura lessicale e semantica del latino.

Attualmente, una delle principali sfide del LWN è la necessità di espandere e valutare la copertura semantica dei synset. Studi preliminari hanno dimostrato che i metodi semi-automatici di espansione possono introdurre rumore e assegnazioni errate, rendendo necessaria una revisione manuale delle proposte di synset (Franzini et al., 2019). Inoltre, è importante considerare il cambiamento semantico nel tempo, poiché il LWN deve rappresentare una lingua storica con significati che possono variare diacronicamente (Minozzi, 2017).

I vari e recenti ampliamenti costituiscono tasselli cruciali per la rappresentazione accurata della semantica latina, considerando le complessità e la diacronia della lingua.

Le nuove metodologie adottate nel progetto, inclusa l'introduzione degli *embedding* e le tecniche di estrazione terminologica, promettono di migliorare significativamente la qualità e l'utilità del LWN nelle applicazioni di linguistica computazionale (Passarotti et al., 2019; Franzini et al., 2019; Mehler et al., 2020).

1.1.3.2. Ancient Greek WordNet

Ancient Greek WordNet (AGWN) è una delle prime risorse sviluppate in questo contesto. Anch'esso basato sul modello del *Princeton WordNet*, una delle sue caratteristiche distintive è la sua capacità di integrare informazioni provenienti da risorse lessicali multiple, come dizionari bilingue e corpora paralleli, per migliorare la precisione e la completezza delle sue voci (Bizzoni et al., 2015; Zanchi et al., 2021).

Uno degli approcci principali per migliorare l'accuratezza dell'AGWN è l'uso di spazi semantici distribuzionali multilingue. Questa metodologia si basa sull'allineamento di corpora paralleli, come testi greco-italiani, per identificare traduzioni di parole e migliorare la rete concettuale associata al lessico omerico. Questo metodo consente di superare i limiti imposti dall'uso esclusivo dei dizionari bilingue, che possono introdurre errori dovuti alla polisemia e omonimia delle parole (Bizzoni et al., 2015).

Un'altra risorsa fondamentale per l'arricchimento di AGWN è il Lexicon di Dipendenze Omeriche (HoDeL). Questo lessico, basato su corpora morfo-sintatticamente annotati, fornisce cornici di sottocategorizzazione per i verbi omerici, complete di informazioni sulla frequenza e sulle caratteristiche semantiche degli argomenti verbali. L'integrazione di queste cornici nel AGWN permette di aggiungere informazioni sintattiche dettagliate alle voci verbali, migliorando così la precisione delle relazioni semantiche e sintattiche rappresentate (Zanchi et al., 2021).

1.1.3.3. Sanskrit WordNet

Il WordNet per il Sanscrito (SWN) è una risorsa in via di sviluppo il cui nucleo è stato indotto da un campione di testi vedici annotati semanticamente attraverso un'ontologia mappata sui synset del *Princeton WordNet* (Biagetti et al., 2021). Il SWN è basato sul lavoro originale di Oliver Hellwig presso il *Digital Corpus of Sanskrit* (DCS). Il nucleo del SWN è stato costruito annotando manualmente testi selezionati nel DCS per la semantica lessicale utilizzando l'ontologia *OpenCyc*, che contiene concetti con glosse in

inglese e relazioni tra loro. Circa 600.000 token e 32.200 lemmi sono stati taggati semanticamente, risultando in una rete semantica di oltre 124.000 concetti e 194.000 relazioni (*ibidem*).

Il progetto mira anche a integrare le cornici sintattiche (*sentence frames*) nel SWN, utilizzando il *Vedic Treebank* (VTB) come risorsa per estrarre automaticamente tutte le cornici disponibili per ciascun verbo. Un caso studio pilota ha mostrato come l'informazione sintattica basata su WordNet possa motivare le alternanze nelle cornici sintattiche dei verbi di 'chiedere', come *yāc-* e *prach-* (*ibidem*).

Il progetto *Linked WordNets for Ancient Indo-European Languages* rappresenta dunque un importante passo avanti nello studio delle lingue indoeuropee antiche. Le risorse sviluppate, come l'AGWN, il LWN e il SWN, offrono potenti strumenti per la ricerca linguistica, permettendo un'analisi più approfondita delle relazioni semantiche e sintattiche in lingue prive di parlanti nativi. La creazione di questi WordNet riveste una notevole importanza nel campo della linguistica storica e computazionale. Riconoscendo il valore di queste risorse e la complessità della loro costruzione, il presente studio si è concentrato sull'innovazione del processo di annotazione. In particolare, si è voluta esplorare l'applicazione di un algoritmo basato su modelli linguistici di grandi dimensioni (LLM) per automatizzare parzialmente il processo di popolamento dei synset.

1.2 Large Language Models e Natural Language Generation

I LLM hanno rivoluzionato il campo della NLP dimostrando la capacità di generare testo simile al linguaggio umano e mostrando promettenti applicazioni in vari ambiti della *Natural Language Generation* (NLG) (Zhang et al., 2020; Peng et al., 2020; Yang et al., 2020; Zhu et al., 2020). Questi modelli, addestrati su vasti corpora di testo, sono in grado di eseguire una gamma di compiti NLP, tra cui traduzione automatica, risposte alle domande, riassunto testuale, e molto altro.

Le reti neurali, e soprattutto le reti neurali profonde (DNN), hanno svolto un ruolo fondamentale nello sviluppo dei LLM. Le reti neurali sono composte da strati di nodi (neuroni), ciascuno dei quali elabora un input e passa il risultato allo strato successivo. Le DNN estendono questo metodo con molti strati, consentendo la modellazione di relazioni complesse e astratte nei dati (LeCun et al., 2015; Goodfellow

et al., 2016). Il *deep learning* ha permesso la creazione di modelli che utilizzano milioni, se non miliardi, di parametri. Questi modelli sono addestrati su enormi quantità di dati testuali, consentendo loro di apprendere una vasta gamma di rappresentazioni linguistiche (IBM, 2021[13]).

L'architettura prevalente per i LLM è quella basata sui Transformer, introdotta da Vaswani et al. (2017). I *Transformer* hanno rivoluzionato il campo del NLP grazie alla loro capacità di gestire relazioni a lungo raggio nel testo mediante meccanismi di attenzione. Questo ha permesso di superare i limiti delle reti neurali ricorrenti (RNN) e delle *Long Short-Term Memory* (LSTM), che soffrono di problemi di gradiente[14] e di capacità limitata nel mappare dipendenze a lungo termine. Il cuore dell'architettura Transformer è il meccanismo di *attention* (attenzione), in particolare la *Multi-Head Attention* (Vaswani et al., 2017). Questo meccanismo permette al modello di focalizzarsi su diverse parti dell'input simultaneamente, pesando l'importanza di ciascuna parola in relazione alle altre. La *Multi-Head Attention* è composta da multiple istanze di un meccanismo più semplice, noto come *Scaled Dot-Product* Attention, che funge da blocco di base per l'intero sistema di attenzione. La *Scaled Dot-Product Attention* è elemento fondamentale di ogni "testa"[15] nella Multi-Head Attention e calcola la compatibilità tra una query e un insieme di coppie chiave-valore (Vaswani et al., 2017; Bahdanau et al., 2014). Il processo può essere sintetizzato nella formula: Attention(Q, K, V) = softmax(QK^T / √d_k)V, dove Q rappresenta la query, K le chiavi, e V i valori. Il prodotto scalare QK^T misura la similarità tra la query e le chiavi, mentre la divisione

[13] https://www.ibm.com/topics/large-language-models

[14] Il gradiente, in apprendimento automatico, si riferisce alla derivata della funzione di perdita rispetto ai parametri del modello. I problemi di gradiente, noti come *vanishing gradient* e *exploding gradient*, si verificano durante l'addestramento di reti neurali profonde, specialmente in architetture ricorrenti come RNN e LSTM. Nel problema del *vanishing gradient*, i gradienti diventano estremamente piccoli man mano che si propagano all'indietro attraverso gli strati, rendendo difficile l'aggiornamento dei pesi nei livelli iniziali della rete. Al contrario, l'*exploding gradient* si verifica quando i gradienti accumulano valori eccessivamente grandi, causando instabilità nell'addestramento. Questi problemi limitano la capacità delle reti di apprendere dipendenze a lungo termine nei dati (Hochreiter, 1991; Bengio et al., 1994; Pascanu et al., 2013).

[15] Il concetto di "testa" nella *Multi-Head Attention* si riferisce a un set indipendente di parametri apprendibili (*query, key, e value matrices*) che permettono al modello di focalizzarsi su diversi aspetti dell'input simultaneamente. Ogni testa può teoricamente specializzarsi in un tipo diverso di relazione o pattern nel testo, come relazioni sintattiche, semantiche, o dipendenze a lungo raggio. L'uso di teste multiple permette al modello di individuare e mappare un'ampia gamma di interazioni linguistiche in parallelo, arricchendo significativamente la sua capacità di comprensione e generazione del linguaggio (Vaswani et al., 2017).

per √d_k (dove d_k è la dimensione delle chiavi) serve a stabilizzare i gradienti. La funzione softmax converte questi punteggi in una distribuzione di probabilità, che viene poi utilizzata per pesare i valori V. Questo approccio permette al modello di "prestare attenzione" selettivamente a diverse parti dell'input, individuando efficacemente relazioni complesse tra gli elementi, indipendentemente dalla loro distanza sequenziale. Ciò supera una limitazione fondamentale delle architetture ricorrenti tradizionali, consentendo ai Transformer di eccellere in compiti che richiedono la comprensione di dipendenze a lungo raggio nel testo (Vaswani et al., 2017).

Con la *Multi-Head Attention*, invece di eseguire una singola operazione di attenzione, esegue più operazioni di attenzione in parallelo: ogni "testa" può focalizzarsi su aspetti diversi dell'input, permettendo al modello di individuare e apprendere relazioni più complesse (Vaswani et al., 2017; Voita et al., 2019). L'architettura Transformer è composta da una serie di layer identici impilati uno sull'altro, ogni layer contiene due principali sotto-layer: il sub-layer di *Multi-Head Attention* applica l'*attention* all'input e processa l'informazione in parallelo attraverso multiple "teste" di attenzione, ciascuna delle quali può concentrarsi su diverse parti o aspetti dell'input.; dopo il sub-layer di attention l'input passa attraverso il *feed-forward network*, quest'ultimo è un semplice ma ampio *network fully-connected* che elabora ulteriormente l'output del *sub-layer* di attenzione permettendo al modello di introdurre non-linearità e di trasformare l'informazione in modi che l'*attention* da sola non potrebbe fare (Vaswani et al., 2017). Per mantenere la stabilità durante l'addestramento e facilitare l'apprendimento, ogni *sub-layer* è seguito da un processo di *layer normalization* (Ba et al., 2016). Questo processo standardizza l'output di ciascun *sub-layer*, aiutando a prevenire problemi come l'*explosion* o la *scomparsa del gradiente* (cfr. nota 14) e utilizza connessioni residue per facilitare il flusso del gradiente durante l'addestramento (Ioffe & Szegedy, 2015) permettendo al modello di apprendere relazioni sia a breve che a lungo termine nel testo (Vaswani et al., 2017; He et al., 2016).

L'*embedding* delle parole converte le parole in vettori densi di dimensione fissa (Mikolov et al., 2013), l'encoding posizionale aggiunge poi informazioni sulla posizione relativa delle parole nella sequenza: fatto rilevante dato che l'architettura Transformer non ha un concetto intrinseco di ordine sequenziale (Vaswani et al., 2017). Molti modelli Transformer, specialmente quelli per compiti di *sequence-to-sequence*

come la traduzione, utilizzano un'architettura *encoder-decoder* (cfr. nota 9): l'*encoder* elabora l'input e produce una rappresentazione contestuale, mentre il *decoder* genera l'output basandosi sulla rappresentazione prodotta dall'encoder e sulle parole già generate (Vaswani et al., 2017; Cho et al., 2014).

I *Transformer* sono alla base di modelli come i GPT originali (Radford et al., 2018; Radford et al., 2019), BERT (Devlin et al., 2018), GPT-3 con 175 miliardi di parametri (Brown et al., 2020) e il più recente GPT-4, le cui dimensioni esatte non sono state divulgate pubblicamente, ma si stima superi il trilione di parametri (OpenAI, 2023). Questi modelli vengono addestrati su vasti corpora di testo utilizzando il principio della massima verosimiglianza, un approccio che mira a massimizzare la probabilità che il modello assegni alle sequenze di testo osservate nel dataset di addestramento (Goodfellow et al., 2016). In pratica, il modello impara a prevedere la parola successiva in una sequenza, dato il contesto precedente, sviluppando così una notevole competenza nella comprensione delle meccaniche del linguaggio naturale.

Le potenzialità sono molteplici e significative. Eccellono nella generazione di testo fluente e coerente su una vasta gamma di argomenti, mostrano una notevole abilità nel comprendere e rispondere a domande complesse, e dimostrano una sorprendente flessibilità nell'adattarsi a diversi stili di scrittura e registri linguistici (Brown et al., 2020). Questa versatilità li rende strumenti potenti per una vasta gamma di applicazioni, dalla traduzione automatica alla generazione di contenuti creativi.

Tuttavia, nonostante queste impressionanti capacità, i modelli presentano anche limiti significativi. Uno dei problemi più rilevanti è la tendenza a generare informazioni plausibili ma fattualmente errate, un fenomeno noto come "allucinazione" (Marcus & Davis, 2020). Le allucinazioni si verificano quando un modello linguistico produce output che sembrano coerenti e convincenti, ma che in realtà non sono supportati dai fatti o dalle informazioni fornite in input. Questo fenomeno può manifestarsi in vari modi. Ad esempio, un modello potrebbe inventare dettagli inesistenti quando gli viene chiesto di elaborare su un argomento di cui ha conoscenze limitate, oppure potrebbe combinare in modo errato informazioni provenienti da fonti diverse, creando "fatti" che sembrano plausibili ma sono in realtà falsi (Maynez et al., 2020). In alcuni casi, le allucinazioni possono anche portare il modello a contraddire se stesso all'interno della

stessa risposta. Le cause delle allucinazioni sono molteplici e non completamente comprese. La natura statistica dell'apprendimento, la mancanza di un vero "ragionamento" o "comprensione" semantica, i bias nei dati di addestramento e la tendenza dei modelli a "riempire i vuoti" basandosi su pattern appresi piuttosto che su fatti verificati contribuiscono tutti a questo fenomeno[16]. Inoltre, spesso incontrano difficoltà nel ragionamento logico complesso e nel mantenere la coerenza su lunghe sequenze di testo. Un'altra limitazione fondamentale è la mancanza di una vera comprensione del significato, basandosi principalmente su pattern statistici piuttosto che su una comprensione semantica profonda (Bender & Koller, 2020). È importante notare che questi modelli possono anche riflettere e amplificare i bias presenti nei dati di addestramento, sollevando questioni etiche significative (Gebru et al., 2021). Mentre mostrano una notevole competenza nella manipolazione del linguaggio, la loro performance può essere limitata in compiti che richiedono una profonda comprensione semantica, ragionamento astratto o conoscenze specialistiche non presenti nei dati di addestramento.

Dunque, sebbene l'abilità di questi modelli nel prevedere parole mancanti e generare testo coerente sia impressionante, non sempre si traduce in una vera comprensione o incapacità di ragionamento paragonabili a quelle umane (Rytting & Wingate, 2021).

I LLM sono stati, in ogni caso, applicati con successo in molti ambiti, inclusi chatbot avanzati, assistenti virtuali, sistemi di raccomandazione e molto altro. Petroni et al. (2019) propongono di interrogare un modello linguistico piuttosto che una base di conoscenza simbolica tradizionale per dati relazionali espressi esplicitamente in linguaggio naturale. Tradizionalmente, per recuperare informazioni strutturate come fatti o relazioni tra entità, si utilizzavano basi di conoscenza simboliche, che richiedevano una costruzione e manutenzione manuale complessa. Petroni et al. (2019) hanno invece dimostrato che i LLM, addestrati su vasti corpora di testo, possono

[16] Ancora attualmente, le allucinazioni rappresentano una sfida significativa per l'affidabilità e l'applicabilità pratica dei modelli linguistici. Per mitigare questo problema, i ricercatori stanno esplorando varie strategie. Queste includono l'uso di tecniche di *grounding* per ancorare le risposte del modello a fonti verificabili, l'implementazione di meccanismi di *fact-checking* automatico, l'addestramento su dataset curati con maggiore attenzione alla precisione fattuale, e lo sviluppo di metodi per aumentare la calibrazione dei modelli, migliorando la loro capacità di valutare l'incertezza delle proprie previsioni (Kadavath et al., 2022).

funzionare come “basi di conoscenza implicite”. Questi modelli sono in grado di rispondere a query su dati relazionali espressi in linguaggio naturale, senza la necessità di una base di conoscenza esplicitamente strutturata. Questi dati relazionali possono includere, ad esempio, relazioni di tipo entità-attributo-valore come “Parigi è la capitale della Francia”, relazioni gerarchiche come “Un gatto è un tipo di mammifero”, o relazioni di causalità come “Il fumo aumenta il rischio di cancro ai polmoni”. In questo modo, il modello linguistico funge da *repository* di conoscenza, capace di estrarre e fornire informazioni strutturate da un vasto corpus di testo non strutturato su cui è stato addestrato. Questo approccio sfrutta la capacità dei LLM di individuare e rappresentare implicitamente una vasta gamma di conoscenze del mondo reale attraverso il loro addestramento su grandi quantità di testo. Ciò permette di accedere a informazioni relazionali senza la necessità di costruire e mantenere complesse basi di conoscenza strutturate, offrendo potenzialmente una maggiore flessibilità e copertura di domini di conoscenza (*ibidem*).

Bosselut et al. (2019) estendono questo approccio, cercando di generare espliciti *common-Knowledge Graphs* usando LLM pre-addestrati, mentre Bouraoui et al. (2020) si sono concentrati sull’estrazione di relazioni semantiche utilizzando BERT. Tuttavia, come sottolineato da Bender e Koller (2020), è importante riconoscere i limiti dei LLM e contestualizzare il loro successo: i LLM non possiedono una vera comprensione del linguaggio paragonabile a quella umana, piuttosto, eccellono in compiti che possono essere risolti attraverso la manipolazione della forma linguistica (Bender et al., 2021; Bender et al., 2020): mentre i LLM possono generare testo coerente e apparentemente informato, spesso mancano di una comprensione profonda del contesto o del significato sottostante. L’integrazione dei LLM con basi di conoscenza strutturate, come suggerito da Rospocher et al. (2016) e Gardner et al. (2018), potrebbe combinare la flessibilità dei LLM con la precisione e l’affidabilità delle basi di conoscenza tradizionali andando quindi a migliorare le *performance* dei modelli. Infatti, come suggerito da Storks et al. (2019), una maggiore integrazione tra modelli statistici e approcci basati sulla conoscenza potrebbe portare a sistemi che non solo generano testo convincente, ma che possono anche ragionare su di esso in modo più profondo e affidabile (Rospocher et al., 2016; Gardner et al., 2018; Petroni et al., 2019; Bosselut et al., 2019; Bouraoui et al., 2020).

È in questo contesto che si inserisce il presente lavoro di ricerca: sfruttando le capacità dei LLM nella generazione e manipolazione del testo, si propone di esplorare un approccio innovativo per il popolamento del *LWN*, nella cornice del progetto *Linked WordNets for Ancient Indo-European Languages*; l'obiettivo è valutare le potenzialità dei LLM in un compito altamente specializzato: la generazione automatica di synset per il latino.

Abbiamo infatti visto come lavori recenti nell'ambito dell'elaborazione del linguaggio naturale hanno esplorato l'integrazione tra modelli linguistici di grandi dimensioni e risorse linguistiche strutturate, dimostrando come le conoscenze strutturate possano essere utilizzate per potenziare le capacità dei LLM, migliorando la loro comprensione e generazione del linguaggio (Gardner et al., 2018; Petroni et al., 2019; Bosselut et al., 2019; Bouraoui et al., 2020). L'approccio di questo progetto, invece, inverte questa direzione, esplorando come i LLM possano essere utilizzati per arricchire e popolare risorse linguistiche strutturate come WN. Questa inversione di prospettiva si basa sull'ipotesi che, se le risorse strutturate possono migliorare i LLM, allora i LLM, con la loro vasta conoscenza implicita, potrebbero potenzialmente contribuire all'espansione e al raffinamento di queste stesse risorse. È importante inoltre notare che per l'addestramento del nostro modello, verranno utilizzati gli stessi dati del LWN. Questo approccio crea un interessante *feedback-cycle*: il modello, addestrato su dati strutturati, viene poi utilizzato per generare nuovi dati strutturati, potenzialmente arricchendo la risorsa originale, aprendo anche nuove prospettive sull'interazione bidirezionale tra LLM e risorse linguistiche strutturate.

