UNIVERSITÀ DI PISA

DIPARTIMENTO DI

FILOLOGIA, LETTERATURA E LINGUISTICA

Corso di Laurea Magistrale in Informatica Umanistica

TESI DI LAUREA

Studio dello spostamento semantico nell'evoluzione storica della lingua italiana tramite modelli di linguaggio neuronali

CANDIDATO
Eva Sassolini

RELATORE
Prof. Giuseppe Attardi

CONTRORELATORE
Prof. Marco Maggiore

ANNO ACCADEMICO 2021/2022

Indice

Capitolo 1

Introduzione

Capire come il tempo agisce sui cambiamenti che avvengono nel linguaggio vuol dire riuscire a comprendere le caratteristiche dei cambiamenti nel significato e nell'uso delle parole. La lingua è in continuo mutamento e le cause riguardano sia fattori esterni come cambiamenti culturali, sociali e tecnologici, ma anche motivazioni più profonde solo parzialmente comprensibili. La storia delle parole può essere anche molto articolata. Parole che acquisiscono nuovi significati e ne perdono di vecchi, nuove parole nascono dalle esigenze espressive più varie, spesso prese in prestito da altre lingue, mentre altre diventano obsolete e se ne perde l'uso definitivamente. In anni recenti la comunità scientifica ha adottato nuovi metodi e strumenti computazionali per supportare l'indagine sul cambiamento di significato diacronico. La linguistica diacronica comprende approcci computazionali in grado di indagare in modo strutturale il fenomeno ed è a questi che il lavoro di tesi vuole rifarsi, con l'intenzione di analizzare i metodi computazionali adatti a trattare il rilevamento del significato delle parole nel tempo, con particolare riguardo al trattamento di varietà storiche della lingua italiana. Si è intrapreso uno studio che mirasse a comprendere quali sono le più recenti tecniche informatiche d'indagine, quali fossero i modelli di riferimento in letteratura per questo compito, quali gli strumenti più efficaci, come si implementassero e con quali risultati. Il lavoro cerca di fare proprie le indicazioni degli autori riportate al termine di '*Survey of computational approaches to lexical semantic change detection*' di Tahmasebi et al. (2018), che si possono riassumere in pochi punti

essenziali: (i) Mostrare e discutere i risultati, fornire il proprio punto di vista e giustificazione, spiegare perché risultati si ritengono corretti o no. (ii) Utilizzare sempre un controllo sulle sperimentazioni cercando dove possibile una controprova, lavorare su parole stabili nel tempo o quando è possibile su set di dati di controllo, poiché se isolati i numeri non sono sufficienti a giustificare le scelte. (iii) Provare a rilevare automaticamente il cambiamento semantico, anche se ciò non è ancora del tutto possibile.

Sarà descritto il lavoro fatto per costruire un metodo valido per lo studio del cambiamento di significato in diacronia, basato sui modelli di linguaggio di tipo BERT (*Bidirectional Encoder Representations from Transformers*) di Devlin et al. (2019). Saranno attraversate tutte le fasi di lavoro, dallo studio dello stato dell'arte (§ 3), alla costruzione di corpora diacronici ILC-Ita (§ 5), proseguendo con lo studio della più opportuna configurazione del modello BERT per l'italiano (§ 6). Sarà quindi descritto come si è arrivati alla scelta del metodo (§ 6) e tutte le sperimentazioni condotte: l'addestramento dei modelli (§ 6.1.3); l'estrazione degli embeddings (§ 6.2); i vari metodi sperimentati: il metodo senza aggregazioni (§ 6.3) , il metodo con aggregazioni (clustering), (§ 6.4); le prove sul task DIACR-Ita corredate dai rispettivi risultati (§ 7) e la sperimentazione sui corpora ILC-Ita realizzati (§ 7.4); le analisi finali e le valutazioni conclusive comprenderanno sia le analisi sul clustering, le analisi di tutti i risultati e infine la valutazione del metodo (§ 8); le conclusioni chiuderanno la tesi (§ 9).