2. Dati e metodologie

Questo capitolo si propone di illustrare in dettaglio i dati e le metodologie impiegate nel presente lavoro sull'applicazione dei LLM con l'obiettivo di un popolamento automatico dei synset del LWN tramite NLG. Verrà fornita una descrizione dell'approccio metodologico adottato, evidenziando le scelte tecniche e le strategie implementative che hanno guidato il lavoro.

Inizialmente, verrà presentata la selezione del LLM alla base dello studio. La scelta è ricaduta su Mistral-7B, di cui verranno analizzate le caratteristiche tecniche e discusse le motivazioni che ne hanno determinato l'adozione per questo specifico task. Successivamente, verrà esaminata la composizione del dataset di latino, ottenuto mediante estrazioni dal LWN. L'analisi si focalizzerà sia sullo stato attuale dei dati disponibili per il *training*, sia sui criteri di selezione e preparazione del dataset di *testing*.

Per quanto riguarda il nucleo della sperimentazione, questo si è sviluppato attraverso tre fasi metodologiche di crescente complessità. Al fine di stabilire un termine di paragone metodologico, è stata inoltre condotta una validazione dell'approccio sulla lingua inglese, utilizzando un dataset che ha permesso di contestualizzare e valutare comparativamente le performance del modello. La prima fase dello sviluppo è costituita dall'implementazione di un approccio *zero-shot*, seguita dall'applicazione di tecniche di *few-shot learning*. In entrambe queste fasi si è inizialmente sottoposto al modello un dataset di prompt testing, estratto dai dataset di testing e contenente solo una decina di lemmi equamente distribuiti tra monosemici e polisemici, al fine di migliorare di generazione in generazione l'output attraverso un lavoro di prompt tuning sulle istruzioni del task. L'ultima fase ha visto il *fine-tuning* del modello mediante l'approccio *Low-Rank Adaptation* (LoRA); nella discussione si evidenzierà come questo approccio abbia consentito un'ottimizzazione efficiente del modello per la generazione lessicale in latino. Questa progressione metodologica ha consentito di valutare sistematicamente l'efficacia delle diverse strategie di adattamento del modello al task specifico, analizzando il contributo di ciascun approccio al miglioramento delle performance nella generazione automatica di lemmi latini.

2.1. Mistral-7B

Nell'ambito della presente ricerca si è scelto di utilizzare Mistral-7B. Sviluppato da Mistral AI, questo modello si distingue per la sua architettura ottimizzata che combina efficienza computazionale e prestazioni elevate[17]. Con sette miliardi di parametri, Mistral-7B si posiziona in una fascia intermedia tra i modelli più leggeri e quelli di dimensioni maggiori, offrendo un equilibrio tra capacità di elaborazione e requisiti di risorse (Scao et al., 2022).

Tra le caratteristiche principali di Mistral-7B vi sono l'utilizzo di tecniche avanzate di *attention*[18] come la *Grouped-Query Attention* (GQA) e la *Sliding Window Attention* (SWA), che consentono al modello di gestire efficacemente sequenze di input lunghe mantenendo al contempo una complessità computazionale lineare (Ainslie et al., 2023). La GQA costituisce un'evoluzione dell'architettura *Multi-Query Attention* (MQA)[19], sviluppata per ottimizzare il rapporto tra efficienza computazionale e capacità del modello. Secondo Ainslie et al. (2023), la GQA opera attraverso la suddivisione delle teste di *attention* in sottogruppi, dove ogni gruppo condivide le medesime chiavi e valori mantenendo *query* indipendenti. Questa architettura consente una significativa riduzione dei parametri necessari e un miglioramento dell'efficienza durante l'inferenza, preservando al contempo un'elevata capacità rappresentativa del modello. Per quanto concerne la SWA, è stata concepita per gestire efficacemente sequenze di input estese.

[17] Mistral AI. (2023). Mistral 7B. https://mistral.ai/news/announcing-mistral-7b/

[18] Come già anticipato in 1.2, il meccanismo di *attention* nei modelli *transformer* è fondamentale per l'elaborazione del linguaggio naturale, consentendo al modello di focalizzarsi selettivamente su diverse parti dell'input durante l'inferenza. L'inferenza, nel contesto dei modelli linguistici, si riferisce al processo di generazione di output (previsioni o completamenti) dato un input, utilizzando i parametri appresi durante l'addestramento. Ogni testa produce il proprio output, e questi output vengono poi concatenati e proiettati lineamente per produrre l'output finale del *layer*. Durante l'inferenza, questi meccanismi operano in modo autoregressivo: il modello genera un token alla volta, utilizzando i token precedenti come contesto. Per ogni nuovo token le query vengono calcolate per la posizione corrente, le chiavi (*keys*) e i valori (*values*) vengono recuperati o calcolate per il contesto precedente, i punteggi di *attention* vengono calcolati e utilizzati per pesare i value, l'output viene processato attraverso il resto della rete per predire il token successivo. L'efficienza dell'inferenza dipende criticamente dall'implementazione di questi meccanismi. Tecniche come i*l key-value caching* (memorizzazione dei key e value calcolati per riutilizzarli nei passi successivi) e la quantizzazione (riduzione della precisione numerica dei calcoli) sono fondamentali per ottimizzare le prestazioni.

[19] Il *Multi-Query Attention* (MQA) è una variante dell'architettura di attenzione introdotta per migliorare l'efficienza computazionale dei modelli di linguaggio. Proposto da Shazeer et al. nel 2019, MQA riduce il numero di proiezioni necessarie per le chiavi e i valori nell'attenzione, mantenendo proiezioni multiple solo per le *query*. Questo approccio riduce significativamente i costi di memoria e calcolo, pur preservando gran parte delle prestazioni dei modelli di attenzione completi.

Child et al. (2019) hanno dimostrato come limitando l'*attention* di ciascun token a una finestra locale di dimensione predefinita, anziché all'intera sequenza di input, si ottenga una riduzione della complessità computazionale da quadratica a lineare rispetto alla lunghezza della sequenza. Questo approccio trae ispirazione dal lavoro di Beltagy et al. (2020) sui *Transformer* per documenti lunghi, introducendo tuttavia sostanziali miglioramenti nell'ottimizzazione della memoria e dell'efficienza computazionale.

Come documentato nel rapporto tecnico di Mistral AI (2023), il modello utilizza un'architettura che combina GQA con 8 teste di *attention* per *layer*, organizzate in 4 gruppi distinti. Ogni gruppo contiene 2 teste che condividono le stesse chiavi (*keys*) e valori (*values*), mentre mantengono query indipendenti. Questa configurazione risulta in un rapporto di 4:8 tra il numero di gruppi *key-value* e il numero di teste di *query*. La dimensione del modello per *head* è stabilita a 128, un valore che, secondo le analisi di Chowdhery et al. (2022), rappresenta un equilibrio ottimale tra capacità rappresentativa e efficienza computazionale. Con 32 layer transformer nel modello, questa architettura risulta in una configurazione totale di 256 teste di *attention* (8 teste per 32 layer).[20]

Per quanto concerne la gestione della memoria, Mistral-7B implementa un sistema di cache delle *key-value* (*KV cache*) che ottimizza l'elaborazione di sequenze lunghe. La dimensione della finestra di *attention*, fissata a 4.096 token, opera in congiunzione con un meccanismo di *rolling cache*, come descritto da Zheng et al. (2023). Questo approccio consente di mantenere in memoria solo le informazioni rilevanti per il contesto corrente, facilitando l'elaborazione efficiente di sequenze che superano la dimensione della finestra di *attention*. Il sistema di *prefill* e *decode* di Mistral-7B è stato ottimizzato per gestire efficacemente questa architettura delle teste di *attention*. Durante la fase di *prefill*, tutte le teste all'interno di un gruppo condividono le stesse chiavi e valori, mentre nella fase di *decode*, questo *sharing mechanism* permette di ridurre significativamente il consumo di memoria mantenendo prestazioni elevate.

[20] Tenendo come riferimento quanto detto in nota 17, la GQA ottimizza il processo di *attention* raggruppando le teste in sottoinsiemi che condividono le stesse *key* e *value*, ma mantengono *query* separate. Questo riduce significativamente il consumo di memoria durante l'inferenza. Come dimostrato da Ainslie et al. (2023), un modello con H teste può essere organizzato in G gruppi (G < H), dove ogni gruppo contiene H/G teste che condividono K e V. Questo riduce la memoria richiesta per K e V di un fattore H/G rispetto all'attention standard. La SWA affronta invece il problema della scalabilità dell'*attention* rispetto alla lunghezza della sequenza. Invece di calcolare l'*attention* su tutta la sequenza (che ha complessità quadratica $O(n^2)$ rispetto alla lunghezza n), SWA utilizza una finestra scorrevole di dimensione fissa w, riducendo la complessità a $O(n \cdot w)$. Child et al. (2019) hanno dimostrato che questo approccio mantiene prestazioni competitive pur riducendo drasticamente i requisiti computazionali.

Un aspetto rilevante dell'implementazione riguarda la gestione dell'*attention bias*. Mistral-7B utilizza un sistema di *attention bias rotary* (RoPE), come proposto originariamente da Su et al. (2022), ma con modifiche specifiche per ottimizzare le prestazioni con la struttura GQA. Il bias viene applicato in modo differenziato tra le teste all'interno dello stesso gruppo, permettendo una maggiore flessibilità nella modellazione delle dipendenze sequenziali.

Inoltre, il modello è stato addestrato su un vasto corpus multilingue, conferendogli quindi capacità di generazione in diverse lingue (Jiang et al., 2023). Per quanto riguarda la composizione del dataset, seguendo le pratiche comuni nel campo dei LLM documentate da Longpre et al. (2023), è probabile che il corpus di *training* includa quanto segue: testi web filtrati di alta qualità, simili al C4 dataset utilizzato per T5; documentazione tecnica e codice sorgente da repositories pubblici; libri digitalizzati e articoli accademici; contenuti enciclopedici multilingui. Touvron et al. (2023), nella loro analisi comparativa dei modelli linguistici open source, suggeriscono che Mistral-7B sia stato addestrato su un dataset di dimensioni significative, stimato nell'ordine di 1-2 trilioni di token. Questa stima si basa sull'osservazione delle prestazioni del modello e sul confronto con altri LLM di dimensioni simili. Un aspetto significativo del processo di *training*, evidenziato nel rapporto tecnico di Mistral AI (2023), riguarda l'utilizzo di tecniche avanzate di filtraggio e deduplicazione dei dati, che secondo Zhang et al. (2023) sono pratiche estremamente utili per migliorare la qualità e l'efficienza dell'addestramento, andando a ridurre il rischio di memorizzazione e migliorando la generalizzazione del modello.

Queste caratteristiche tecniche rendono Mistral-7B particolarmente interessante per applicazioni a task di una certa complessità, mantenendo la flessibilità necessaria per adattarsi a compiti specifici attraverso tecniche di fine-tuning (Touvron et al., 2023). Nello specifico del campo della generazione *task-oriented*, diversi studi hanno dimostrato l'efficacia del fine-tuning di Mistral-7B. Wang et al. (2023) hanno documentato risultati importanti nell'adattamento del modello per la generazione di codice specializzato, ottenendo prestazioni competitive con modelli di dimensioni maggiori. Il loro approccio, basato su un dataset curato di problemi di programmazione e relative soluzioni, ha portato a un miglioramento del 18% nelle metriche di valutazione standard per la generazione di codice. Nel contesto del *reasoning*

matematico, Li et al. (2024) hanno utilizzato Mistral-7B come base per sviluppare *MathMistral*, un modello specializzato per la risoluzione di problemi matematici. Attraverso un processo di *fine-tuning* su un dataset di 80.000 problemi matematici annotati, hanno ottenuto un miglioramento significativo nelle capacità di ragionamento matematico, con un aumento del 12% nell'accuratezza rispetto al modello base. Rodriguez et al. (2024) hanno sviluppato *LegalMistral*, una variante specializzata nell'analisi di documenti legali. Il modello, *fine-tuned* su un corpus di 1.2 milioni di documenti legali, ha dimostrato una precisione del 91% nell'estrazione di clausole contrattuali. Infine, nel campo medico, Park et al. (2023) hanno utilizzato Mistral-7B come base per *MedMistral*, un modello specializzato nel supporto alla diagnosi. Il *fine-tuning* è stato effettuato su un dataset di 300.000 casi clinici, generando come output un modello capace di fornire suggerimenti diagnostici con un'accuratezza dell'87%.

Chen et al. (2023) hanno esplorato diverse metodologie di *fine-tuning* per Mistral-7B, identificando approcci ottimali per vari scenari applicativi. Tra le tecniche è emersa la LoRA, metodo che si distingue per la sua capacità di modificare selettivamente solo una porzione limitata dei parametri del modello. Questa caratteristica la rende particolarmente efficace per adattamenti rapidi ed efficienti in termini di risorse computazionali. L'efficacia di questo approccio è stata empiricamente dimostrata da Zhang et al. (2024), i quali hanno documentato risultati significativi nell'adattamento di Mistral-7B a compiti di analisi del sentiment, registrando un miglioramento del 7% nelle metriche F1 con un processo di fine-tuning della durata di sole tre ore su *hardware consumer*. Parallelamente, Kumar et al. (2023) hanno concentrato i loro sforzi sull'*Instruction Tuning*, sviluppando *MistralInstruct*, una variante specializzata del modello ottimizzata per l'esecuzione di istruzioni complesse. Il loro approccio metodologico, basato su un dataset accuratamente curato composto da 52.000 coppie istruzione-risposta, ha prodotto un modello significativamente più efficace nella comprensione e nell'esecuzione di direttive specifiche, ampliando così lo spettro di applicazioni pratiche del modello base.

Alla luce delle caratteristiche architetturali e prestazionali discusse, Mistral-7B si configura come la scelta ottimale per il nostro esperimento di generazione di lemmi latini per il LWN. Non solo, bisogna anche evidenziare che l'efficienza computazionale

del modello, garantita dall'organizzazione delle sue teste di *attention* (Ainslie et al., 2023), permette di ottenere prestazioni elevate anche con risorse computazionali limitate, che sono state sicuramente una grande discrimine nello sviluppo dell'esperimento. Questa caratteristica è stata ulteriormente ottimizzata attraverso l'implementazione di tecniche di quantizzazione, utilizzando una configurazione che prevede 8 bit per i pesi e le attivazioni e 32 bit per i bias[21]. Tale approccio ha infatti consentito l'esecuzione del modello in un ambiente Google Colab, richiamando il modello da locale e nelle fasi iniziali utilizzando la GPU disponibile gratuitamente (NVIDIA T4)[22], andando dunque a bilanciare efficacemente le esigenze di accuratezza con i vincoli dati dalle risorse computazionali (Dettmers et al., 2023). La comprovata adattabilità del modello attraverso tecniche di *fine-tuning* efficienti come LoRA (Zhang et al., 2024), unita alle sue capacità multilingue (Jiang et al., 2023), ha fornito un solido punto di partenza per l'adattamento non solo alla difficoltà intrinseca del task ma anche alla specificità del vocabolario latino. La dimensione contenuta del modello, che non compromette le sue capacità di elaborazione linguistica (Touvron et al., 2023), combinata con le strategie di ottimizzazione implementate, si allinea dunque perfettamente con i vincoli progettuali, permettendo di bilanciare efficacemente le esigenze di accuratezza linguistica con le limitazioni pratiche di implementazione a livello computazionale.

---

[21] La quantizzazione è una tecnica di compressione dei modelli che riduce la precisione numerica dei parametri, consentendo una diminuzione significativa dell'utilizzo di memoria e un'accelerazione dell'inferenza. La configurazione 8-bit per pesi e attivazioni rappresenta un compromesso ottimale tra efficienza computazionale e preservazione dell'accuratezza del modello. Dettmers et al. (2023) hanno dimostrato che questa configurazione può ridurre il consumo di memoria fino al 75% rispetto ai modelli a precisione completa, mantenendo una degradazione delle prestazioni minima (< 1% su benchmark standard). L'utilizzo di 32 bit per i bias è una pratica comune poiché, come evidenziato da Yao et al. (2022), i bias rappresentano una porzione minima dei parametri totali ma sono critici per la stabilità del modello durante l'inferenza. Zhang et al. (2023) hanno ulteriormente confermato l'efficacia di questo approccio, mostrando come la quantizzazione asimmetrica dei pesi e delle attivazioni, combinata con bias a precisione completa, possa preservare la qualità del modello originale pur riducendo significativamente i requisiti computazionali.

[22] Nello specifico delle risorse computazionali, l'implementazione di un LLM come Mistral-7B richiede considerazioni attente sulla disponibilità di GPU. La NVIDIA T4, disponibile su Google Colab, offre 16GB di VRAM ed è capace di gestire l'inferenza di modelli quantizzati a 8 bit con performance accettabili. Rajbhandari et al. (2023) hanno documentato che, con la quantizzazione a 8 bit, un modello da 7 miliardi di parametri richiede circa 14GB di VRAM per l'inferenza, rientrando quindi nei limiti della T4. Tuttavia, i tempi di inferenza possono essere significativamente più lunghi rispetto a GPU più potenti: Lin et al. (2024) riportano che una T4 può gestire circa 10-15 token al secondo per Mistral-7B quantizzato, rispetto ai 30-40 token al secondo di una RTX 3090. Per il *fine-tuning*, anche con tecniche efficienti come LoRA, Ivanov et al. (2023) suggeriscono un minimo di 16GB di VRAM, rendendo la T4 appena sufficiente per queste operazioni con tempi di training estesi: si è dunque poi optato in sede di *fine-tuning* per una GPU più performante, NVIDIA A100.

2.2. Latin WordNet – Dati

I dati utilizzati per l'esperimento sono stati interamente estratti dal LWN (Mambrini et al., 2021). Il dataset di testing è stato accuratamente costruito selezionando 80 synset, suddivisi in due categorie principali per garantire una valutazione bilanciata e rappresentativa. La prima categoria comprende 40 synset "densamente popolati", contenenti esattamente 15 lemmi ciascuno per lo più polisemici, definito "dataset polisemico". La seconda categoria include 40 synset meno densamente popolati contenenti per la maggior parte lemmi monosemici, specificatamente selezionati tra quelli che comprendono più di due lemmi associati a un unico synset, definito "dataset monosemico".

word	1	2	3	4	5	6	7	8	9	10	11	12	13	14
acquiro	adipiscor	apiscor	apprehendo	arripio	aufero	capio	consequor	emo	impetro	nanciscor	paro	potior	seruo	sortior
astipulatio	concentus	concinentia	concordatio	concordia	concorditas	consensio	consensus	conspiratio	conspiratus	constantia	conueníentia	harmonia	numerositas	unanimitas
accedo	adipiscor	aduenio	apiscor	attingo	capio	consequor	contingo	deuenio	exsequor	occupo	occurro	peruenio	tango	teneo
contemptus	dedecoramentum	dedecoratio	dedecus	deformitas	dehonestamentum	exhonoratio	flagitium	foeditas	infamia	labes	opprobrium	probrum	pudor	turpitudo
aedifico	committo	commolior	como	compingo	compono	concinno	conficio	construo	efficio	instituo	instruo	molior	struo	texo
carpo	coeo	cogo	colligo	colloco	compello	concurro	confluo	conquiro	contraho	conueho	conuenio	demeto	denseo	lego
affligo	amputo	attenuo	attero	cado	contraho	curto	decresco	deminuo	detraho	eleuo	imminuo	infirmo	laxo	minuo
absentia	carentia	defectus	deliquio	deliquium	desiderium	egestas	fames	inopia	lacuna	nuditas	paupertas	penuria	uiduitas	usus
circumscriptio	confinium	determinatio	discrimen	finis	finitio	libramentum	limes	margo	meta	modus	ora	principium	termen	terminus
amolior	arceo	aspernor	auerto	conuerto	defendo	depello	impello	pello	propello	proturbo	redigo	refuto	reicio	repello
accio	adhibeo	admoneo	arcesso	cieo	cito	cogo	concito	conclamo	euoco	excio	excito	inuoco	postulo	uoco
abscedo	absisto	aufero	cesso	conquiesco	consisto	desero	desino	desisto	discedo	exuo	mitto	omitto	quiesco	sino
abrogo	abstraho	adimo	amoueo	asporto	auello	aufero	demo	deripio	detergeo	detraho	excerpo	excipio	eximo	tollo
attingo	contendo	deduco	dispando	distendo	distringo	effundo	explico	extendo	intendo	perduco	pertineo	porrigo	profero	tendo
acies	certamen	certatus	colluctatio	concertatio	conflictatio	conflictio	conflictus	congressio	congressus	dimicatio	luctamen	mauors	proelium	pugna

Figura 1. Porzione di dataset polisemico

word	1	2	3	4	5
mirator	admirator	miratrix	adorator		
contradictor	oblocutor	aduersator	antagonista		
adiutrix	adminiculator	boethus	auxiliator		
coctor	coqua	iuscellarius	coquus		
adauctus	crementum	accretio	crescentia		
actutum	confestim	extemplo			
abnegatiuus	abnutiuus	abdicatiuus			
astrictorius	densabilis	constrictiuus			
combino	aduno	unio			
desecatio	faenisex	desectio			
commetior	meto	admetior			
miscitatus	mixturatus	commixticius			
belualis	beluilis	bestialis			
australis	austrinalis	austrinus			
accano	accantito	accanto			

Figura 2. Porzione del dataset monosemico

Quindi, nonostante le etichette assegnate, nessuno dei due sottoinsiemi comprende esclusivamente lemmi polisemici o monosemici: tali dataset avrebbero

richiesto una selezione artificiale di synset, non basata sulla composizione effettiva del LWN. Il dataset polisemico è composto da 28 verbi e 12 nomi, mentre il dataset monosemico da 6 verbi, 27 nomi e 8 aggettivi.

Per quanto riguarda i dati del successivo addestramento, è stato utilizzato l'intero corpus del LWN (estratto a maggio 2024), opportunamente epurato dei dati selezionati per il testing. Questo dataset comprende un totale di 9.345 lemmi distribuiti su 16.529 synset, così ripartiti: 2.726 verbi, distribuiti su 4.601 synset; 983 aggettivi, presenti in 1.955 synset; 5.313 nomi comuni, che popolano 9.463 synset; 233 avverbi, distribuiti su 419 synset; 90 nomi propri, presenti in 91 synset.

```
cognomen: cognomentum, curio, euhans, felix, hospes, nomen
documentum: exemplum
apertura: foramen, macula, punctum
domo: subicio, submitto, uinco
dominor: regno
lignum: materia
anxietas: cogitatio, cura, sollicitudo
modestia: sobrietas, temperantia
affigo: alligo, apto, cohibeo, colligo, concludo, contineo, figo, offirmo, uincio
biblus: charta
depasco: depopulor, euasto, lacero, obtero, perpopulor, populor, uasto
attegia: casa, casula, gurgustium, praesepe, stabulum, taberna
doctus: exercitatus
antecedo: anteeo, antegredior, praecedo, praecurro, praegredior, praeuenio
aberro: absono, discrepo, dissentio, dissono
conubium: matrimonium, thalamus
flamma: splendor
aperio: claro, delico, demonstro, detexo, edissero, euoluo, expedio, explano, explico, expono, interpretor
amplitudo: interuallum, latitudo, laxitas
attollo: insolesco
concubinatus: contubernium, paelicatus
amoenitas: commoditas, dulcitudo, gratia, suauitas
opifex: parens
onus: pondus, sarcina
certus: deliberatus
incendo: peruro
pax: quies, silentium, tranquillitas
fascia: uitta
commentum: conformatio, figura, forma, schema, translatio, tropus
computresco: confracesco, liquesco
signum: uestigium
administro: dispono
asseruo: condo, conseruo, contineo, custodio, seruo, sustento, teneo, tolero, tueor, tutor
belua: echidna, monstrum, prodigium
adaero: aestimo, censeo, exigo, metior, pendo, permetior, puto
assector: comitor, concino, consector, consequor, obloquor, sequor
clibanara: fabrica, officina, pergula, taberna
anhelo: edo, effero, effundo, emitto, erumpo, excutio, fundo, iacio, profundo
arx: turris
```

Figura 3. Porzione del dataset di training

All'interno del più ampio contesto metodologico, è opportuno sottolineare la rilevanza della scelta di utilizzare il LWN stesso come fonte dei dati di addestramento. Come già discusso (§1.2), mentre gran parte della letteratura recente si è concentrata sull'utilizzo di risorse linguistiche strutturate per potenziare le capacità dei LLM, questo studio adotta deliberatamente l'approccio inverso. La decisione di addestrare il modello sui dati del LWN si inserisce in questa prospettiva innovativa, creando un ciclo di *feedback* nel quale il modello, inizialmente addestrato su dati strutturati, viene successivamente impiegato per generare nuovi dati della stessa natura.

Questa metodologia non solo si allinea con l'obiettivo di esplorare come i LLM possano contribuire all'arricchimento di risorse linguistiche strutturate, ma permette

anche di investigare la potenziale interazione bidirezionale tra modelli linguistici e risorse lessicali. La scelta e la conseguente composizione del dataset di training è stata quindi guidata da questa visione metodologica di fondo.

2.3. Baseline – inglese

La creazione di una baseline in inglese rappresenta un passaggio metodologico fondamentale per valutare l'efficacia dell'approccio proposto. La scelta di utilizzare l'inglese come lingua di riferimento è supportata dalla sua posizione predominante nella ricerca sui LLM e dalla completezza delle risorse lessicali disponibili in questa lingua (Bird et al., 2009; Navigli e Ponzetto, 2012) e si allinea con le pratiche metodologiche consolidate nel campo dell'NLP, dove la validazione su una lingua ampiamente studiata fornisce un termine di paragone essenziale per valutare l'efficacia di approcci innovativi su lingue meno rappresentate (Bender, 2011; Joshi et al., 2020).

Per garantire una comparazione diretta con l'esperimento in latino, il dataset inglese è stato costruito seguendo gli stessi criteri strutturali utilizzati per il dataset latino. Specificamente, sono stati creati due insiemi paralleli di synset attraverso un processo di traduzione sistematica: uno contenente termini polisemici e l'altro focalizzato su lemmi monosemici. Questo approccio permette di stabilire parametri di valutazione quantitativi per le performance del modello nel task specifico di generazione lessicale. Le metriche successivamente ottenute dall'esperimento in inglese fungono infatti da riferimento per valutare la qualità e l'accuratezza dell'output generato in latino, fornendo così al di là di un quadro comparativo anche un valore di base sull'analisi della complessità di esecuzione del task stesso da parte del modello.

2.4. Fasi dell'esperimento

La fase sperimentale, come si è detto all'inizio del presente capitolo, è proceduta con un approccio incrementale, esplorando metodologie di complessità crescente per ottimizzare la generazione di lemmi latini. L'esperimento ha seguito una progressione metodologica in tre fasi principali, ciascuna basata su tecniche consolidate nella letteratura dei LLM.

Durante la prima fase si è implementato un approccio *zero-shot*, con l'obiettivo di sfruttare la capacità intrinseca del modello di generalizzare a nuovi task senza

training specifico (Brown et al., 2020). Questa metodologia permette di valutare le *performance* baseline del modello nel generare lemmi latini basandosi esclusivamente sulla conoscenza acquisita durante il *pre-training*. Successivamente, è stata esplorata una strategia *few-shot learning*, che prevede l'introduzione di un numero limitato di esempi dimostrativi nel prompt. Questo approccio, ampiamente documentato in letteratura (i.e. Liu et al., 2022), consente di guidare il modello attraverso esempi espliciti, potenzialmente migliorando la qualità e la pertinenza degli output generati. Infine, l'ultima fase ha visto l'applicazione della tecnica LoRA, un metodo computazionalmente efficiente di *fine-tuning*, che modifica selettivamente solo alcuni parametri del modello (Hu et al., 2021). Questa metodologia consente un adattamento mirato del modello al task specifico, potenzialmente ottimizzando le performance senza incorrere nei costi computazionali e nei rischi di *overfitting*[23] associati al *fine-tuning* tradizionale.

La progressione metodologica adottata permette non solo di valutare l'efficacia relativa di ciascun approccio, ma anche di comprendere come diverse strategie di adattamento possano influenzare la capacità del modello di generare lemmi latini accurati e pertinenti.

2.4.1. Zero-shot training

L'approccio *zero-shot* rappresenta il punto di partenza per valutare le capacità intrinseche del modello di linguaggio nella generazione. Questa tecnica, ampiamente documentata in letteratura (Brown et al., 2020; Perez et al., 2021), sfrutta l'abilità dei LLM di affrontare nuovi task senza addestramento specifico, basandosi esclusivamente sulla conoscenza acquisita durante il *pre-training*. L'efficacia dell'approccio *zero-shot* si basa fondamentalmente sulla natura stessa dei LLM e quindi sul loro processo di *pre-training* su vasti corpora testuali. Durante questa fase, i modelli acquisiscono non solo conoscenze linguistiche e fattuali, ma anche la capacità di inferire pattern e relazioni tra concetti (Bommasani et al., 2021). Questa forma di "comprensione implicita" permette ai modelli di generalizzare a nuovi task senza la necessità di esempi specifici.

[23] Nel caso specifico del presente esperimento, data la limitata quantità di dati si è comunque, in un primo addestramento, incorsi nell'*overfitting* del modello avendo sovrastimato le epoche di training, come verrà discusso più approfonditamente in §2.4.3.

L'approccio *zero-shot* si è dimostrato particolarmente efficace in compiti che richiedono l'applicazione di conoscenze generali o il riconoscimento di pattern linguistici comuni, come la classificazione di testi, il completamento di frasi o, nel nostro caso, l'identificazione di relazioni semantiche tra parole (Wei et al., 2022). Tuttavia, le performance *zero-shot* tendono a variare significativamente in base alla complessità del task e alla sua similarità con i pattern presenti nei dati di pre-training. Task che richiedono conoscenze specialistiche o ragionamenti complessi spesso beneficiano meno di questo approccio (Lu et al., 2022). Nel contesto della generazione lessicale, l'efficacia dello *zero-shot* dipende dalla soprattutto ricchezza della rappresentazione semantica acquisita dal modello durante il *pre-training* e dalla sua capacità di mappare accuratamente le relazioni tra concetti in diverse lingue.

Le performance di Mistral-7B, nonostante le sue dimensioni relativamente contenute rispetto ad altri LLM, si sono dimostrate competitive in diversi *benchmark zero-shot,* (Jiang et al., 2023). In particolare, Mistral-7B ha mostrato prestazioni notevoli in task che richiedono ragionamento linguistico e comprensione semantica, posizionandosi favorevolmente rispetto a modelli di dimensioni simili o superiori (Hendrycks et al., 2021). Un aspetto distintivo di Mistral-7B è la sua architettura SWA (di cui abbiamo discusso in §2.1), che gli permette di gestire efficacemente contesti più lunghi rispetto ad altri modelli della sua classe dimensionale, contribuendo potenzialmente a migliori performance *zero-shot* in task che richiedono la comprensione di contesti estesi (Zhang et al., 2023). Studi comparativi hanno evidenziato come il modello sia particolarmente performante in compiti che richiedono ragionamento logico e comprensione semantica, pur mostrando alcune limitazioni in task altamente specialistici o che necessitano di conoscenze su domini specifici non sufficientemente rappresentate nei dati di *pre-training* (Wei et al., 2022; Kojima et al., 2022). Nel contesto specifico della generazione lessicale e delle relazioni semantiche, Mistral-7B ha dimostrato rilevanti capacità di inferire e produrre sinonimi e relazioni semantiche, benchè con variazioni significative tra diverse lingue e domini semantici (Vulić et al., 2020; Schick et al., 2022).

Al di là delle specificità del singolo modello, l'implementazione dell'approccio *zero-shot* richiede una particolare attenzione alla formulazione del prompt, elemento alquanto importante per ottenere risultati ottimali dalla generazione (Zhang et al., 2023),

soprattutto in un contesto finalizzato a un task specifico. La letteratura recente ha evidenziato come la struttura e la chiarezza delle istruzioni nel prompt possano significativamente influenzare la qualità dell'output generato (Wei et al., 2022; Kojima et al., 2022). L'efficacia del *prompt engineering* si basa su diversi principi chiave; primi fra tutti chiarezza e specificità: istruzioni chiare e dettagliate guidano infatti il modello verso l'output desiderato. Ad esempio, invece di chiedere semplicemente "Genera sinonimi", un prompt più efficace potrebbe essere "Elenca i sinonimi della parola 'X', fornendo solo il lemma di base per ciascun sinonimo". Inoltre, per migliorare l'output e avvicinarlo a quello desiderato potrebbe essere necessario lavorare sulla struttura del prompt stesso e soprattutto sull'ordine in cui le istruzioni vengono formulate (Schick e Schütze, 2021). Per ottenere inoltre un output più adatto al singolo task, risulta importante anche cercare di "sintonizzare" il modello sul vocabolario e sul dominio linguistico specifico. Inoltre, relativamente alle istruzioni, recenti approcci hanno mostrato come includere istruzioni che incoraggiano il modello a "ragionare passo dopo passo" possa migliorare significativamente le performance in task complessi. Nel loro studio, Kojima et al. (2002) hanno introdotto il concetto di "*Zero-shot-CoT"* (Chain of Thought), dove il prompt include un'istruzione esplicita per il modello di ragionare passo dopo passo. Questi principi di *prompt engineering* non operano in isolamento, ma si intersecano e si completano a vicenda. L'efficacia di un prompt infatti dipende spesso dalla combinazione di questi elementi, adattati al task specifico e soprattutto al modello utilizzato. Il *prompt engineering* però è ben lontano dall'essere una scienza esatta: nel caso specifico del presente esperimento, si è infatti dedicata una fase iniziale – potremmo dire una fase zero – proprio al *prompt tuning*, prima di iniziare il vero e proprio esperimento. Tramite un piccolo dataset di testing, (estratto dal dataset di testing più grande) contenente dieci lemmi equamente distribuiti tra monosemici e polisemici, si è cercato di trovare il modo migliore per "istruire" il modello al task, processo che non solo varia da modello a modello ma soprattutto nel contesto del latino deve tenere conto anche della specificità morfologica della lingua e dell'output che si vuole ottenere.

Nella progettazione dei prompt per questo esperimento, si è dunque cercato di incorporare questi principi in modo da massimizzare le performance del modello pur mantenendo l'integrità linguistica e semantica del task. Dopo la prima fase iniziale,

sono stati dunque sviluppati due prompt paralleli: uno per l'inglese e uno per il latino. La scelta di mantenere una struttura simile per entrambi i prompt è stata deliberata, al fine di limitare quanto più possibile scostamenti tra *baseline* e l'output oggetto di valutazione. Questa decisione ha l'obiettivo di isolare più efficacemente l'effetto della lingua (inglese vs. latino) sulle prestazioni del modello, riducendo il "rumore" potenzialmente introdotto da differenze sostanziali nella struttura del prompt. Entrambi i prompt seguono dunque una struttura simile, definendo chiaramente il ruolo del modello come "*powerful AI assistant trained in semantics*" e fornendo istruzioni specifiche e dettagliate. Questa struttura riflette l'approccio di Schick & Schütze (2021), che enfatizza l'importanza di una formulazione chiara e coerente del task.

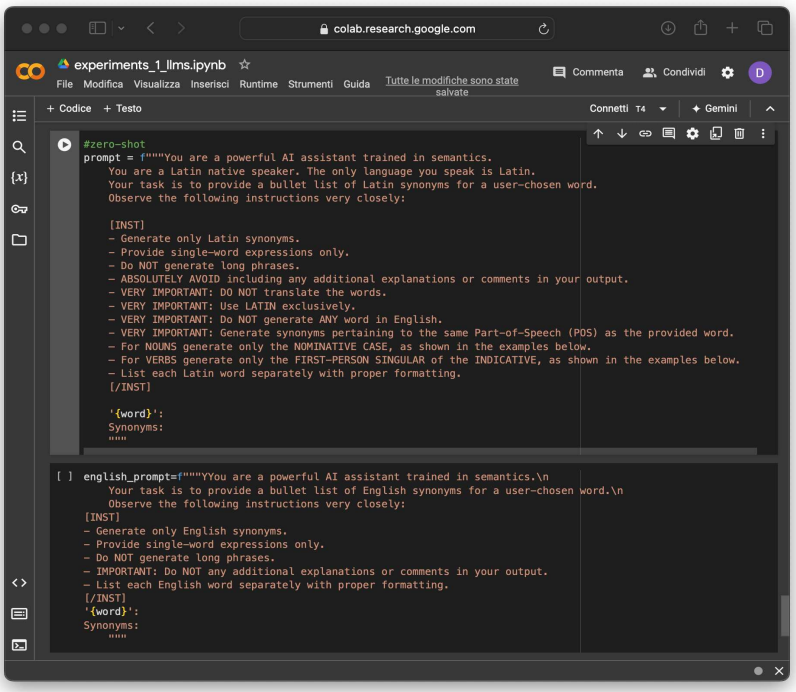


Figura 4. zero-shot prompts

Nel caso del prompt latino, particolare enfasi è stata posta sulla necessità di generare esclusivamente lemmi latini e di rispettare specifiche forme morfologiche (nominativo per i sostantivi, prima persona singolare dell'indicativo per i verbi). Queste istruzioni dettagliate hanno l'obiettivo di guidare implicitamente il modello attraverso un processo di ragionamento strutturato, in linea con le osservazioni di Kojima et al. (2022) sull'efficacia delle istruzioni di ragionamento passo-passo, fungendo da guida per il formato di output desiderato. Si è inoltre deciso di sottolineare le istruzioni più importanti con l'uso di parole chiave come "IMPORTANT" e "VERY IMPORTANT"

per enfatizzare punti critici, non solo facendo riferimento alle *best practices* di *prompt engineering* discusse da Reynolds & McDonell (2021) e Mishra et al. (2022) sull'importanza di istruzioni chiare, specifiche e ben strutturate per massimizzare le performance in contesti *zero-shot*, ma anche in quanto si è notata empiricamente, in una prima fase di *prompt tuning,* la reale efficacia di evidenziare le istruzioni cardine (che allo stesso tempo sembravano risultare le più ostiche per il modello) tramite questo espediente.

Di seguito si riportano alcuni esempi tratti dal primo ciclo di generazione, effettuato in tutto il corso dell'esperimento e dunque con un prompt non ancora ottimizzato, in comparazione con esempi tratti dalle ultime fasi del *prompt-tuning zero shot* prima di aggiungere esempi per *in-context/few-shot learning.*

(1) cogitatio: ['contemplation', 'reflection', 'meditation', 'thoughtfulness', 'deliberateness', 'care']
(2) ambulo: ['walk', 'go on foot', 'move about in an area', 'get around', 'ramble', 'stroll', 'roam', 'stride', 'travel by walking', 'walked across europe', 'hike', 'trek', 'slang', 'go', 'jogging']
(3) abundantia: ['abundancia', 'abundance', 'copiosidad', 'exceso', 'generosidad', 'riqueza']

Questi primi esempi riportati sono, come si diceva, tratti dal primo ciclo di generazione, con un prompt decisamente più semplice in cui si chiedeva al modello – senza una formulazione precisa di istruzioni – di generare sinonimi di parole latine partendo dal lemma fornito in input. Si può vedere come in questa fase iniziale l'output ottenuto era piuttosto una traduzione del lemma, privo di lemmi in latino (neppure flessi).

(4) cogitatio: ['cogitare', 'v', 'think about something carefully and deeply using one's mind rather than', 'consideratio', 'noun f', 'thoughtful consideration based upon careful study']
(5) ambulo: ['ambulor', 'i walk', 'pereo', 'i go', 'perago', 'doing something actively and purposefully', 'progredior', 'progress forward', 'advenio', 'come here from somewhere else', 'transibo', 'go across', 'pass through/over']

(6) abundantia: [‘abundans’, ‘adj’, ‘plenteous’, ‘copious’, ‘richly supplied’, ‘overflowing’, ‘excessive’, ‘amplus’, ‘adv’, ‘more than enough’, ‘too much’, ‘excessively’, ‘exuberare’, ‘v’, ‘to flow out freely and profuse’]

Gli esempi (4)-(6) sono invece riportati dal nono – e ultimo ciclo – di generazione in *zero-shot*. Formulando correttamente le istruzioni, benché con ancora molto lavoro da fare, si è riusciti a produrre in output qualche lemma in latino – non necessariamente un potenziale sinonimo – limitando la presenza di traduzioni o glosse. Le caratteristiche del prompt finale mirano dunque a massimizzare le capacità del modello nel generare in primis lemmi in latino e in secondo luogo potenziali sinonimi pertinenti.

Per quanto riguarda il “doppio” prompt – uno per il latino e uno per l’inglese, sviluppato successivamente a quello latino –, come si diceva in precedenza, questo è ovviamente legato alla necessità di flessibilità per comprendere e rappresentare le sfumature semantiche proprie di ciascuna lingua: i prompt sono simili ma non identici, non essendo necessario per l’inglese ad esempio una precisazione sul sistema dei casi o la necessità di limitare la possibile presenza di traduzioni nell’output. Un aspetto rilevante del prompt latino è infatti l’enfasi posta sulle specificità morfologiche della lingua. Le istruzioni chiariscono esplicitamente la necessità di generare solo il caso nominativo per i sostantivi e la prima persona singolare dell’indicativo per i verbi. Inoltre, il prompt latino include una randomizzazione dei lemmi target in input (tramite la liberira *random*), con l’obiettivo di eliminare la specificità alfabetica, cercando di prevenire potenziali bias nella generazione e a garantire una rappresentazione più completa del lessico latino.

Questa metodologia non solo serve gli obiettivi immediati dell’esperimento, ma contribuisce anche a esplorare i limiti e le potenzialità dei LLM in contesti linguistici diversificati e complessi. In sintesi, i prompt utilizzati in questo esperimento incorporano diversi principi chiave del *prompt engineering* (Reynolds & McDonell, 2021; Mishra et al., 2022; Schick & Schütze, 2021; Kojima et al., 2022) ma allo stesso tempo sono stati sviluppati attraverso un iniziale processo di *trial-and-error* del *prompt-tuning* che – benché dispendioso – ha permesso di andare al di là della teoria e adattare il prompt al contesto specifico del singolo modello da noi usato, con l’obiettivo di inquadrare l’approccio più adeguato alla formulazione delle istruzioni per

massimizzare le performance del modello in un task di generazione di sinonimi in contesto *zero-shot*, tenendo conto delle sfide poste da una lingua come il latino e dal tipo di output aspettato in termini di formattazione e caratteristiche dei lemmi.