Capitolo 2

Obiettivi del progetto

Il progetto di tesi è stato concepito come un approccio operativo all'uso di modelli neuronali di rappresentazione del linguaggio per varietà storiche della lingua italiana, uno studio mirato alla costruzione di un metodo computazionale in grado di trattare lo spostamento di significato in diacronia. In particolare si intende dimostrare che gli embedding contestuali (*contextualized word embeddings)*, analizzati come vettori dello spazio da essi generato o mediante algoritmi di clustering, sono un metodo efficace per l'analisi del significato delle parole in diacronia. Per confutare questa ipotesi si sono valutati in letteratura i modelli che li producono e quali risposte sono in grado di offrire per lo studio del fenomeno. Tutte le prove fatte nelle sperimentazioni condotte indicano che l'ipotesi è corretta. Le argomentazioni che la confermano sono diverse, tutte verificate sperimentalmente: (i) lo spazio degli embeddings contestuali creati con modelli BERT permette di rappresentare la polisemia; (ii) la possibilità di specializzare i modelli su varietà particolari della lingua è possibile e ottiene buoni risultati; (iii) i metodi di indagine sugli embeddings contestuali sono adeguati e riguardano sia metriche di distanza che agiscono direttamente sullo spazio degli embeddings, sia l'applicazione di algoritmi di clustering di vario tipo; (iv) è inoltre possibile una flessibilità dell'indagine rispetto al dato trattato: dimensioni, studio del significato vs studio dell'uso.

Tutte queste condizioni rendono l'approccio non solo possibile ma anche efficace e generalizzabile. La letteratura recente ha indicato il clustering come lo strumento più adatto per uno studio di questo tipo, ma non lo ha dimostrato definitivamente. Questa tesi prova a dare una risposta, cercando di fare chiarezza su questioni sulle quali molti studi si interrogano: le parole di significato simile sono effettivamente vicine nello spazio degli embeddings? Come si configura l'operazione di clustering rispetto alla rappresentazione dei sensi? Ovvero, la distanza tra cluster indica necessariamente che le parole che appartengono ad uno, siano 'distanti' da quelle dell'altro? Intuitivamente si potrebbe pensare che tutte le parole semanticamente simili siano aggregate in uno stesso cluster, ma la prova nei fatti dimostra che non è così. Questo vuol dire che il clustering non funziona come dovrebbe?

Il lavoro svolto e il contesto sperimentale proposto in questa tesi dimostrano che lo strumento è corretto, ma come tale va considerato, ossia un metodo computazionale che riesce a trattare strutture complesse come gli embeddings meglio di altri metodi, ma con logiche proprie, di cui occorre tenere conto. Per esempio il numero di cluster prodotti per una parola non è in diretto rapporto con il numero di sensi/usi, dipende dall'algoritmo di clustering e dai dati sui quali lavora. Quello che si può dimostrare è che, se configurato correttamente, se ne può controllare il funzionamento: ai fini dell'indagine non è importante che un algoritmo generi un numero corrispondente di cluster per senso, ma è determinante capire quali sono gli elementi chiave che aiutano a valutare il clustering prodotto per ogni algoritmo. Per esempio esistono algoritmi che generano almeno due cluster per impostazione predefinita, anche in assenza di un nuovo senso per una parola; mentre altri possono non produrne affatto, questo non inficia l'indagine, basta conoscere la chiave di lettura di ognuno di questi algoritmi. Questo è molto importante per poter confrontare i risultati, che altrimenti sarebbero ambigui.

Appurato che lo strumento del clustering conferma la sua idoneità per lo studio del cambiamento semantico in archi temporali diversi. È necessario

spostare l'attenzione dallo strumento di indagine all'oggetto della stessa, ovvero: cosa realmente aggregano questi strumenti? L'embedding di una parola è estratto dalla frase in cui si trova e questa può essere composta di parole distribuite nei modi più vari. Quanto contribuisce il conteso della frase al valore che viene prodotto per quella parola? Quali sono le conoscenze linguistiche che il modello addestrato utilizza nella produzione degli embeddings? Tali interrogativi mettono in luce uno stretto legame tra modello di rete che si implementa e dato di input e confermano l'estrema sensibilità di queste rappresentazioni. Questo stretto legame è emerso da tutte le sperimentazioni condotte e la capacità di comprenderlo e affrontarlo rappresenta il contributo sperimentale maggiore che questa tesi possa offrire. I capitoli 7 e 8 mostreranno con l'evidenza dei dati quale complessità sta dietro queste rappresentazioni e come possa essere approfondito lo studio interpretativo da condurre su di esse.

Il compito sul quale è stato implementato il metodo viene anche definito *detecting meaning change* [Azarbonyad et al., 2017; Del Tredici et al., 2019]. In realtà quando si considerano intervalli di tempo relativamente brevi, è più corretto parlare di 'cambiamento d'uso' piuttosto che di 'cambiamento di significato' di una parola, come proposto da [Gonen et al. 2020]. L'autore infatti sostiene che le parole possono avere gli stessi significati in vari corpora, ma diverso senso dominante in ciascuno di essi, corrispondente solo ad un uso diverso della parola. Per questo motivo, afferma che sia più corretto riferirsi a questa evoluzione come 'rilevamento del cambiamento d'utilizzo'. Le sperimentazioni fatte nel lavoro di tesi confermano questa opinione e suggeriscono di adottare la stessa interpretazione in contesti di studio simili.

La letteratura specifica identifica cinque componenti essenziali per gli studi in questo campo: corpus diacronico; caratterizzazione diacronica del senso delle parole; modellazione del cambiamento; dati di valutazione e tecniche di visualizzazione dei dati [Tang, X. 2018]. Il progetto di tesi ha provato a toccare tutti i punti cercando di entrare nel merito delle scelte fatte, valutando sempre le alternative. Si è partiti dalla ricerca e/o creazione di

corpora testuali sincronici di addestramento relativi a epoche diverse; si è proseguito con la scelta degli strumenti più adatti allo studio del cambiamento semantico, in particolare utilizzando il modello BERT da addestrare sui corpora creati. A partire dal modello individuato si è continuato con lo studio e l'implementazione di strategie per l'estrazione e l'aggregazione delle rappresentazioni contestualizzate delle parole.

Per proporre un metodo in grado di condurre uno studio sistematico del fenomeno, oltre a tutte le motivazioni teoriche, occorre tenere presente anche aspetti di accuratezza e interpretabilità, che riguardano la qualità e precisione del metodo, così come quelli di stabilità e scalabilità, che attengono a caratteristiche quantitative di applicabilità dello stesso.

Per una più rigorosa valutazione del metodo si è ricorsi alla letteratura di settore, adottando i criteri più consolidati oggi disponibili. Con questa intenzione si è studiato quanto proposto prima in SemEval[1] 2020 nel task LSC (*Lexical Semantic Change*), in particolare nel task 1: '*Unsupervised Lexical Semantic Change Detection*' in cui vennero introdotte per la prima volta due attività[2] correlate, che miravano a identificare il cambiamento di significato delle parole nel tempo, utilizzando dati di un corpus. La competizione ha reso possibile confrontare vari modelli e tipi di rappresentazioni semantiche come: rappresentazioni dense di parole (contextualized word embeddings); immersioni di tipi; modelli di argomenti rispetto a modelli di spazi vettoriali e metriche per la misura del cambiamento semantico. In particolare, era attinente agli scopi di questo studio il sotto-task 1.1, ossia quello riconducibile a una classificazione binaria, utile a decidere quali parole avessero perso o acquisito significati nell'arco temporale che caratterizzava i due corpora diacronici forniti. Lo studio dei vari approcci e delle strategie utilizzate ha

[1] SemEval (Semantic Evaluation) è un'iniziativa internazionale, nata come evoluzione di Senseval e configurato come una serie di workshop di ricerca internazionali sull'elaborazione del linguaggio naturale. Gli obiettivi dell'iniziativa riguardano il progresso della ricerca nell'analisi semantica e l'aiuto nella creazione di set di dati annotati di alta qualità, utilizzabili per una vasta gamma di problemi attinenti la semantica del linguaggio naturale.