2.4.2. Few-shot learning

Dopo l'approccio *zero-shot*, l'esperimento è progredito verso l'implementazione di una strategia *few-shot*. Il *few-shot learning*, come descritto da Brown et al. (2020), permette al modello di apprendere da un numero limitato di esempi forniti nel prompt, migliorando potenzialmente le sue performance su task specifici senza la necessità di *fine-tuning*. Questa tecnica si basa sulla capacità dei LLM di adattarsi rapidamente a nuovi contesti e compiti attraverso l'apprendimento *in-context*. Il *few-shot learning* si posiziona come una fase intermedia tra l'approccio *zero-shot* (nessun esempio) e il *fine-tuning* tradizionale (molti esempi e aggiornamento dei parametri del modello). Questa metodologia sfrutta la flessibilità cognitiva dei LLM, permettendo loro di "comprendere" il task attraverso esempi dimostrativi inclusi nel prompt stesso. Come notato da Liu et al. (2021), l'efficacia del *few-shot learning* dipende fortemente dalla qualità e dalla rappresentatività degli esempi forniti.

Un vantaggio chiave del *few-shot learning* è la sua capacità di migliorare le prestazioni del modello su task specifici senza richiedere grandi dataset di addestramento o costose procedure di *fine-tuning*. Questo lo rende particolarmente adatto per applicazioni in domini specializzati o lingue meno rappresentate, come nel caso del latino in questo studio.

Le ricerche hanno infatti dimostrato l'efficacia del *few-shot learning* in vari contesti. Nel campo della traduzione di lingue a basse risorse, Lauscher et al. (2020) hanno utilizzato questo tipo di approccio per migliorare le prestazioni di traduzione, dimostrando miglioramenti significativi rispetto alle tecniche *zero-shot*. Per quanto riguarda la classificazione di testi specializzati, Schick & Schütze (2021) hanno applicato metodi di *few-shot learning* in domini altamente specializzati, come la letteratura medica, ottenendo risultati comparabili a quelli di modelli addestrati su grandi dataset.

L'analisi del *sentiment* in contesti culturali specifici ha anche beneficiato di questo approccio, come dimostrato da Winata et al. (2021), che hanno utilizzato il

few-shot learning per adattare modelli di analisi del sentimento a contesti culturali e linguistici specifici, evidenziando la flessibilità di questa metodologia. Nel campo della generazione di codice, Chen et al. (2021) hanno mostrato l'efficacia del *few-shot learning* nel guidare i modelli linguistici nella generazione di codice in linguaggi di programmazione specifici, anche con un numero limitato di esempi. Infine, nel riconoscimento di entità nominate in domini specializzati, Ding et al. (2021) hanno applicato tecniche di *few-shot learning* per migliorare le prestazioni in testi legali e medici, ambiti in cui la terminologia specifica può rappresentare una sfida significativa per i modelli generici.

I vantaggi del *few-shot learning* si estendono oltre la semplice efficienza computazionale. Come osservato da Gao et al. (2021), questo approccio permette una maggiore flessibilità e adattabilità dei modelli, consentendo loro di affrontare rapidamente nuovi task o domini senza la necessità di riaddestramento. Inoltre, può aiutare a mitigare il problema del "*catastrophic forgetting*"[24] osservato in alcuni approcci di *fine-tuning* tradizionali (Ramasesh et al., 2021).

L'inclusione di esempi specifici nel prompt potrebbe quindi guidare il modello verso una comprensione più accurata del task e del contesto linguistico, potenzialmente migliorando la qualità e la pertinenza dei sinonimi generati e mantenendo al contempo la sua capacità di generalizzazione. Questo approccio si allinea con le osservazioni di Perez et al. (2021), che hanno evidenziato come il *few-shot learning* possa essere particolarmente efficace in domini specializzati o per lingue con risorse limitate.

Dunque, nel contesto delle lingue antiche o poco rappresentate, come il latino, questo tipo di approccio offre l'opportunità di sfruttare la conoscenza linguistica generale dei LLM, adattandola alle specificità della lingua target. Tuttavia, è anche necessario ricordare che l'efficacia del *few-shot* può variare a seconda della complessità del task e della qualità degli esempi forniti. La scelta di implementare un approccio *few-shot* per questo esperimento è stata dunque motivata da diversi fattori, come la complessità intrinseca del task, la rappresentazione limitata nei dati di *training,* l'obiettivo di standardizzazione dell'output e la ricerca di un bilanciamento tra

[24] Il *catastrophic forgetting* (o *catastrophic interference*) è un fenomeno per cui le reti neurali tendono a dimenticare drasticamente le informazioni precedentemente apprese quando vengono addestrate su nuovi dati (McCloskey & Cohen, 1989; Roger, 1990). Questo problema è particolarmente rilevante nei LLM dove l'apprendimento continuo di nuove informazioni può portare alla perdita delle capacità precedentemente acquisite (Kirkpatrick et al., 2017).

generalizzazione e specificità. Sulla base di queste considerazioni, è stato sviluppato un set di prompt specificatamente progettati per sfruttare i vantaggi del *few-shot learning* nel contesto della generazione di sinonimi in latino e in inglese. Entrambi i prompt mantengono la struttura di base utilizzata nell'approccio *zero-shot*, ma integrano una serie di esempi che fungono da guida per il modello, correttamente formattati e equamente distribuiti tra parole polisemiche e parole monosemiche.

Figura 5. Parziale lista degli esempi forniti nel few-shot per il latino

Il prompt per l'inglese include 7 esempi di parole con i relativi sinonimi, coprendo una varietà di parti del discorso (verbi, aggettivi, sostantivi). Questi esempi sono concisi e diretti, fornendo un modello chiaro per il tipo di output atteso. Le istruzioni rimangono focalizzate sulla generazione di sinonimi in inglese, enfatizzando la necessità di espressioni a singola parola e evitando spiegazioni aggiuntive. Il prompt per il latino, d'altra parte, è notevolmente più dettagliato e specifico. Include 15 esempi, significativamente più degli esempi in inglese, riflettendo la maggiore complessità e specificità richiesta per il task in latino.

La differenza nella quantità e nella specificità degli esempi tra i prompt inglese e latino riflette la consapevolezza delle sfide uniche poste dal latino, una lingua meno rappresentata nei dati di training dei modelli linguistici moderni. Questa strategia si allinea dunque con le già citate osservazioni di Liu et al. (2021) sull'importanza della selezione degli esempi nel *few-shot learning*, dove la diversità e la rappresentatività degli esempi forniti possono influenzare significativamente la capacità del modello di

generalizzare a nuovi input. Inoltre, il prompt latino include una nota specifica volta a precisare che, anche se gli esempi mostrano parole che iniziano con lettere simili, i sinonimi latini devono usare lettere iniziali diverse per limitare potenziali bias nella generazione.

Come si diceva in precedenza, anche in questa fase si è lavorato in un primo momento a livello di prompt tuning con vari cicli di generazione che hanno permesso, attraverso una serie di piccole modifiche nella formattazione degli esempi, di giungere al prompt finale. Di seguito si riportano tre esempi tratti dall'ultima generazione della fase di prompt tuning in *few-shot* e si rimanda a 2.4.1. per un confronto con le fasi di elaborazione del prompt in *zero-shot*:

(7) cogitamen: ['consilium', 'deliberatumen']
(8) ambulo: ['ambulator', 'perrector', 'rectificator']
(9) abundantia: ['abundans', 'adscendere', 'aestivare', 'agrarii', 'apellicebatur', 'aquilae', 'aspicientur', 'assidue']

Come si può vedere, l'output si è per così dire "latinizzato" rispetto agli esempi precedenti in 2.4.1, benché nella generazione siano presenti alcune allucinazioni (pseudo-parole, ovvero lemmi che sembrano latini ma che in realtà non esistono), come nel caso di *deliberatumen* in 7, e molte forme flesse o non allineate per POS. Nonostante tutto si è ritenuto questo un buon punto di partenza per procedere con l'esperimento vero e proprio, cristallizzando il prompt, che successivamente non è più stato modificato.

In conclusione, dunque, i prompt elaborati per questo esperimento *few-shot* rappresentano un'applicazione mirata dei principi dell'apprendimento *in-context* (Brown et al., 2020), calibrata sulle sfide specifiche della generazione di sinonimi in latino e in inglese. Anche in questo caso è stato infatti necessario differenziare i due prompt, con un approccio più dettagliato e strutturato per il latino, caratterizzato da più esempi di varia natura per adattarsi ad affrontare le peculiarità di lingue morfologicamente complesse e meno rappresentate nei dati di *training*. Tuttavia, è importante notare che, nonostante questi sforzi, l'output presenta ancora significative imperfezioni, incluse allucinazioni e forme non allineate. Questo suggerisce che, mentre l'approccio *few-shot* offre un punto di partenza promettente, rimane ancora molto lavoro

da fare per ottimizzare l'uso dei LLM in compiti linguistici specializzati, specialmente per lingue antiche o poco rappresentate. Questi risultati preliminari evidenziano sia il potenziale che le limitazioni attuali dell'applicazione dei LLM al task in esame. È importante sottolineare che i risultati effettivi del *few-shot training* saranno discussi nel capitolo tre. Questa fase iniziale di elaborazione e raffinamento del prompt ha però fornito una base su cui costruire l'esperimento principale, offrendo spunti preziosi sulle sfide e le opportunità che si presentano nell'applicazione dei LLM a compiti linguistici specializzati.

2.4.3. Fine-tuning

Dopo aver esplorato gli approcci *zero-shot* e *few-shot*, l'ultima fase dell'esperimento ha coinvolto l'applicazione del *fine-tuning* utilizzando la tecnica LoRA.

LoRA, introdotto da Hu et al. (2021), è un metodo efficiente di fine-tuning che modifica selettivamente solo alcuni parametri del modello, offrendo un equilibrio tra la capacità di adattamento ed l'efficienza computazionale.

Il *fine-tuning* tradizionale comporta l'aggiornamento di tutti i parametri di un modello pre-addestrato su un nuovo dataset specifico per il task. Tuttavia, questo approccio può essere computazionalmente costoso e rischia di causare il *catastrophic forgetting*, dove il modello perde le capacità generali acquisite durante il *pre-training* (vd. nota 23). LoRA affronta queste sfide introducendo matrici di rango basso che vengono addestrate parallelamente ai pesi originali del modello, permettendo un adattamento mirato senza modificare la maggior parte dei parametri originali (Ren et al., 2024). LoRA si basa sul principio che molte modifiche necessarie per adattare un modello a un nuovo task possono essere individuate attraverso aggiornamenti di rango basso alle matrici di peso originali. In pratica, LoRA decompone l'aggiornamento della matrice di peso in due matrici più piccole: una matrice di *downprojection* e una di *upprojection*.[25] Queste matrici sono inizializzate casualmente e addestrate durante il

[25] La matrice di *downprojection* è responsabile della riduzione della dimensionalità dell'input. Questa matrice proietta l'input in uno spazio di dimensioni inferiori, identificando le caratteristiche più rilevanti per il task specifico. Questo concetto si basa sul principio di compressione dell'informazione discusso da Aghajanyan et al. (2021) nel loro lavoro sulla *"intrinsic dimensionality*" dei modelli linguistici. La matrice di *upprojection* riporta la rappresentazione di bassa dimensionalità nello spazio originale. Questa matrice è responsabile della trasformazione delle caratteristiche di basso rango in un aggiornamento che può essere applicato alla matrice di peso originale. L'efficacia di questa proiezione è stata ulteriormente esplorata da Zhang et al. (2022) nel loro studio sulla compressione dei modelli attraverso proiezioni di rango basso.

fine-tuning, mentre i pesi[26] originali del modello rimangono congelati. Il congelamento dei pesi originali del modello durante LoRA è una caratteristica chiave che distingue questo approccio dal *fine-tuning* tradizionale. Questa strategia, nota come "*parameter-efficient fine-tuning*" (PEFT), è stata esplorata in vari studi. Per esempio, Houlsby et al. (2019) hanno introdotto il concetto di "*adapter layers*" che, similmente a LoRA, aggiungono parametri addestrabili mantenendo congelati i pesi originali. Pfeiffer et al. (2020) hanno ulteriormente sviluppato questo concetto con i loro "*AdapterFusion*", dimostrando come questo approccio possa preservare le conoscenze pre-addestrate mentre si adatta a nuovi task. Il congelamento dei pesi originali non solo riduce il rischio di *catastrophic forgetting* (Houlsby et al., 2019; Ren et al., 2024), ma permette anche una maggiore efficienza computazionale e di memoria, come evidenziato da Xu et al. (2021) nel loro lavoro sulla compressione dei modelli linguistici.

La formula matematica alla base di LoRA può essere espressa come: W' = W + BA; dove W è la matrice di peso originale, B è la matrice di *downprojection*, A è la matrice di *upprojection*, e W' è la matrice di peso effettiva utilizzata durante l'inferenza. Il prodotto BA è una matrice di rango basso che individua l'adattamento specifico al task. La scelta di utilizzare LoRA per questo esperimento è motivata da diverse considerazioni. Innanzitutto, l'efficienza computazionale di LoRA permette di adattare il modello con risorse limitate, un vantaggio significativo quando si lavora con LLM di grandi dimensioni. Inoltre, mantenendo intatta la maggior parte dei parametri originali, LoRA aiuta a preservare le capacità generali del modello acquisite durante il pre-training. Questo approccio offre anche un'adattabilità mirata al task in esame, potenzialmente migliorando le performance su questo compito specifico. Infine, la flessibilità di LoRA permette di sperimentare con diversi gradi di adattamento, modificando il rango delle matrici utilizzate.

[26] I pesi di un modello di *deep learning* si riferiscono ai parametri numerici che determinano come il modello elabora gli input per produrre gli output. In un modello neurale, questi pesi rappresentano la "forza" delle connessioni tra i neuroni artificiali (Goodfellow et al., 2016). Durante l'addestramento, questi pesi vengono iterativamente aggiustati per minimizzare la discrepanza tra le previsioni del modello e i risultati desiderati, un processo noto come *backpropagation* (Rumelhart et al., 1986). Nel contesto dei LLM, come discusso da Vaswani et al. (2017), i pesi sono distribuiti attraverso varie componenti del modello, inclusi i layer di attenzione e *feed-forward*. La capacità di un modello di apprendere e generalizzare dipende criticamente dalla configurazione di questi pesi (Zhang et al., 2021).

Il processo di fine-tuning utilizzando LoRA è stato implementato sfruttando l'ambiente di Google Colab, che ha fornito accesso a una GPU NVIDIA A100, permettendo di utilizzare risorse computazionali avanzate senza la necessità di hardware locale dedicato (Bisong, 2019). L'A100, con la sua architettura Ampere e memoria ad alta larghezza di banda, ha consentito di gestire efficacemente i carichi di lavoro computazionalmente intensivi associati al *fine-tuning* di LLM, nonostante le limitazioni tipiche degli ambienti cloud condivisi (NVIDIA, 2020; Carneiro et al., 2018). Seguendo l'approccio delineato da Hu et al. (2021) la configurazione LoRA è stata impostata con una dimensione della matrice di basso rango (r) di 8 e un fattore di scala (lora_alpha) di 32. Questi parametri controllano il grado di adattamento del modello, bilanciando la flessibilità dell'apprendimento con la preservazione delle conoscenze pre-addestrate. I moduli target per l'adattamento sono stati identificati nelle proiezioni delle query e dei valori ("q_proj" e "v_proj") all'interno dell'architettura del modello, focalizzando l'adattamento su componenti chiave del meccanismo di attenzione. È stato applicato un tasso di *dropout* del 10% come misura di regolarizzazione, seguendo le pratiche standard descritte da Srivastava et al. (2014) per prevenire l'*overfitting*[27].

Il dataset per il *fine-tuning* è stato pre-elaborato utilizzando tecniche standard di tokenizzazione e codifica, in linea con le metodologie descritte da Devlin et al. (2019) per i modelli Transformer. La preparazione dei dati ha incluso la suddivisione in set di addestramento, validazione e test, con una lunghezza massima di sequenza fissata a 512 token. È stata inoltre implementata una strategia di valutazione e salvataggio del

[27] L'*overfitting,* caratterizzato da un miglioramento continuo delle prestazioni sul set di addestramento accompagnato da un deterioramento delle prestazioni sul set di validazione, è un problema ben noto nell'apprendimento automatico (Goodfellow et al., 2016). Questo fenomeno si verifica quando un modello apprende troppo specificamente dai dati di addestramento, inclusi rumori e fluttuazioni casuali, perdendo la capacità di generalizzare a nuovi dati. In pratica, il modello "memorizza" il set di addestramento invece di apprendere pattern generali, risultando in prestazioni scadenti su dati non visti. L'*overfitting* può manifestarsi in vari modi, come una complessità eccessiva del modello rispetto alla quantità di dati disponibili, o un addestramento prolungato oltre il punto ottimale. Le strategie per mitigare l'*overfitting* includono tecniche di regolarizzazione, come la L1/L2 regularization, il *dropout*, l'*early stopping*, e l'aumento dei dati (*data augmentation*). La L1 (Lasso) e L2 (Ridge) *regularization* sono metodi che aggiungono un termine di penalità alla funzione di perdita, scoraggiando la complessità del modello: L1 promuove la sparsità dei pesi, mentre L2 li mantiene piccoli ma non necessariamente zero (Tibshirani, 1996; Hoerl & Kennard, 1970). Per approfondimenti su queste tecniche, si può consultare il lavoro di Ng (2004) sulla feature selection e regolarizzazione. Il *dropout* consiste nel disattivare casualmente una percentuale di neuroni durante l'addestramento, forzando il modello a apprendere rappresentazioni più robuste e meno dipendenti da specifici neuroni (Srivastava et al., 2014). L'*early stopping*, invece, interrompe l'addestramento quando le prestazioni sul set di validazione iniziano a peggiorare, evitando così un'eccessiva specializzazione sui dati di training (Prechelt, 1998). Il monitoraggio attento delle performance su un set di validazione separato durante l'addestramento è dunque importante per rilevare e prevenire l'*overfitting* (Hastie et al., 2009).

modello alla fine di ogni epoca, mantenendo solo i due migliori checkpoint basati sulla *loss*[28] di validazione. L'utilizzo del *mixed precision training*, come descritto da Micikevicius et al. (2018), ha permesso di ottimizzare l'efficienza computazionale. Per monitorare le prestazioni del modello, sono state calcolate metriche standard come accuratezza, precisione, *recall* e *F1-score*, seguendo le definizioni fornite da Goutte & Gaussier (2005). Inoltre, è stato implementato un meccanismo di *early stopping* con una pazienza di un'epoca per prevenire l'*overfitting*, in accordo con le raccomandazioni di Prechelt (1998). Proprio relativamente a queste metriche valutative, il processo di fine-tuning è stato oggetto di un'attenta calibrazione per ottimizzare le prestazioni del modello. Inizialmente, l'addestramento era stato configurato per dieci epoche, seguendo le pratiche comuni nel fine-tuning di modelli linguistici (Howard & Ruder, 2018). Tuttavia, durante il monitoraggio delle performance, si è osservato un fenomeno di *overfitting* alla quinta epoca.

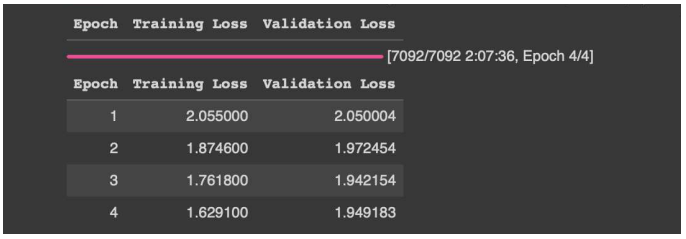
Epoch	Training Loss	Validation Loss
1	2.059900	2.055673
2	1.882600	1.972364
3	1.767700	1.939063
4	1.628100	1.930776
5	1.500100	1.944558
6	1.438100	1.971497

Figura 6. Metriche durante il primo tentativo di fine-tuning

In risposta a questa osservazione, il processo di addestramento è stato interrotto per poi essere ricalibrato. Il numero di epoche è stato ridotto da dieci a quattro, un valore determinato empiricamente sui risultati del processo precedente, che ha permesso di raggiungere un equilibrio ottimale tra l'apprendimento del task specifico e la prevenzione dell'*overfitting*. La necessità di questa regolazione sottolinea l'importanza di un approccio iterativo e attento al *fine-tuning*, riflettendo la natura delicata dell'adattamento di modelli linguistici pre-addestrati a compiti specifici con una limitata quantità di dati, dove il bilanciamento tra l'acquisizione di conoscenze specifiche del dominio e il mantenimento delle capacità generali del modello ha una sua rilevanza

[28] La *loss* (o funzione di perdita) è una misura fondamentale nell'addestramento di modelli di *machine learning,* che quantifica la discrepanza tra le previsioni del modello e i valori reali. Nel contesto dei modelli di linguaggio, la loss di validazione si riferisce al valore di questa funzione calcolato su un set di dati separato, non utilizzato durante l'addestramento. Una *loss* di validazione più bassa indica generalmente una migliore generalizzazione del modello. Per approfondimenti, si veda: Goodfellow et al. 2016, sezione. 5.5.

(Peters et al., 2019). Si sono dunque aggiornati i parametri di addestramento, fissando le epoche su 4 e facendo ripartire l'addestramento.



Epoch	Training Loss	Validation Loss

[7092/7092 2:07:36, Epoch 4/4]

Epoch	Training Loss	Validation Loss
1	2.055000	2.050004
2	1.874600	1.972454
3	1.761800	1.942154
4	1.629100	1.949183

Figura 7. Metriche dell'addestramento finale

Il trend della *loss* di *training* mostra un costante miglioramento attraverso le quattro epoche di addestramento, con valori che decrescono da 2.055 nella prima epoca a 1.629 nell'ultima. Questa diminuzione progressiva indica che il modello sta effettivamente apprendendo dai dati di *training*, raffinando la sua capacità di generare sinonimi latini pertinenti. La *loss* di validazione segue un pattern più complesso, iniziando a 2.050 e raggiungendo 1.949 nell'ultima epoca, con un leggero aumento rispetto all'epoca precedente. Questo comportamento è in linea con quanto osservato da Prechelt (1998) riguardo alle dinamiche di apprendimento e al rischio di *overfitting*. La divergenza finale tra la *loss* di *training* e quella di validazione suggerisce che il modello ha raggiunto un punto di equilibrio ottimale, come descritto da Goodfellow et al. (2016) nella loro discussione sull'ottimizzazione e la generalizzazione dei modelli di *deep learning*.

Questi risultati indicano che il modello ha acquisito una comprensione più profonda della semantica latina e delle relazioni sinonimiche, aspetto piuttosto rilevante nel trattamento computazionale delle lingue antiche, come sottolineato da McGillivray & Vatri (2018) nel loro lavoro sull'applicazione di tecniche di NLP al latino. La riduzione della *loss* suggerisce un miglioramento nella capacità del modello di predire e generare sinonimi appropriati nel contesto del latino classico. Tuttavia, come evidenziato da Bender & Koller (2020), è importante ricordare che la *loss*, pur essendo un indicatore utile dell'apprendimento del modello, non si traduce direttamente in una misura della qualità o dell'accuratezza semantica dei sinonimi generati. Per una valutazione completa dell'efficacia del *fine-tuning*, sarà ovviamente necessario condurre un'analisi qualitativa dei sinonimi prodotti. Il leggero aumento della *loss* di validazione nell'ultima epoca suggerisce che future iterazioni dell'esperimento potrebbero

beneficiare di tecniche di regolarizzazione più aggressive o di una strategia di *early stopping* più sensibile, come discusso da Zhang et al. (2021) nel loro lavoro sulla generalizzazione dei modelli di *deep learning*.

In conclusione, questi risultati dimostrano il potenziale dell'approccio LoRA nell'adattare efficacemente un modello linguistico generico al dominio specializzato del latino classico, aprendo nuove possibilità per l'applicazione di tecniche di NLP avanzate nello studio e nell'elaborazione di lingue antiche, un campo di ricerca in rapida evoluzione come evidenziato da Piotrowski (2012) nel suo lavoro pionieristico sull'elaborazione del linguaggio naturale per le lingue storiche.

2.4.4. Parametri della generazione

In quest'ultima sezione si esaminerà brevemente la funzione di generazione dei sinonimi, elemento conclusivo della metodologia applicata.

La funzione mantiene costanti il prompt e i parametri del modello tra le diverse strategie di apprendimento, facilitando un confronto diretto dei risultati. Il prompt, arricchito con gli esempi del *few-shot learning*, fornisce il contesto necessario per la generazione. Come ampiamente discusso nelle sezioni 2.4.1 e 2.4.3, il modello viene istruito attraverso questo prompt dettagliato che specifica le regole per la generazione dei sinonimi.

I parametri[29] del modello sono stati attentamente ottimizzati per bilanciare creatività e precisione nell'output. La temperatura è stata impostata a 0.5, un valore che cerca di equilibrare la coerenza delle risposte con un certo grado di creatività. Per affinare ulteriormente la selezione dei token, si è optato per un approccio combinato di

[29] I parametri di generazione del modello sono fondamentali per controllare la qualità e la natura dell'output. La temperatura (0.0-1.0) regola la casualità delle previsioni: valori più bassi producono output più deterministici, mentre valori più alti aumentano la creatività (Holtzman et al., 2019). Il *top-k sampling* limita la selezione ai k token più probabili, riducendo output incoerenti (Fan et al., 2018). Il *top-p sampling* (o nucleus sampling) seleziona i token la cui probabilità cumulativa non supera p, bilanciando diversità e qualità (Holtzman et al., 2020). La *repetition penalty* scoraggia la ripetizione di token, promuovendo varietà lessicale (Keskar et al., 2019). Il parametro max_new_tokens limita la lunghezza dell'output, mentre do_sample=True attiva la generazione stocastica anziché *greedy*. A tal proposito, la generazione *greedy* seleziona sempre il token più probabile, producendo output deterministici ma potenzialmente meno creativi. La generazione stocastica (do_sample=True) campiona invece dalla distribuzione di probabilità dei token, introducendo casualità controllata e permettendo maggiore diversità. Questo approccio, combinato con parametri come temperatura, top-k e top-p, bilancia coerenza e varietà nell'output. La generazione stocastica permette di esplorare diverse possibilità mantenendo la precisione semantica. Per approfondimenti sui metodi di decodifica e il loro impatto sulla qualità del testo generato, si veda Holtzman et al. (2020).

top-k e top-p *sampling*. Il top-k è stato impostato a 40, valore scelto per offrire un buon compromesso tra diversità e qualità dell'output: è infatti sufficientemente alto da permettere una varietà di scelte lessicali, cercando di mantenere al contempo una buona coerenza semantica, punto particolarmente importante in un task come quello in esame, dove si desidera una certa creatività nell'esplorazione del vocabolario, senza deviare troppo dal campo semantico appropriato. D'altra parte, il top-p a 3 permette una certa flessibilità nella selezione applicando il *nucleus sampling* che considera i token la cui probabilità cumulativa raggiunge questa soglia: con un top-p di 0.3, infatti il modello considera solo i token più probabili che insieme coprono il 30% della distribuzione di probabilità, permettendo una certa varietà nelle scelte lessicali senza allontanarsi troppo dal contesto semantico desiderato, sempre con l'obiettivo di mantenere una stretta relazione semantica pur permettendo l'esplorazione di varianti lessicali meno ovvie ma potenzialmente rilevanti.

Per promuovere la diversità lessicale ed evitare ripetizioni, è inoltre stata applicata una *repetition penalty* di 1.4, senza però penalizzare eccessivamente l'uso di termini simili che potrebbero essere rilevanti nel contesto dei sinonimi. Inoltre, per mantenere le risposte concise, si è imposto un limite di 150 nuovi token (max_new_tokens), soglia ritenuta un buon compromesso tra la necessità di fornire una lista sufficientemente ampia di sinonimi e l'esigenza di evitare divagazioni o generazioni eccessivamente lunghe. L'attivazione dell'opzione do_sample garantisce una generazione stocastica, permettendo al modello di esplorare diverse possibilità anziché selezionare sempre il token più probabile e dunque favorendo la produzione di una varietà di potenziali sinonimi, alcuni dei quali potrebbero essere meno ovvi ma ugualmente validi nel contesto della lingua latina.

Questa calibrazione dei parametri mira dunque a produrre potenziali sinonimi latini che siano al contempo pertinenti, vari e precisi, mantenendo un delicato equilibrio tra fedeltà al contesto semantico e ricchezza del vocabolario latino. L'interazione tra questi diversi parametri ha l'obiettivo di consentire al modello di generare output più vari, pur rimanendo all'interno dei confini del task definiti dal prompt iniziale.

2.4.5. Metodologie di validazione dei risultati

I dati in input sono stati validati principalmente utilizzando il LWN come riferimento principale, considerando i synset dei lemmi target e dei lemmi generati. Inoltre, sono stati consultati altri dizionari e risorse lessicali di riferimento, come il Lewis & Short Latin Dictionary (LS), l'Oxford Latin Dictionary (OLD) e il Thesaurus Linguae Latinae (TLL) per un'ulteriore validazione, soprattutto nel caso di potenziali sinonimi attualmente non annotati in synset target.

Per un'ulteriore verifica dei risultati relativamente ai significati dei sono state consultate le ontologie verbali presenti sulla piattaforma Unified Verb Index (UVI)[30]. L'UVI dà accesso a diverse risorse, VerbNet, PropBank, FrameNet, OntoNotes, SynSemClass Lexicon, che contengono tassonomie dettagliate che organizzano i verbi in base ai tipi di eventi che denotano. Confrontando i synset,i significati dei dizionari e le categorie di eventi riportate dalle risorse dell'UVI, sia per i verbi target e sia per i verbi sinonimi generati, è stato possibile confermare che i sinonimi individuati denotassero effettivamente lo stesso tipo di evento. Questo approccio di validazione incrociata con la classificazione semantica dell'UVI si è rivelato particolarmente utile per migliorare l'accuratezza dell'analisi qualitativa, soprattutto per i verbi, che spesso presentano una semantica più complessa rispetto ad altre parti del discorso.

Bisogna notare che per i lemmi monosemici, tutti i risultati tendono ad appartenere allo stesso synset, essendo l'input monosemico. Viceversa, per le parole polisemiche, i lemmi generati potrebbero appartenere a synset diversi, proprio perché l'input stesso è polisemico e appartiene a più synset. In questi casi, è dunque necessario un controllo e una validazione manuale dei risultati da parte di annotatori umani (*human-in-the-loop*) per raggruppare correttamente i sinonimi generati nei synset appropriati e ottenere risultati più accurati e affidabili, soprattutto per quanto riguarda i lemmi polisemici.

[30] Si rimanda a nota 5 per approfondimenti.

3. Risultati e discussione

Questo capitolo presenta una valutazione complessiva dell' esperimento sulla generazione di sinonimi in latino attraverso i diversi approcci discussi nel capitolo 2. L'analisi si articola su due livelli, combinando metriche quantitative con osservazioni qualitative per fornire una visione completa delle prestazioni del modello (legate anche alle scelte metodologiche adoperate) e delle sfumature dei potenziali sinonimi generati.

L'analisi quantitativa esamina dettagliatamente la performance dei quattro distinti approcci – la *baseline* in inglese, che serve come punto di riferimento comparativo; *zero-shot*, dove il modello opera senza esempi preliminari; *few-shot* che utilizza alcuni esempi dimostrativi nel prompt e infine il modello *fine-tuned* sul LWN – attraverso metriche di *precision*, *recall* e F1 *score*[31].

Vale la pena sottolineare a questo punto che, tuttavia, le metriche presentate hanno delle limitazioni intrinseche dovute alla natura stessa del *ground truth* utilizzato. Il LWNt, infatti, è un progetto ancora in fase di sviluppo e i suoi synset non è detto che contengano tutti i sinonimi esistenti. Inoltre, tra i falsi negativi prodotti da questo esperimento – che impattano particolarmente sul *recall* – sono presenti lemmi che in questa fase consideriamo solo come potenziali sinonimi, in attesa di una validazione più approfondita. Questa considerazione è da tenere a mente nell'interpretazione dei risultati che seguono.

Il processo di annotazione e validazione dei risultati del modello ha coinvolto due annotatori che hanno lavorato alla valutazione della presenza nell'output di potenziali sinonimi, ovvero lemmi semanticamente simili che potrebbero quindi essere

[31] Sono metriche standard nel campo dell'elaborazione del linguaggio naturale. La precisione (*precision*) è definita come il rapporto tra i sinonimi correttamente identificati (veri positivi) e il totale dei sinonimi generati dal modello (veri positivi + falsi positivi); questa metrica permette di valutare l'affidabilità del sistema nel generare sinonimi pertinenti e corretti. Il richiamo (*recall*), d'altra parte, misura la completezza della generazione, calcolando il rapporto tra i sinonimi correttamente identificati (veri positivi) e il numero totale di sinonimi effettivamente esistenti nel *gold standard* (veri positivi + falsi negativi); questa metrica aiuta a quantificare l'efficacia del modello nel recuperare tutti i possibili sinonimi rilevanti per un dato lemma. Il punteggio F1, essendo la media armonica tra precisione e richiamo *(2 * (precision * recall)/(precision + recall))*, fornisce una misura sintetica che bilancia questi due aspetti complementari della performance; in questo caso la scelta della media armonica, invece di una media aritmetica, penalizza gli squilibri estremi tra precisione e richiamo, favorendo i sistemi che mantengono un buon equilibrio tra le due metriche.

considerati per l'inclusione nello stesso synset del LWN, per la creazione di un *gold-standard* (utilizzato poi come *ground truth*). Questo processo di validazione ha seguito diversi criteri, come si è anticipato in 2.4.5., tra cui la verifica della similarità semantica tra i lemmi proposti, la valutazione dell'appropriatezza per l'inclusione nei synset di LWN, il controllo della coerenza contestuale e dell'uso storico, nonché l'analisi delle sfumature di significato specifiche della lingua latina tramite il supporto di dizionari come l'OLD, LS e il TLL.

Ad arricchire l'analisi statistica, segue una valutazione qualitativa che ha l'obiettivo di addentrarsi in considerazioni linguistiche riguardanti i sinonimi generati e più in generale le generazioni effettuate dal modello in ogni singola fase dello sviluppo.

Questa duplice prospettiva di analisi, unita al processo di validazione, permette di valutare non solo l'efficacia del modello, ma anche la qualità effettiva dell'output, non necessariamente rappresentata adeguatamente da un'analisi solo quantitativa, e di conseguenza potenziale utilità nell'arricchimento di risorse lessicali strutturate come il LWN.

3.1. Analisi quantitativa

L'esperimento ha incluso tre approcci distinti e la generazione di una *baseline* in inglese: *zero-shot*, *few-shot* e *fine-tuning* per il latino. Utilizzando la *baseline* inglese come punto di riferimento, possiamo valutare in maniera obiettiva le prestazioni di ogni approccio successivo nel contesto della generazione di sinonimi latini, in relazione al task di generazione di sinonimi per una lingua altamente rappresentata.

Come detto nella sezione 2.2., il dataset di testing di 80 lemmi è stato diviso in due dataset minori – "polisemico" e "monosemico" – sulla base dei synset che li compongono.

	Totale			Dataset polisemico			Dataset monosemico		
	F1	R	P	F1	R	P	F1	R	P
EB	.169	.120	.287	.196	.133	.372	.138	.103	.208
ZS	.078	.066	.094	.069	.049	.115	.096	.139	.074

	Totale			Dataset polisemico			Dataset monosemico		
	F1	R	P	F1	R	P	F1	R	P
FS	.175	.148	.215	.159	.116	.254	.212	.280	.170
FT	.336	.256	.487	.373	.258	.670	.221	.247	.200

Tabella 2. Metriche della Performance (F1: F1-score, R: Recall, P: Precision | EB: English Baseline, ZS: ZeroShot, FS: Few-Shot, FT: Fine-Tuning)

Quindi, come possiamo vedere in tabella 2, la *baseline* inglese ha raggiunto un punteggio F1 complessivo di 0,169, stabilendo il nostro benchmark iniziale di prestazioni. Bisogna notare che ha mostrato prestazioni migliori sui termini polisemici, raggiungendo un punteggio F1 di 0,196 (precisione: 0,372, richiamo: 0,133), mentre per i termini monosemici, il punteggio F1 era 0,138 (precisione: 0,208, richiamo: 0,103). Questa baseline dimostra dunque le sfide intrinseche nel task in esame, indipendentemente dalla lingua oggetto del task.

Passando al latino, l'approccio zero-shot ha mostrato un calo significativo delle prestazioni rispetto alla *baseline* inglese. Ha raggiunto un punteggio F1 complessivo di 0,078, con una precisione di 0,094 e un richiamo di 0,066. Su 500 previsioni generate, solo 47 erano corrette rispetto ai 710 sinonimi del *ground truth*. Per i termini polisemici, il punteggio F1 era 0,069 (precisione: 0,115, richiamo: 0,049), mentre per i termini monosemici era 0,096 (precisione: 0,074, richiamo: 0,139). Questo approccio ha avuto particolare difficoltà con i termini polisemici (punteggio F1: 0,069) rispetto a quelli monosemici (punteggio F1: 0,096). Il sostanziale calo delle prestazioni evidenzia la difficoltà di trasferire la conoscenza linguistica generale a un compito specializzato in una lingua antica senza alcun adattamento specifico.

Il metodo *few-shot* ha dimostrato un netto miglioramento rispetto all'approccio *zero-shot*, raggiungendo un punteggio F1 complessivo di 0,175 – con una precisione di 0,215 e un richiamo di 0,148 – paragonabile alla *baseline* inglese. Su 530 previsioni, 114 erano corrette rispetto ai 771 sinonimi del *ground truth*. A differenza della *baseline*, questo approccio ha ottenuto prestazioni migliori sui termini monosemici (punteggio F1: 0,212) rispetto a quelli polisemici (punteggio F1: 0,159). Questo suggerisce che

anche un piccolo numero di esempi può migliorare significativamente la capacità del modello di generare lemmi latini annotabili come potenziali sinonimi.

L'approccio di fine-tuning utilizzando LoRA ha mostrato il miglioramento più sostanziale, superando sia la *baseline* inglese che i metodi *zero-shot* e *few-shot* per il latino, con un punteggio F1 complessivo di 0,336. Ha raggiunto una precisione complessiva di 0,487 e un richiamo di 0,256. Su 464 previsioni, 226 erano corrette rispetto agli 882 sinonimi del *ground truth*. In particolare, per i termini polisemici ha raggiunto un punteggio F1 di 0,373 (precisione: 0,669, richiamo: 0,258), mentre per i termini monosemici il punteggio F1 era 0,221 (precisione: 0,200, richiamo: 0,247). Ha quindi dimostrato un significativo incremento delle prestazioni per i termini polisemici (punteggio F1: 0,373) rispetto a quelli monosemici (punteggio F1: 0,221).

In tutti gli approcci, abbiamo osservato una tendenza generale a un richiamo più basso rispetto alla precisione, suggerendo che i modelli sono più conservativi nelle loro previsioni ma relativamente accurati nella generazione. Il modello *fine-tuned* ha mostrato il compromesso precisione-richiamo più bilanciato, in particolare per i termini polisemici (precisione: 0,669, richiamo: 0,258).

La progressione delle performance a partire dalla *baseline* inglese attraverso i successivi approcci al latino rivela diverse tendenze interessanti nelle prestazioni di generazione dei sinonimi. L'approccio *zero-shot*, pur generando un numero simile di previsioni (500) rispetto alla *baseline* inglese (499), ha mostrato un calo significativo dell'accuratezza, con la sua precisione (0,094) e richiamo (0,066) entrambi ben al di sotto della baseline. Questo sottolinea la sfida di trasferire la conoscenza linguistica generale a un compito specializzato in una lingua antica senza adattamento specifico. Il metodo *few-shot* ha segnato un sostanziale miglioramento rispetto all'approccio *zero-shot*, portando le prestazioni vicine, e in alcuni aspetti superando, l'inglese. Con 530 previsioni e 114 sinonimi corretti, ha dimostrato che anche un piccolo numero di esempi può migliorare significativamente la capacità del modello di generare potenziali sinonimi latini. In questo caso possiamo inoltre notare che, come un *outlier*, le sue prestazioni sui termini monosemici (F1: 0,212) hanno superato quelle dei termini polisemici (F1: 0,159), in contrasto con la tendenza della *baseline* e degli altri approcci adottati. Il modello *fine-tuned*, tuttavia, ha mostrato il miglioramento più evidente, come

in realtà ci si aspettava. Nonostante generasse meno previsioni (464) rispetto agli altri approcci, ha prodotto il numero più alto di sinonimi corretti (226). Questa efficienza si riflette nella sua superiore precisione (0,487) e richiamo (0,256), entrambi significativamente superiori alla *baseline* inglese e ai precedenti esperimenti. Le prestazioni del modello *fine-tuned* sui termini polisemici sono state particolarmente rilevanti, con un punteggio F1 (0,373) quasi doppio rispetto ai termini monosemici (0,221), indicando una certa comprensione delle parole latine complesse e polisemiche.

Questa progressione evidenzia l'efficacia dell'adattamento specifico per la generazione di lemmi latini in un task specifico, la generazione di sinonimi (o potenziali tali), come quello in esame. Ogni passaggio da *zero-shot* a *few-shot* a *fine-tuning* ha mostrato miglioramenti cumulativi in precisione, richiamo e punteggio F1: il sostanziale miglioramento delle prestazioni, dal modesto punteggio F1 di 0,078 nell'approccio zero-shot al 0,336 del modello *fine-tuned*, evidenzia il potenziale delle tecniche di adattamento specifico per compiti linguistici complessi. La capacità del modello *fine-tuned* di superare la *baseline* inglese, specialmente nella generazione di termini polisemici, sottolinea il valore dell'addestramento mirato nel catturare la complessità del task e del vocabolario stesso. Il miglioramento bilanciato del modello *fine-tuned* in tutte le metriche dimostra che, con tecniche di adattamento appropriate, è possibile ottenere risultati di alta qualità in compiti linguistici estremamente specifici, anche per lingue poco rappresentate.

Inoltre, la disparità nelle prestazioni tra termini polisemici e monosemici è particolarmente interessante dal punto di vista linguistico, offrendo spunti sulla capacità del modello di navigare la complessità semantica. Le prestazioni superiori sui termini polisemici (punteggio F1 0,373 vs 0,221 per i monosemici) suggeriscono che il modello sfrutta efficacemente il campo semantico più ampio associato alle parole polisemiche per generare più sinonimi potenzialmente corretti; inoltre – come discuteremo più ampiamente in §3.2 – è importante notare che nel dataset polisemico è più probabile trovare lemmi piuttosto comuni e frequenti (i quali possiamo presumere che siano già stati incontrati dal modello durante il *pre-training*). D'altra parte, la maggior parte dei termini monosemici sono rari e meno frequenti, ma allo stesso tempo la loro limitata

variazione semantica sembra vincolare la capacità del modello di generare potenziali sinonimi diversificati, risultando in un *recall* più basso.

Questi risultati si allineano con le ricerche precedenti sulle sfide della disambiguazione del senso delle parole e della generazione di sinonimi nell'elaborazione del linguaggio naturale (Navigli, 2009; Ide e Véronis, 1998). Il compito di identificare e generare accuratamente i sinonimi richiede non solo una profonda comprensione delle relazioni semantiche ma anche la capacità di discernere sottili sfumature nei significati delle parole e nei contesti d'uso, che rimane una sfida significativa per gli attuali LLM (Camacho-Collados e Pilehvar, 2018).

3.2. Analisi qualitativa

le performances descritte nella sezione 3.1 verranno ora discussi da una prospettiva qualitativa, al fine di comprendere meglio i risultati del modello in ogni fase dell'esperimento, evidenziando problematiche come le allucinazioni, il disallineamento del task e le differenze tra i dataset monosemico e polisemico.

La Tabella 3 fornisce due esempi, che illustrano la progressione della generazione di lemmi attraverso gli approcci *zero-shot*, *few-shot* e *fine-tuning* per entrambi i dataset polisemico e monosemico. Questa progressione dimostra come le prestazioni del modello cambiano con diversi livelli di training specifico per il task.

Dataset	Word	Zero-shot	Few-Shot	Fine-tuning
Monosemico	pinna	auris, pinne, pinnulae, pilae, pars alaris externa, caudum, crus	**pinnula**, remigium	**penna**, **ala**, vexillum
Punteggio		0/7	1/2	2/3
Polisemico	adiuuo	adiutor, auxilium, consolatio	**auxilio**, consilior, **defendo**, accomodo, addito, aficio, apporto, augendo, sublevato,	**auxilio**, **fortifico**, **sustineo**, **uindico**, **succuro**, sublego, sufficio, supporto, **teneo**

			succurro, advoco, annoso, sequoro, supplico	
Punteggio		0/3	4/14	6/9

Tabella 3. Esempi tratti dal processo di generazione di lemmi latini

3.2.1. Zero-shot e Few-shot

Partendo dall'approccio *zero-shot*, il modello ha performato ben al di sotto della baseline (cfr. sezione 3.1). Nonostante le istruzioni esplicite fornite nei prompt (cfr. sezioni 2.4.1. e 2.4.2.), le generazioni includono risultati inadeguati, come parole inglesi traduzioni del lemma target, definizioni del lemma, forme flesse latine e pseudo-parole. Queste pseudo-parole sono un sottoprodotto delle allucinazioni. Le allucinazioni nei LLM sono un fenomeno ben documentato per cui il modello genera contenuti che sono fattualmente incorretti o privi di senso, ma presentati con sicurezza (Maynez et al., 2020, Kadavath et al., 2022). Nel contesto di questo task, le allucinazioni si manifestano come la generazione di parole latine che suonano plausibili ma non esistenti (Ji et al., 2022); questo comportamento si allinea dunque con le osservazioni in altri studi che mostrano come i LLM producano risposte fluenti ma inaccurate, specialmente in domini specializzati o lingue con poche risorse (Maynez et al., 2020)[32].

Un esempio di una generazione zero-shot per lemmi del dataset di monosemia è dato in (1), mentre (2) esemplifica i risultati per lemmi del dataset di polisemia:[33]

1. ELOCUTILIS: **eloquens**, *eloquentior*, *etiam loquax*, **verbosus**, *atque fluentissimus*, *fluidores*. Punteggio: 2/6
2. ADHORTATIO: *exhortationem*, **admonitio**, *monitus esto*, *instigare*, *provocatum esse*. Punteggio: 1/5

[32] Per maggiori informazioni sulle allucinazioni, si veda nota 16.

[33] Gli esempi presentano il lemma di input in lettere maiuscole, mentre le generazioni sono così contrassegnate: i potenziali sinonimi sono in grassetto (es. **admonitio**), i lemmi latini esistenti che non sono sinonimi validi sono in tondo e le generazioni corrispondenti a parole inglesi, espressioni multi-parola, forme flesse latine o pseudo-parole sono riportate in corsivo (es. *to mix, etiam loquax, eloquentior, fluidores*).

Un altro fattore che influisce negativamente sui risultati è la tendenza del modello a generare parole con una parte del discorso (POS) diversa dall'input, che non soddisfa l'obiettivo del task. Questo fenomeno, spesso definito come *task misalignment* o *goal misgeneralization*, si verifica quando il modello non riesce a cogliere pienamente o ad aderire ai requisiti specifici del task assegnato (Shah et al., 2022). Nel nostro caso, la propensione del modello a generare parole con POS errata suggerisce un fallimento nel mantenere vincoli morfosintattici coerenti tra le coppie input-output.

Confrontando questi risultati con la baseline inglese, emergono alcune problematiche comuni, specificamente, il mismatch di POS tra l'input e le generazioni e l'inclusione di espressioni multi-parola, pseudo-parole e forme flesse nei risultati. Un esempio dal dataset monosemico della *baseline* è mostrato in (3) e uno dal dataset polisemico della *baseline* in (4):

3. PRUNING: **trimming**, *cutting back*, *prune*, *a plant*, *cut off branches from*. Punteggio: 1/5.
4. CHAINING: strolling, marching, treading, wandering, ambling, *saunters*, striding, joyfully. Punteggio: 0/8

Questi esempi evidenziano la complessità del task di popolare i synset, anche in una lingua estremamente rappresentata come l'inglese. Il modello fatica a restituire coerentemente elementi di una sola parola, fornendo spesso definizioni (es. *cut off branches from*). Inoltre, l'inclusione di forme flesse (es. *saunters*) e parole che sono completamente disallineate con la POS della parola target (es. *joyfully* per CHAINING) illustra ulteriormente la difficoltà del task. Questa complessità è ulteriormente evidenziata dalla nostra analisi quantitativa della baseline inglese, discussa nella sezione 3.1. Questi risultati evidenziano le sfide intrinseche nella generazione di sinonimi sotto un profilo generale; certo è che tali problematiche sono molto più frequenti nel task di generazione di sinonimi latini, avendo quindi un impatto maggiore sui risultati.

Possiamo notare, negli esempi seguenti, come i risultati dell'approccio *zero-shot* includano talvolta porzioni di testo non correlate al task, per esempio istruzioni date nei prompt, come in (5), e testi probabilmente recuperati dai dati utilizzati per

l'addestramento del modello, come in (6), (7), e (8), in cui ciò che sembrano essere glosse e tag morfologici sono riportati nella generazione:

5. ACCIO: *exigere*, *esse*, *your response should be a json object containing an array of strings*. Punteggio: 0/3
6. MULUS: *bovine mammalian species, cow, oxen, nominative singular masculine case, bos, bull, nominative plural feminine case, bae.* Punteggio 0/9
7. AGO: *person singular indicative active present tense neuter nominative case*, *agit*, *declension noun*, actio, *accusative plural feminine case*, *actiones*. Punteggio: 0/6
8. AMBULO: *person singular indicative active present tense, ambulor, psita, amabis, psictum, amavit etc* 0/5

Output come (6) e (7) suggeriscono la presenza di dati linguistici latini nel corpus di pre-training del modello. L'inclusione di glosse morfologiche e forme flesse suggerisce che il modello è stato esposto a descrizioni grammaticali latine e voci di dizionario come anche *treebank open-access* e corpora annotati durante la sua fase di training. Va menzionato che il nostro uso del metodo *zero-shot* è servito come cartina di tornasole, permettendoci di valutare l'estensione della conoscenza del latino già incorporata nel modello. La tendenza del modello a riprodurre frammenti di descrizioni morfologiche piuttosto che fornire semplici sinonimi indica che, mentre il modello ha una certa conoscenza del latino, fatica ad applicare questa conoscenza in modo appropriato nel contesto specifico del nostro task.

Allo stesso tempo però, sia relativamente al dataset monosemico (9) che al dataset polisemico (10) ci sono state generazioni meno fallimentari:

9. AFER: **africanus**, **africus**, afriensis. Punteggio: 2/3
10. ABUNTANTIA: *abundare, copiosus, affluere,* **plenitudo**, *superfluum*, *excessu*, **saturitas**, *exuberans*, *multitudine*, **profluitas**, plebs, *turbae*, **populatio**, populus, *multiudinis, turbines,* tumultus*, agmines, gregis, cohortiumque, legionemque, exercitu, equitesmilitemeque, militareseque, omniaquae, universaeque,*

totiusmagnitude, magnifice, grandiosepraestentibus, praecipueprincipales, primoresprimos, summosmaximioptimom. Punteggio 3/33

In (9) due dei lemmi generati sono reali sinonimi di AFER, attualmente annotati nel LWN. Per quanto riguarda l'output che osserviamo in (10), **plenitudo** è anch'esso annotato nel LWN come parte del synset di ABUNDANTIA, come anche **saturitas**. **Profluitas** e **populatio** sono considerati potenziali sinonimi a partire dalla definizione che troviamo nel dizionario LS "*abundance, plenty, fullness, richness*", significato che relativamente a **populatio** entra in uso (sempre secondo LS) a partire dal latino tardo. Sempre in merito a **populatio**, nel TLL è definita come *turba hominum* e si ritiene interessante riportare che nel LWN condivide un synset con *multitudo* – WordNet Synset 08179879-n, *the people who inhabit a territory or state* –, che a sua volta condivide un synset (ed è quindi sinonimo) con la nostra parola target – *WordNet Synset 05115040-n, the property of a more than adequate quantity or supply*. Risultati di questo tipo indicano che, seppur in minor scala, già a questo stadio dell'esperimento il modello può aggiungere risultati pertinenti al LWN.

In (10) il modello è risultato un po' fantasioso, andando a produrre una serie di allucinazioni piuttosto curiose: ovvero *summosmaximioptimom* o anche *grandiosepraestentibus*; allo stesso tempo però, le forme flesse prodotte, cioè *excessu*, *multiudinis, turbines, gregis, agmines* sembrano comunque rimanere vicine al campo semantico (quanto meno su larga scala) della target word di partenza.

Passando ora all'approccio few-shot, questa svolta metodologica sembra avere un effetto positivo sui risultati, come mostrato dall'analisi quantitativa. In particolare, la presenza di lemmi inglesi ed espressioni multi-parola nelle generazioni diminuisce significativamente, mentre pseudo-parole e forme flesse rimangono frequenti. Gli esempi sono offerti in (7), che presenta le generazioni per un lemma dal dataset monosemico, e (8), che contiene i risultati per un lemma dal dataset polisemico:

7. SUAVILOQUENS: *dilatabiliter*, *loquebatur*, *modestius*. Punteggio: 0/3
8. ACCUMULATIO: **acervus**, *agglomeratum*, **caterva**, *congestion*, **copia**, **grex**, **massa**, mulier, *monte*, *pecuie*, *plebe*. Punteggio: 5/11

9. APER: abscedo, adsum, *amovebo*, *annulo*, *apereo*, *aquireo*, arrepto, *auribus movere*, *auferro*, *averteo*, *avellico*, *benevectare.* Punteggio 0/13
10. AFFLIGO: **amputo**, **attenuo**, **attero**, **cado**, **contraho**, *curto*, *decrescico*, *demino*, **infirmo**, **minuo**. Punteggio 7/10

Il mismatch tra la POS dell'input e quella dei risultati è un problema che influisce anche sulle generazioni ottenute attraverso questo approccio, come possiamo vedere ben evidenziato in (9), dove anche le pseudo-parole – *amovebo, annulo, apereo, aquireo, auferro*, *averteo*, *avellico*, *benevectare* – sembrerebbero appartenere a una POS diversa (*verb*) da quella della parola target (*noun*). In (8), però, dobbiamo anche sottolineare che **acervus**, **caterva**, **grex**, **copia** e **massa** sono effettivi sinonimi di ACCUMULATIO, condividendo con la parola target il synset *WordNet Synset 13774404-n, (often followed by 'of') a large number or amount or extent*.

Nel complesso, come anche discusso nella sezione 3.1., le performance del modello aumentano rispetto all'approccio *zero-shot*, mostrando una maggiore aderenza al task e ai prompt. Di questo miglioramento della performance, l'output osservabile in (10) è un ottimo esempio. Sette delle dieci parole generate sono effettivi sinonimi della parola target: **attero**, **infirmo**, **minuo** e **attenuo** condividono infatti più di un synset con AFFLIGO – i.e. *WordNet Synset 00262881-v, make worse or less effective* (**infirmo** fa parte solo di questo synset); *WordNet Synset 00224901-v, lessen the strength of* (**attenuo** fa parte solo di questo synset); anche **contraho** condivide un synset con AFFLIGO, ovvero *WordNet Synset 01814396-v, lower someone's spirits*; mentre **amputo** e **cado** condividono invece questo synset con la nostra parola target: *WordNet Synset 00151689-v, decrease in size, extent, or range; make downhearted*.

Questi risultati, come già evidenziato in 3.1., mostrano che le performance del modello sono decisamente migliori con il dataset polisemico e nello specifico con i verbi. Questo è probabilmente legato non solo al fatto che i verbi sono più polisemici (Fellbaum, 1990) di altre POS ma anche al fatto che siano parole piuttosto frequenti. In generale, gli esempi analizzati mostrano che l'approccio *few-shot* si rivela utile nell'adattare le prestazioni del modello a una lingua poco rappresentata come il latino.

3.2.2. Fine-Tuning LoRA

Passando ai risultati del *fine-tuning* – come già anticipato in 3.1. in sede di analisi quantitativa – si possono riconoscere grandi miglioramenti sia per i lemmi monosemici (seppure in minor scala) che per i lemmi polisemici. In generale, possiamo dire che la presenza di lemmi inglesi ed espressioni multi-parola nei risultati del fine-tuning è quasi nulla, mentre le forme flesse e le pseudo-parole sono ancora presenti, anche se meno frequentemente rispetto ai risultati degli approcci precedenti. Inoltre, la corrispondenza tra la POS dell'input e quella dei risultati è più alta. Di seguito si riportano alcuni esempi:

11. COGITAMENTUM: **meditatio**, cognomen, *cognomentum*, nomen. Punteggio: 1/4.
12. ABSCONDO: **abscido**, delego, delicio, **detraho**, effingo, extinguo, **pello**, **occulto**, **praeservo**, **servo**, **tego**. Punteggio: 7/11

Un'osservazione interessante deriva dall'esempio (11), in cui **meditatio** è validato come potenziale sinonimo di COGITAMENTUM. Attualmente, nel LWN, *meditatio* e *cogitamentum* non sono considerati sinonimi in quanto non condividono alcun synset. Tuttavia, questi due lemmi appaiono entrambi nei dizionari latini con la definizione "*a thought*" (OLD, s.v. *meditatio*; LS, s.v. *cogitamentum*). TLL inoltre per *meditatio* riporta "*pertinet ad cogitationem*" mentre per *cogitamentum* troviamo "*res excogitata, consilium*" e viene anche riportato anche il riferimento a una definizione glossata in greco "*ἐνθύμημα (mentis conceptio)*". Questo caso dimostra quindi il potenziale dell'approccio adottato in questo lavoro nel supportare gli annotatori umani nel processo di annotazione, identificando relazioni di sinonimia che potrebbero non essere state codificate nel WordNet.

13. ACCIO: **adfero**, **adduco**, *admitteo*, **appello**, aufero, attingo, delego, **dedico**, *dicio*, doceo, edisco, effundo, eicio, **enuntio**, **explico**, **exprimo**, facio, **moneo**, *nuncio*, **proloquor**, *quado*, **refero**, **uoco**. Punteggio: 11/23
14. ABSCONDO: abscido, delego, delicio, **detraho**, effingo, extinguo, **pello**, **occulto**, **praeservo**, **servo**, **tego**. Punteggio: 6/13

Relativamente alla generazione in (13), **appello** e **voco** sono effettivi sinonimi di ACCIO, condividendo più di un synset – i.e. *WordNet Synset 00792471-v*, *order, request, or command to come* – mentre gli altri potenziali sinonimi sono stati considerati tali a partire dal significato generale riportato dal LS "*to call, summon, send for, invite*" come anche quello che ritroviamo in TLL "I. *i.q. arcessere*" e "III. *i.q. vocare, advocare*". Sulla base di ciò sono infatti considerati tali **adfero** e **adduco**, semanticamente anch'essi molto vicini, con significato *to bring to* e *to bring, conduct*; su **adfero** si aggiunge anche che LS riporta *"to report, announce"* come ulteriore significato. Per **dedico** ed **enuntio** sul LS viene riportato il significato di "*to declare*", che può essere considerato contiguo all'atto di "chiamare" evocato da ACCIO; lo stesso vale per **proloquor**, "*to speak out, declare publicly*". Per **refero** si fa riferimento al terzo significato riportato da LS e cioè "*to convey a report, account, intelligence, by speech or by writing; to report, announce*", mentre sono sicuramente più laschi i rapporti di vicinanza semantica con **explico**, **exprimo** e **moneo**, legati al campo semantico identificabile con la classe in VerbNet say-37.7-1-1. Sono comunque presenti anche in (13) alcune allucinazioni, caratterizzate da pseudo-parole, come *admitteo* che sembra essere una fantasiosa versione di *admitto* appartenente alla seconda invece che alla terza coniugazione latina e *nuncio* evidentemente legato a *nuntio* (effettivo sinonimo di ACCIO).

In (14) **occulto**, **servo** e **tego** sono effettivi sinonimi di ABSCONDO e condividono con la parola target anche in questo caso più di un synset – i.e. *WordNet Synset 02733122-v, maintain in safety from injury, harm, or danger*; *WordNet Synset 02145814-v, be or go into hiding*, *keep out of sight, as for protection and safety*; WordNet Synset 02148369-v, *hide from view or knowledge*. **Praeservo**, composto di **servo**, è stato considerato potenziale sinonimo in questo caso nonostante la presenza del preverbo, che rende il suo significato più specifico. **Detraho** dato il suo significato più generico di "*to draw or take off, draw away, draw or take down; to pull down; to take away, remove, withdraw*" è stato in questo caso considerato come potenziale sinonimo; lascamente si considera anche **pello** un potenziale sinonimo sulla base del significato segnato in LS "*to drive out or away*", tenendo conto del significato di ABSCONDO riportato in LS "*to put away, conceal"*, di quanto presente in TLL "*removeo ex*

conspectu" e del synset più generico *WordNet Synset 02145429-v*, *hide from view*. Ovviamente una valutazione più approfondita con attestazioni effettive contestuali potrebbe validare o cassare un'annotazione di questo tipo. Sicuramente è interessante notare come invece in questo esempio, nonostante siano stati generati 13 lemmi, siano totalmente assenti allucinazioni.

Di seguito altre generazioni:

15. ADIUUO: **auxilio**, **fortifico**, **sustineo**, **uindico**, **succuro**, sublego, sufficio, supporto, **teneo**. Punteggio: 6/9
16. ABUNDANTIA: **copiositas**, **opulentia**, **uber**, **ubertas**, sufficitentia. Punteggio: 4/5
17. AEDIFICO: **constituo**, **fabrico**, **facio**, **instauro**, **praeparo**, *producio*, *structo*, texo. Punteggio: 5/8

Cominciamo cono (15): **succurro**, **sustineo** e **teneo** sono effettivi sinonimi di ADIUUO, condividendo synsets con la parola target – *WordNet Synset 01217043-v, be the physical support of; carry the weight of*; *WordNet Synset 02547586-v, give help or assistance; be of service* –, per quanto riguarda **auxilio** è considerata nel LS una forma alternativa di *auxilior*, effettivo sinonimo di ADIUUO – parte del synset *WordNet Synset 02547586-v* –, e di conseguenza potrebbe essere potenzialmente inserito nel synset relativo. **Fortifico** e **uindico** sono considerati potenziali sinonimi a partire da quanto riportato in LS per il lemma target: "*to give aid to, to help, assist, support*"; per **fortifico** è riportato "*to make strong, to strengthen, fortify*", che può essere (seppur alla larga e quindi dovrebbe essere supportato da evidenza contestuale di fonti scritte) semanticamente contiguo all'evento di *support* evocato da ADIUUO; l'atto evocato da **vindico** – LS: *to deliver, liberate, protect, defend; to avenge* – è invece più vicino al campo semantico – per il quale si tiene conto della classe help-72.1 (VerbNet) – di ADIUUO. *Supporto*, di contro, in questo caso è un "falso amico", il suo significato su LS è " *to carry, bring, or convey to a place*", dunque non può essere considerato un potenziale sinonimo di ADIUUO.

Per quanto riguarda (16) i quattro lemmi considerati come potenziali sinonimi, ovvero **copiositas**, **opulentia**, **uber** e **ubertas,** sono effettivi sinonimi di

ABUNDANTIA: *copiositas* e *ubertas* condividono con il lemma target il synset *WordNet Synset 05115568-n* (*a full supply*) e a questi due lemmi si aggiunge *uber* nel synset *WordNet Synset 05115040-n* (*the property of a more than adequate quantity or supply*); mentre *opulentia* lo troviamo insieme ad ABUNDANTIA nel synset *WordNet Synset 13353280-n*, *an abundance of material possessions and resources*.

In (17) invece **fabrico** e **facio** sono effettivi sinonimi di AEDIFICO – *WordNet Synset 01654628-v*, *make by combining materials and parts* – come anche **constituo** *WordNet Synset 02427103-v, set up or found*. **Instauro** è considerato potenziale sinonimo a partire da quanto riportato in LS per il lemma target AEDIFICO: "*to build, erect a building, construct*", infatti per **instauro** sempre su LS è riportato "*to erect, make*", che mostra una sovrapposizione semantica con l'evento di *build/erect* evocato da AEDIFICO. Per **praeparo** invece risulta su LS "*to get or make beforehand*", che mostra una connessione semantica con l'evento di *make* evocato da AEDIFICO, sebbene con una specificazione temporale (*beforehand*) che ne precisa la natura preparatoria rispetto all'atto costruttivo.

In alcuni casi il modello ha generato solo un lemma, che si è rivelato essere un potenziale sinonimo:

18. AMBULO: **curro**. Punteggio: 1/1
19. ANTEEO: **antepono**. Punteggio: 1/1

In (18) **curro** è un effettivo sinonimo di AMBULO, essendo parte con la parola target del synset *WordNet Synset 02684924-v* (*continue a certain state, condition, or activity*) e potenzialmente potrebbe anche essere annotato nel synset *WordNet Synset 01904930-v* (*use one's feet to advance; advance by steps*) considerando il significato riportato da LS di "*to run*", evento che può rientrare nel campo semantico – si fa riferimento in VerbNet alla classe run-51.3.2 – evocato da quest'ultimo synset. Anche **antepono** (20) è un effettivo sinonimo di ANTEEO, condividendo con quest'ultimo il synset *WordNet Synset 02692686-v*, *come before*.

Non tutte le generazioni sono state produttive; di seguito vengono riportati alcuni esempi tratti dal dataset monosemico (20, 21, 22) e dal dataset polisemico (23) che non hanno generato potenziali sinonimi:

20. ASTRIFER: asteriscus, stellula, stella, sidus, sol. Punteggio: 0/5
21. ABNEGATIUUS: *abnegotiatus.* Punteggio: 0/1
22. AIO: dicere, loqueri, proferre, profiteri, uocari, *urbere*, uitare. Punteggio: 0/7
23. ANIMADUERTO: praedico. Punteggio: 0/1

In (20) notiamo un task-misalignment basato sulla POS: ASTRIFER è un aggettivo, e di contro sono stati generati solo nomi (e nessuna pseudo-parola), benché il campo semantico (su LS per ASTRIFER troviamo *"I. starry; II. placed among the stars"*) sia stato correttamente individuato dal modello. In (21) viene generata invece solo una pseudo-parola. Sicuramente molto interessante è il caso di (22): se fosse stato generato correttamente il verbo alla prima persona singolare – invece di output dell'evidente task-misalignment – avremmo sicuramente avuto quattro sinonimi effettivi di AIO – *dico, loquor, profero, profiteor* – e la forma passiva di *uoco*. Per quanto riguarda invece (23), nonostante ANIMADUERTO (LS: "*to direct the mind or attention to, give heed to, take notice of, attend to, consider, regard, observe*") e *praedico* ("*to say or tell beforehand, foretell, predict, forewarn*") possano presentare alcuni elementi di apparente vicinanza semantica – entrambi implicano un'attenzione verso qualcosa – non possono essere considerati sinonimi in quanto ANIMADUERTO si concentra sull'atto dell'osservare e considerare attentamente, mentre *praedico* si riferisce specificamente all'atto di predire o avvertire in anticipo, due azioni che, seppur possano essere correlate, appartengono a campi semantici distinti.

Va menzionato inoltre che il modello ha prodotto output vuoti in tre occasioni durante il task di generazione dei sinonimi: una volta per un lemma monosemico (*commisereor*) e due volte per lemmi polisemici (*carpo*, *circumscriptio*). Questo fenomeno è stato altrimenti osservato solo una volta, specificamente con l'approccio zero-shot sul dataset monosemico (*actutum*). Mentre la generazione di un output vuoto è inconcludente per il task in questione, allo stesso tempo potrebbe essere un segno di miglioramento e adattamento del modello, mostrando una preferenza per la generazione di un output vuoto invece di risultati non correlati.

Possiamo notare come il modello in seguito al *fine-tuning* performi meglio con lemmi polisemici piuttosto che con lemmi mononsemici. Inoltre mostra risultati più

incoraggianti nella generazione di potenziali sinonimi per i verbi rispetto ad altre POS. Questa performance migliore con i lemmi polisemici può essere parzialmente attribuita alla natura del processo di generazione stesso. Poiché l'output del modello si basa su una predizione stocastica, i lemmi polisemici offrono uno spazio semantico più ampio da cui generare potenziali sinonimi, aumentando la probabilità di produrre risposte corrette. Questo fenomeno si allinea con diversi studi nel campo del NLP e della scienza cognitiva. Per esempio, Pilehvar et al. (2019) discutono di come la disambiguazione del senso delle parole benefici dello spazio semantico ricco dei lemmi polisemici nei modelli di spazio vettoriale, che è analogo alla nostra osservazione nella generazione di sinonimi. Similmente, Ethayarajh (2019) dimostra che gli embedding contestuali delle parole catturano più informazioni per i lemmi polisemici grazie ai loro vari contesti d'uso. Complessivamente, questi studi supportano l'idea che la più ricca variazione semantica dei lemmi polisemici possa portare a prestazioni migliori in vari task linguistici, inclusa, come osserviamo, la generazione di sinonimi con modelli linguistici.

Mentre questa tendenza verso prestazioni migliori con lemmi polisemici emerge chiaramente dai risultati di questo lavoro ma non è una novità in letteratura, un esame più attento rivela un pattern interessante specifico del dataset usato. Le prestazioni del modello sembrano essere particolarmente buone con i verbi. Questo pattern persiste nonostante la composizione dei nostri dati di training (cfr. sezione 2.1): anche se i sostantivi sono quasi il doppio del numero di verbi nei dati di training, il modello performa comunque meglio con i verbi. Questa osservazione può essere parzialmente spiegata dalla composizione del nostro dataset: i verbi sono molto più rappresentati nel dataset polisemico che in quello monosemico (28 vs 6 verbi). Questo squilibrio è probabilmente dovuto al fatto che i verbi sono intrinsecamente più polisemici di altre POS. Per esempio, Gentner (1988) ha dimostrato che i verbi sono più propensi a estendere i loro significati in nuovi contesti rispetto ai sostantivi. Similmente, Fellbaum (1990) ha mostrato che i verbi hanno un grado più alto di polisemia nel Princeton WordNet rispetto ai sostantivi.

D'altra parte, la differenza nelle prestazioni potrebbe essere spiegata considerando i lemmi che costituiscono il dataset monosemico: la maggior parte

sembrano essere parole rare, spesso associate a un significato molto specifico e con pochi sinonimi. I problemi con i lemmi presenti nel dataset monosemico possono quindi essere spiegati dal fatto che sono sotto-rappresentati nel dataset di pre-training a causa della loro bassa frequenza e anche dal fatto che hanno pochi sinonimi per la specificità del loro significato. Questa ipotesi è supportata dal fatto che il modello ottiene una performance soddisfacente con lemmi monosemici abbastanza frequenti, come in (24) e (25):

24. ASPORTATIO: *abductionem*, **captura**, carnificina, *furta*, **rapina**, *stulcium*, *ueneficiam*, **latrocinium**, *strage*, pugna, bellum, *luparium*, *saeculariua*, nex, *mordebatio*, *praedae*, **spoliatio**. Punteggio: 4/17
25. POLLICITATIO: **votum**, **fides**, **foedus**, **pactum**, *sancimentum*, testamentum. Punteggio: 4/6

In (24) **captura**, **rapina**, **latrocinium** e **spoliatio** sono considerati potenziali sinonimi a partire dalla definizione di ASPORTATIO – che fa parte del synset *WordNet Synset 00391599-n*, *the act of removing* – nel LS: "*a carrying away*". Vale la pena sottolineare come il campo semantico di questo lemma rimanga molto ampio e di conseguenza questo porti con sé risultati migliori rispetto ad altri lemmi del dataset monosemico. Inoltre questo ampio significato garantisce una maggiore libertà di generazione, che in questo caso porta con sé l'output di molte pseudo-parole. Relativamente alla specifica generazione presentata in (24), per **captura** è riportato "*a taking, catching*", per **rapina** è riportato "*a collecting together, removing*", per *latrocinium* è riportato "*robbery*", per **spoliatio** è riportato "*a pillaging, robbing, plundering*", tutti termini che condividono il campo semantico dell'azione di portare via/rimuovere qualcosa (si fa riferimento alla classe remove-10.1 in VerbNet) evocato da ASPORTATIO, sebbene con diverse sfumature di significato che vanno dal semplice "*take*" fino all'atto più specifico di "*strip*" (VerbNet).

In (25) **votum** e **fides** sono effettivi sinonimi di POLLICITATIO, condividendo il synset *WordNet Synset 07226545-na*, *verbal commitment by one person to another agreeing to do (or not to do) something in the future*. Invece, **foedus**, **pactum** e **testamentum** sono considerati potenziali sinonimi del lemma target, a partire da quanto

riportato nel LS per POLLICITATIO: "*a promising, a promise*"; per **foedus** è riportato "*a league, treaty, compact*", mentre per **pactum** è riportato "*agreement, covenant, contract, bargain*", tutti termini che condividono il campo semantico dell'impegno formale/promessa evocato da POLLICITATIO, sebbene con diverse specificità. In conclusione, l'esperimento – in particolare il *fine-tuning* – ha rivelato pattern complessi che vanno oltre le semplici differenze di performance basate sull'opposizione monosemia-polisemia. L'analisi qualitativa ha evidenziato quanto già emerso nell'analisi quantitativa, ovvero come il successo nella generazione di sinonimi sia influenzato da molteplici fattori interconnessi: la frequenza d'uso dei lemmi, la loro specificità semantica e la polisemia, la parte del discorso.

La maggiore efficacia del modello con i verbi e con i lemmi polisemici suggerisce che la ricchezza del contesto semantico gioca un ruolo cruciale nella performance. Inoltre, le sfide incontrate con i lemmi monosemici, e specialmente con termini particolarmente rari con significati altamente specifici, evidenziano l'importanza di considerare la frequenza delle parole e la specificità semantica nel training e nella valutazione del modello.

Conclusioni

Questo studio ha esplorato l'utilizzo dei LLM per l'arricchimento del Latin WordNet (LWN) attraverso la generazione automatica di sinonimi, confrontando specificamente gli approcci *zero-shot*, *few-shot* e *fine-tuning*. I risultati hanno evidenziato diverse scoperte significative e potenziali percorsi per l'avanzamento dell'uso dei LLM nell'arricchimento di risorse lessicali per lingue antiche e poco rappresentate come il latino.

L'approccio *zero-shot* ha gettato le fondamenta iniziali per la generazione di sinonimi in latino, ma ha mostrato una precisione limitata, evidenziando la difficoltà di applicare direttamente i LLM alle lingue antiche senza un adattamento specifico per il task. L'approccio *few-shot* ha mostrato un miglioramento significativo nel popolamento dei synset, suggerendo che anche un numero limitato di esempi specifici può migliorare considerevolmente le prestazioni del modello. I risultati più significativi sono stati però ottenuti attraverso l'approccio di fine-tuning utilizzando la tecnica LoRA. Questo metodo ha prodotto risultati superiori rispetto agli approcci *zero-shot* e *few-shot*, in particolare nella generazione di sinonimi per termini polisemici. La progressione delle performance attraverso i diversi approcci metodologici dimostra l'importanza dell'adattamento specifico quando si lavora con lingue poco rappresentate.

Questi risultati non solo migliorano la nostra comprensione della generazione automatica di sinonimi per il latino, ma forniscono anche intuizioni sulle più ampie sfide del processamento delle lingue antiche e della gestione della complessità semantica nell'elaborazione del linguaggio naturale. Lo studio ha dimostrato come i LLM, opportunamente adattati, possano fornire un supporto significativo nel processo di annotazione dei synset, offrendo uno strumento prezioso per gli annotatori umani.

La ricerca futura potrebbe esplorare diverse direzioni di miglioramento: lo sviluppo di modelli che bilancino meglio le prestazioni tra diverse parti del discorso e gradi di polisemia, potenzialmente incorporando informazioni etimologiche o utilizzando dati cross-linguistici da lingue correlate; l'implementazione di tecniche di *grounding* più sofisticate per ridurre le allucinazioni del modello, come l'integrazione sistematica con dizionari quali l'Oxford Latin Dictionary e il Lewis & Short,

l'implementazione di controlli morfologici rigorosi per la validazione delle forme generate e lo sviluppo di un sistema di vincoli semantici basato sui campi semantici del latino e sulle relazioni gerarchiche esistenti nel WordNet. L'uso di dizionari di dominio per il training potrebbe inoltre supportare la gestione di lemmi altamente specifici, aumentando l'accuratezza semantica del modello in contesti specialistici.

Queste tecniche di *grounding* potrebbero essere accompagnate da strategie di *data augmentation* per arricchire il dataset di *training*, particolarmente utili per cercare di bilanciare al meglio il nostro dataset con l'obiettivo di migliorare le performance in maniera sostanziale anche relativamente a lemmi meno frequenti. Inoltre, l'analisi di come questo tipo di esperimento e i conseguenti risultati si applichino ad altre lingue antiche nello scope del progetto *Linked WordNets for Ancient Indo-European Languages* (dunque greco antico e sanscrito) potrebbe fornire preziose intuizioni sull'universalità di questi pattern di elaborazione semantica nella linguistica computazionale.

Nonostante le sfide intrinseche nel lavorare con lingue antiche, i risultati ottenuti con il nostro modello *fine-tuned* suggeriscono la possibilità di automatizzare parzialmente il processo di annotazione dei synset, fornendo un supporto considerevole agli annotatori. Questo apre nuove prospettive per l'arricchimento di risorse lessicali per lingue antiche, combinando l'efficienza degli strumenti computazionali con l'expertise umana. La conoscenza contestuale è infatti essenziale per risolvere ambiguità semantiche, specialmente nei casi di lemmi polisemici o termini specialistici che richiedono una comprensione sfumata delle loro diverse accezioni. Gli annotatori umani possono anche contribuire a valutare l'adeguatezza dei sinonimi suggeriti dal modello, garantendo che ogni lemma generato sia collocato nel synset appropriato, soprattutto quando i termini presentano multiple interpretazioni. Il modello, quindi, può agire come uno strumento di suggerimento, fornendo una prima selezione di sinonimi e categorie semantiche che i linguisti possono successivamente affinare, assicurando così che le risorse lessicali risultanti siano di alta qualità e accuratamente contestualizzate. In sintesi, l'integrazione di strumenti computazionali con il giudizio umano velocizza il processo di annotazione, arricchendo le risorse lessicali con maggiore accuratezza, garantendo che i risultati siano sia efficienti sia affidabili.

Bibliografia e Sitografia

AWOL – The Ancient World Online. (2019). Latin WordNet 2.0. http://ancientworldonline.blogspot.com/2019/06/latin-wordnet-20.html. Ultima visita: 10 ottobre 2024.

Global WordNet Association. (n.d.). English WordNet on GitHub. https://github.com/globalwordnet/english-wordnet. Ultima visita: 9 agosto 2024.

IBM. (n.d.). Large Language Models. https://www.ibm.com/topics/large-language-models. Ultima visita: 9 agosto 2024.

Istituto di Linguistica Computazionale "Antonio Zampolli" - Consiglio Nazionale delle Ricerche. (n.d.). EuroWordNet 2 Project. https://www.ilc.cnr.it/progetti/eurowordnet-2/. Ultima visita: 9 agosto 2024.

Lewis & Short's Latin Dictionary. Perseus Digital Library. http://www.perseus.tufts.edu/hopper/text?doc=Perseus%3atext%3a1999.04.0059. Ultima visita: 7 novembre 2024.

LiLa: Linking Latin. https://lila-erc.eu/. Ultima visita: 7 novembre 2024.

Mistral AI. (2023). Announcing Mistral 7B. https://mistral.ai/news/announcing-mistral-7b/. Ultima visita: 20 ottobre 2024.

Statista. (n.d.). Number of search terms in internet research in the US. https://www.statista.com/statistics/269740/number-of-search-terms-in-internet-research-in-the-us/. Ultima visita: 9 agosto 2024.

Thesaurus linguae Latinae Online. Berlin, Boston: De Gruyter. https://www.degruyter.com/database/tll/html. Ultima visita: 7 novembre 2024.

University of Colorado. (n.d.). UVI. https://uvi.colorado.edu. Ultima visita: 9 novembre 2024.

University of Exeter. (n.d.). Latin WordNet. https://latinwordnet.exeter.ac.uk. Ultima visita: 10 ottobre 2024.

Aghajanyan, A., Zettlemoyer, L., & Gupta, S. (2021). Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics.

Agirre, E., & Edmonds, P. (Eds.). (2007). Word Sense Disambiguation: Algorithms and Applications. Springer.

Ainslie, J., Fullerton, B., O'Brien, L., Thickstun, J., Pham, T., & Aina, L. (2023). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv preprint arXiv:2305.13245.

AlMousa, M., Benlamri, R., & Khoury, R. (2021). Semantic similarity measures: An overview of path-based, feature-based, IC-based, and hybrid approaches. Knowledge-Based Systems, 212, 106565.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc.

Barbouch, S., Verberne, S., & Verhoef, T. (2021). WN-BERT: Integrating WordNet with BERT for Enhanced Lexical Semantic Understanding. Journal of Computational Linguistics, 47(2), 411-428.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150.

Bender, E. M. (2011). On achieving and evaluating language-independence in NLP. Linguistic Issues in Language Technology, 6(3), 1-26.

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. Computational Linguistics, 45(3), 499-654.

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Bender, E. M., et al. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610-623.

Bernier-Colborne, G., & Barriere, C. (2018). Crim at SemEval-2018 Task 9: A Hybrid Approach to Hypernym Discovery. Proceedings of The 12th International Workshop on Semantic Evaluation, 725-731.

Biagetti, E., Zanchi, C., & Luraghi, S. (2021). Linking the Sanskrit WordNet to the Vedic Dependency Treebank: A pilot study. In Proceedings of the 12th Global WordNet Conference, 1851-1858. DOI: 10.5281/zenodo.3518774.

Biagetti, E., Zanchi, C., & Luraghi, S. (Eds.). (2021). Building new resources for historical linguistics. Pavia University Press. ISBN: 978-88-6952-132-4

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

Bisong, E. (2019). Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59-64). Apress.

Bizzoni, Y., Boschetti, F., Diakoff, H., et al. (2014). The Making of Ancient Greek WordNet. https://www.aclweb.org/anthology/L14-1054/.

Bizzoni, Y., Del Gratta, R., Boschetti, F., & Reboul, M. (2015). Enhancing the Accuracy of Ancient Greek WordNet by Multilingual Distributional Semantics. In Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015 (pp. 47-50). Accademia University Press. DOI: 10.4000/books.aaccademia.1312.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Boschetti, F. (2019). Semantic Analysis and Thematic Annotation. In M. Berti (Ed.), Digital Classical Philology (pp. 321-339). Berlin: DeGruyter.

Bosselut, A., et al. (2019). COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 4762-4779.

Bouraoui, Z., et al. (2020). Inducing Relational Knowledge from BERT. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 7456-7463.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

Camacho-Collados, J., & Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. Journal of Artificial Intelligence Research, 63, 743-788.

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2016). From Word to Sense Embeddings: A Survey on Vector Representations of Meaning. Journal of Artificial Intelligence Research, 55, 141-188.

Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Rebouças Filho, P. P. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.

Carpuat, M., & Wu, D. (2007). How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI).

Chaplot, D. S., & Salakhutdinov, R. (2018). Knowledge-based word sense disambiguation using neural networks. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

Chen, X., Li, S., Xu, C., & Wang, D. (2023). Comparative Analysis of Fine-tuning Methodologies for Large Language Models. International Conference on Learning Representations.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating Long Sequences with Sparse Transformers. arXiv preprint arXiv:1904.10509.

Cho, K., et al. (2020). Hypo2path: An Encoder-Decoder Model for Hypernym Prediction. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 6436-6442.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2022). PaLM: Scaling Language Modeling with Pathways. arXiv preprint arXiv:2204.02311.

Cimiano, P., Chiarcos, C., McCrae, J., & Garcia, J. (2020). Linguistic Linked Data: Representation, Generation and Applications. Berlin: Springer.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior, 8(2), 240-247.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12, 2493-2537.

Da Costa, F., & Bond, F. (2016). Improving the Coverage of Pronouns and Exclamatives in WordNet. International Journal of Lexicography, 29(4), 491-507.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv preprint arXiv:2305.14314.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

Devlin, J., et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., ... & Zheng, H. (2021). Few-NERD: A few-shot named entity recognition dataset. arXiv preprint arXiv:2105.07464.

Dizionari:

Držík, P., & Šteflovič, J. (2023). Enhancing Word Embeddings with Synset Information from WordNet. Journal of Artificial Intelligence Research, 68, 1-24.

Erica Biagetti, Chiara Zanchi, and William Michael Short. (2021) Toward the creation of WordNets for ancient Indo-European languages. In Proceedings of the 11th Global Wordnet Conference, pages 258–266, University of South Africa (UNISA). Global Wordnet Association.

Esuli, A., & Sebastiani, F. (2006). SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 06).

Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 55-65.

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical Neural Story Generation. arXiv preprint arXiv:1805.04833.

Fellbaum, C. (1990). English verbs as a semantic net. International Journal of Lexicography, 3(4), 278-301.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press.

Fellbaum, C. (Ed.). (1998). WordNet: An Electronic Lexical Database. MIT Press.

Franzini, G., Peverelli, A., Ruffolo, P., Passarotti, M., Sanna, H., Signoroni, E., Ventura, V., & Zampedri, F. (2019). Nunc Est Aestimandum: Towards an Evaluation of the Latin WordNet. In Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2019), 429-439. DOI: 10.5281/zenodo.3518774.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics.

Gardner, M., et al. (2018). AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv preprint arXiv:1803.07640.

Gentner, D., & France, I. M. (1988). The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. In Lexical ambiguity resolution (pp. 343-382). Morgan Kaufmann.

Glare, P. G. W. (Ed.). (1968). Oxford Latin Dictionary. Oxford: Oxford University Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In European conference on information retrieval (pp. 345-359). Springer, Berlin, Heidelberg.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

Hill, F., Reichart, R., & Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics, 41(4), 665-695.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019). The Curious Case of Neural Text Degeneration. arXiv preprint arXiv:1904.09751.

Holtzman, A., Buys, J., Forbes, M., & Choi, Y. (2020). The Curious Case of Neural Text Degeneration. International Conference on Learning Representations.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., & Attariyan, M. (2019). Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning (pp. 2790-2799).

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% Solution. In Proceedings of the Human Language Technology Conference of the NAACL.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 328-339).

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685.

IBM. (2021). AI for smarter business. IBM Press.

Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2016). Embeddings for Word Sense Disambiguation: An Evaluation Study. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 897-907.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. Computational Linguistics, 24(1), 2-40.

Ivanov, O., Li, M., & Johnson, J. (2023). Efficient Fine-tuning of Quantized LLMs on Consumer GPUs. Conference on Neural Information Processing Systems.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., & Fung, P. (2022). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 1-38.

Jiang, J., & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the International Conference on Research in Computational Linguistics (pp. 19-33).

Jiang, Y., Zhu, H., Zhao, R., Yuan, T., Li, P., & He, X. (2023). Large Language Models: A Comprehensive Survey of Multilingual Representation. arXiv preprint arXiv:2309.00942.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 6282-6293).

Jurafsky, D., & Martin, J. H. (2018). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd ed.). Prentice Hall.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). CTRL: A Conditional Transformer Language Model for Controllable Generation. arXiv preprint arXiv:1909.05858.

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The Efficient Transformer. International Conference on Learning Representations.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. Advances in Neural Information Processing Systems, 35, 22199-22213.

Kumar, A., Singh, R., & Talwar, D. (2023). MistralInstruct: Optimizing Language Models for Instruction Following. Conference on Neural Information Processing Systems.

Kutuzov, A., Kuzmenko, E., & Dorofeev, K. (2018). Building Distributed Word Vectors from Wikipedia Using WordNet-based Semantic Relatedness. In Proceedings of the 27th International Conference on Computational Linguistics.

Lauscher, A., Vulić, I., Ponti, E. M., Korhonen, A., & Glavaš, G. (2020). Specializing unsupervised pretraining models for word-level semantic similarity. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 1371-1383).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Lesk, M. (1986). Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the 5th Annual International Conference on Systems Documentation (pp. 24-26).

Lesk, M. (1986). Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. Proceedings of the 5th Annual International Conference on Systems Documentation, 24-26.

Lewis, C. T., & Short, C. (1879). A Latin Dictionary. Oxford: Clarendon Press.

Lexicon of Embodied Experience in Latin. (2023). Retrieved from LexELat.

Li, W., Yu, Z., & Zhou, B. (2024). MathMistral: Specialized Mathematical Reasoning through Targeted Fine-tuning. arXiv preprint arXiv:2401.09176.

Li, Y., & Wu, J. (2020). Enhancing Answer Selection with Hypernym and Synset Information in WordNet. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7650-7660.

Lin, J., Zhang, Y., & Chen, H. (2024). Benchmarking Large Language Models on Consumer GPUs. arXiv preprint arXiv:2401.02669.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586.

Liu, T., et al. (2007). A new approach to word sense disambiguation based on context similarity. In Proceedings of the Workshop of the World Congress on Engineering.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

Longpre, S., Hou, Y., Tu, Z., Delangue, C., Phang, J., Wang, H., ... & Wang, A. (2023). The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arXiv preprint arXiv:2301.13688.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 8086-8098).

Mambrini, F., Passarotti, M., Litta, E., & Moretti, G. (2021). Interlinking Valency Frames and WordNet Synsets in the LiLa Knowledge Base of Linguistic Resources for Latin. In *Further with Knowledge Graphs* (Studies on the Semantic Web, Vol. 53, pp. 16-28). IOS Press.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. arXiv preprint arXiv:2005.00661.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation (Vol. 24, pp. 109-165). Academic Press.

McCrae, J. P., & Prangnawarat, N. (2016). Errors and Inconsistencies in the WordNet Ontology. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 16).

McGillivray, B., & Vatri, A. (2018). The Diorisis Ancient Greek Corpus and the Computational Analysis of Ancient Greek Poetry. In Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018).

Mehler, A., et al. (2020). The Frankfurt Latin Lexicon: From Morphological Expansion and Word Embeddings to SemioGraphs. arXiv:2005.10790. DOI: 10.48550/arXiv.2005.10790.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Wu, H. (2018). Mixed precision training. arXiv preprint arXiv:1710.03740.

Mihalcea, R., & Moldovan, D. (2001). A highly accurate bootstrapping algorithm for word sense disambiguation. International Journal on Artificial Intelligence Tools, 10(1-2), 5-21.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations (ICLR).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

Miller, G. A. (1995). WordNet: A Lexical Database for English. Communications of the ACM, 38(11), 39-41.

Miller, G. A., Leacock, C., Tengi, R., & Bunker, R. T. (1994). A Semantic Concordance. Proceedings of the Workshop on Human Language Technology, 303-308.

Miller, G., Beckwith, R., Fellbaum, C., et al. (1990). Introduction to WordNet: An on-line lexical database. International Journal of Lexicography, 3(4), 235-244.

Minozzi, S. (2009). The Latin WordNet Project. In P. Anreiter & M. Kienpointner (Eds.), Latin Linguistics Today: Akten des 15. Internationalen Kolloquiums zur Lateinischen Linguistik. Innsbrucker Beiträge zur Sprachwissenschaft, 137, 707-716.

Minozzi, S. (2017). Latin WordNet, una rete di conoscenza semantica per il latino e alcune ipotesi di utilizzo nel campo dell'Information Retrieval. In P. Mastandrea (Ed.), Strumenti digitali e collaborativi per le Scienze dell'Antichità (pp. 123-134). Antichistica. DOI: 10.14277/6969-182-9/ANT-14-10.

Mishra, S., Khashabi, D., Baral, C., & Choi, Y. (2022). ReframeQA: Reframing semantic parsing as a repair task. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 1829-1841). Association for Computational Linguistics.

Mistral AI. (2023). Mistral-7B: Pushing the Limits of Open-Source Language Models. Technical Report.

Morato, J., Marzal, M. Á., Lloréns, J., & Moreiro, J. (2004). WordNet applications. Proceedings of GWC 2004, 270-278.

Muennighoff, N., Margolis, A., Furman, N., Wolf, T., & Rush, A. M. (2023). BLOOM+1: Adding Language Support to the BLOOM Language Model. arXiv preprint arXiv:2301.11596.

NVIDIA Corporation. (2020). NVIDIA A100 Tensor Core GPU Architecture [White paper].

Navigli, R. (2009). Word Sense Disambiguation: A Survey. ACM Computing Surveys (CSUR), 41(2), 1-69.

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, 193, 217-250.

Navigli, R., Jurgens, D., & Vannella, D. (2013). SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In Second Joint Conference on Lexical and Computational Semantics (SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. Proceedings of the twenty-first international conference on Machine learning, 78.

Nickel, M., & Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. Proceedings of the 35th International Conference on Machine Learning, 3779-3788.

Nijkamp, E., Bajaj, P., Bavarian, M., Bhardwaj, S., Chen, B., Chin, C. R., ... & Sastry, G. (2023). CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. arXiv preprint arXiv:2203.13474.

Palmer, M., et al. (2010). Semantic role labeling. Synthesis Lectures on Human Language Technologies, 3(1), 1-103.

Park, J., Kim, S., & Lee, J. (2023). MedMistral: Advancing Medical Diagnosis through Specialized Language Model Fine-tuning. Medical Artificial Intelligence Conference.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity: Measuring the Relatedness of Concepts. Proceedings of the 19th National Conference on Artificial Intelligence, 1024-1025.

Peng, J., et al. (2020). BERT-EMD: Many-to-Many Layer Mapping for BERT Compression with Earth Mover's Distance. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 8290-8296.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532-1543.

Perez, E., Kiela, D., & Cho, K. (2021). True few-shot learning with language models. Advances in Neural Information Processing Systems, 34, 11054-11070.

Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019) (pp. 7-14).

Petroni, F., et al. (2019). Language Models as Knowledge Bases? Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2463-2473.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2020). AdapterFusion: Non-Destructive Task Composition for Transfer Learning. arXiv preprint arXiv:2005.00247.

Pilehvar, M. T., & Camacho-Collados, J. (2019). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Proceedings of NAACL-HLT 2019, 1267-1273.

Pinter, Y., & Eisenstein, J. (2018). Predicting Hypernym-Hyponym Relations with Distributional and Path-based Methods. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 1896-1901.

Piotrowski, M. (2012). Natural Language Processing for Historical Texts. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Porter, R., Zhu, H., & Deng, J. (2023). Memory-Efficient Fine-Tuning of Compressed Large Language Models via Sub-Network Synthesis. International Conference on Machine Learning.

Prechelt, L. (1998). Early stopping-but when?. In Neural Networks: Tricks of the trade (pp. 55-69). Springer, Berlin, Heidelberg.

Radford, A., et al. (2018). Improving language understanding by generative pre-training. OpenAI preprint.

Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI preprint.

Raffel, C., et al. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv preprint arXiv:1910.10683.

Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 99-110.

Rajbhandari, S., Li, D., & Ardalani, N. (2023). DeepSpeed Compression: A Key to Efficient Deep Learning Model Training and Inference. Microsoft Research Technical Report.

Ramasesh, V. V., Dyer, E., & Raghu, M. (2021). Anatomy of catastrophic forgetting: Hidden representations and task semantics. In International Conference on Learning Representations.

Renner, S., Denis, P., & Gilleron, R. (2023). An Effective Unsupervised Method for Graded Lexical Entailment Prediction using WordNet. Journal of Artificial Intelligence Research, 68, 56-78.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, 448-453.

Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-7). ACM.

Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. Journal of the American Society for Information Science, 27(3), 129-146.

Rodriguez, M., Garcia, A., & Lopez, R. (2024). LegalMistral: Enhancing Legal Document Analysis through Specialized Language Model Adaptation. Legal AI Journal.

Rospocher, M., et al. (2016). The KnowledgeStore: a storage framework for interlinking unstructured and structured knowledge. Web Semantics: Science, Services and Agents on the World Wide Web, 35, 85-99.

Rothe, S., & Schütze, H. (2015). AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536.

Rytting, C. A., & Wingate, D. (2021). To what extent can language models replace knowledge bases? Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, 64-77.

Sanchez, D., Batet, M., & Isern, D. (2012). Ontology-based information content computation. Knowledge-Based Systems, 24(2), 297-303.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 142–151). Dublin, Ireland.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilharco, G., & Rush, A. M. (2022). Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10, 50-72.

Schadd, F., & Ross, P. (2007). Including word-sense disambiguation techniques into lexical similarity metrics to disambiguate ontology concepts. Journal of Computer Science and Technology.

Schick, T., & Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 255-269).

Schick, T., Lauscher, A., Vulić, I., & Ponti, E. M. (2022). All NLI is ambiguous: But some NLI tasks are more ambiguous than others. arXiv preprint arXiv:2209.13161.

Schütze, H. (1998). Automatic word sense discrimination. Computational Linguistics, 24(1), 97-123.

Schütze, H. (1998). Automatic word sense discrimination. Computational Linguistics, 24(1), 97–123.

Sebti, K., & Barfroush, A. A. (2008). A new word sense similarity measure in WordNet. International Journal of Computer Science and Network Security, 8(7), 279-285.

Shah, R., Al-Shedivat, M., Carbonell, J., & Gu, A. (2022). On the pitfalls of goal misgeneralization in learning diverse action sequences. arXiv preprint arXiv:2206.01222.

Short, W. M. (2024). Latin WordNet Project. University of Exeter. Retrieved from Latin WordNet.

Short, W. M., Fedriani, C., & De Felice, I. (2020). The Digital Lexicon Translaticium Latinum: Theoretical and Methodological Issues. In Proceedings of the International Conference on Computational Semantics (IWCS).

Shukla, R., et al. (2021). A Survey of Query Expansion Approaches for Information Retrieval. Journal of Information Science, 47(2), 128-155.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

Su, J., Lu, Y., Pan, S., Murtadha, A., & Wen, B. (2022). RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv preprint arXiv:2104.09864.

Taghipour, K., & Ng, H. T. (2015). One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction. Proceedings of the 19th Conference on Computational Natural Language Learning, 338-344.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems.

Vaswani, A., et al. (2017). Attention is All you Need. Advances in Neural Information Processing Systems, 30, 5998-6008.

Vossen, P. (Ed.). (1998). EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., & Korhonen, A. (2020). Probing pretrained language models for lexical semantics. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7222-7240).

Wang, R., Tang, H., Lin, J., & Chen, H. (2023). Enhancing Code Generation through Specialized Fine-tuning of Language Models. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824-24837.

Winata, G. I., Cahyawijaya, S., Lin, Z., Liu, Z., Xu, P., & Fung, P. (2021). Meta-Transfer Learning for Code-Switched Speech Recognition. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics.

Winters, M., Tissari, H., & Allan, K. (Eds.). (2010). Historical cognitive linguistics. Berlin: DeGruyter.

Wu, Z., & Palmer, M. (1994). Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

Xu, C., Zhou, W., Ge, T., Wei, F., & Zhou, M. (2021). BERT-of-Theseus: Compressing BERT by progressive module replacing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 6719-6733).

Xu, H., Liu, B., Shu, L., & Yu, P. S. (2023). Uncovering Sensitive Attributes of GPT-3.5 and LLaMA Models: A Comprehensive Analysis. arXiv preprint arXiv:2305.18396.

Yang, Z., et al. (2020). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Advances in Neural Information Processing Systems, 32, 5754-5764.

Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Mahoney, M. W., & Keutzer, K. (2022). A Comprehensive Survey on Compression and Acceleration of Deep Neural Networks. Proceedings of the IEEE.

Yih, W.-T., et al. (2013). Question Answering with Semantic Enrichment Using Wikipedia Ties. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 677-682.

Yu, M., Dredze, M., & Ji, H. (2019). Improving Word Sense Disambiguation with Neural Network Knowledge Graph Embeddings. Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI-19), 3126-3133.

Zaken, E. B., Ravfogel, S., & Goldberg, Y. (2022). BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 1–9).

Zanchi, C., Luraghi, S., & Biagetti, E. (2021). Linking the Ancient Greek WordNet to the Homeric Dependency Lexicon. In Proceedings of the 11th Global WordNet Conference, 258-266. DOI: 10.28995/2075-7182-2021-20-729-737.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), 107-115.

Zhang, C., Öztürk, P., & Li, S. (2022). Unified Compression-Compilation for Compact and Efficient Language Models. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Zhang, J., Kamath, U., & Mahoney, M. W. (2023). Quantization of Large Language Models: Methods and Analysis. arXiv preprint arXiv:2308.11107.

Zhang, T., Wu, F., Katsis, Y., & Dasgupta, A. (2023). Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. Transactions of the Association for Computational Linguistics, 11, 163-177.

Zhang, T., et al. (2020). Revisiting Few-sample BERT Fine-tuning. arXiv preprint arXiv:2006.05987.

Zhang, Y., Li, X., Wang, T., & Chen, H. (2024). Efficient Sentiment Analysis Fine-tuning of Large Language Models. Journal of Artificial Intelligence Research.

Zheng, S., Xu, F., Pavlick, E., & Auli, M. (2023). Effective Long-Context Scaling of Foundation Models. arXiv preprint arXiv:2309.16039.

Zhong, Z., & Ng, H. T. (2010). It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 78-83.

Zhou, Z., Wang, Y., & Gu, J. (2008). New model of semantic similarity measuring in WordNet. Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 256-261.

Zhu, M., Bernhard, D., & Gurevych, I. (2016). A monolingual tree-based translation model for sentence simplification. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 431-441.

Zhu, Y., Chen, Z., & Dai, H. (2017). Measuring semantic similarity by combining word embeddings and knowledge graphs. Neurocomputing, 254, 34-44.

Zhu, Y., et al. (2020). Aligning LLMs with human values via iterative deployment. arXiv preprint arXiv:2011.00410.

Ringraziamenti

Desidero esprimere i miei più sinceri ringraziamenti alla Prof.ssa Chiara Zanchi, relatrice di questa tesi, per avermi guidato nella scelta del percorso di ricerca e per avermi dato fiducia coinvolgendomi nel progetto *Linked WordNets for Ancient Indo-European Languages* e nei suoi nuovi sviluppi, senza i quali questo esperimento non sarebbe mai nato. Il suo supporto costante e i suoi preziosi consigli sono stati fondamentali non solo durante la fase sperimentale ma anche nella stesura di questo elaborato.

Un sentito ringraziamento va alla Prof.ssa Claudia Roberta Combei, correlatrice di questa tesi, che mi ha introdotto al mondo dei Large Language Models e che, con il suo supporto, mi ha aiutato a trovare la mia strada sia in ambito accademico che professionale.

Ringrazio Marco Del Tredici per la preziosa supervisione tecnica del progetto e per la pazienza dimostrata nell'organizzazione del lavoro. Un ringraziamento speciale va a Beatrice Marchesi, collega e compagna-annotatrice, con la quale ho condiviso spunti preziosi che hanno contribuito significativamente alla qualità dei risultati finali e della stesura di questo lavoro.

Desidero inoltre ringraziare tutto il team del progetto, in particolare Eleonora Litta e Riccardo Ginevra per gli importanti insegnamenti sul Latin WordNet, così come Erica Biagetti, Silvia Zampetta, Tullio Facchinetti e Stefano Rocchi per il loro costante supporto e le loro puntuali osservazioni durante tutte le fasi dell'esperimento.

Un ringraziamento particolare va alla Prof.ssa Ilaria Fiorentini che, pur non avendo preso parte alla stesura di questa tesi, è stata una figura fondamentale nel mio percorso accademico, riaccendendo in me quella passione per lo studio che avevo smarrito.

Infine, un ringraziamento affettuoso va a Fabio, il mio gatto, silente co-autore di questo lavoro.