[2] Il sotto-task 1.2 per la rilevazione del grado di cambio di senso è stato proposto per la prima volta in SemEval 2020.

permesso di costruire i metodi per la sperimentazione e il confronto dei risultati.

Le lingue utilizzate nel task LSC non comprendevano l'italiano, ma una formulazione simile, questa volta specificamente per l'italiano, è stata proposta nel task DIACR-Ita[3] di EVALITA[4] 2020. È stato quindi possibile applicare le tecniche implementate ai dati forniti dagli organizzatori di questa competizione e avvalersi del *gold standard*[5] per la valutazione dei risultati. La valutazione effettuata ha permesso un'analisi puntuale dell'efficacia delle tecniche di indagine adottate. Con l'intento di sfruttare al meglio l'opportunità di avere un riscontro sui risultati prodotti, si sono messi a confronto diversi approcci e, ove possibile, si sono analizzati i punti critici affrontati nella modellazione del metodo e le possibili prospettive.

[3] https://diacr-ita.github.io/DIACR-Ita/
DIACR-Ita è il primo task sul cambiamento semantico lessicale per l'italiano, che unisce linguistica computazionale e storica. Il task mette in competizione i partecipanti nello sviluppare sistemi in grado di rilevare automaticamente se una determinata parola ha cambiato significato nel tempo, date le informazioni contestuali presenti nei corpora.
[4] https://www.evalita.it/
[5] Dati corretti, annotati manualmente da esperti umani

Capitolo 3

Il cambiamento di significato

La lingua è in costante cambiamento e comprendere tale fenomeno è un compito complesso. Per il glottologo Bréal[6], ritenuto il fondatore della semantica moderna, l'unica motivazione dello sviluppo delle lingue è la volontà umana di comunicare, che opera come un'energia interna alla storia stessa delle lingue. L'interesse principale di Bréal era infatti la ricostruzione della storia dell'evoluzione del linguaggio. Lo spostamento semantico o cambiamento di significato può avvenire a diversi livelli linguistici: fonologico, morfologico, sintattico e semantico, ragione per cui il significato della parola può essere oggetto di diversi tipi di cambiamento: cambiamento di polarità, cioè spostamento di significato da positivo a negativo (peggioramento) o passaggio da significato negativo a positivo (miglioramento); generalizzazione e specializzazione si riferiscono invece a un cambiamento di significato nella tassonomia lessicale e mentre il primo ha significato di 'allargamento', l'altro ne indica invece un restringimento. Alcune delle spinte che provocano questi spostamenti si possono identificare, per esempio la perdita di significato semantico: quando una parola perde il suo senso originario trasformandosi in un elemento grammaticale (es. 'mica', originariamente *briciola* oppure 'tranne' in origine una forma di imperativo: *tràine* 'togline'). Altri spostamenti di significato si ritrovano quando una parola da aggettivo diventa sostantivo, per esempio nelle parole: *cellulare, mediano*, *abbattitore*, *acceleratore*, ecc. Esistono anche i prestiti e le estensioni, che sono forme di penetrazione da una lingua all'altra, in genere dovuti a contatti di tipo culturale, per esempio il

[6] Michel Bréal, (Essai de Sémantique, science des significations. Paris 1921).

verbo realizzare significa “rendere reale”, che acquista il significato di “rendersi conto” come traduzione dall’inglese *realize*. I calchi sono invece parole che entrano in una lingua come traduzioni integrali di parole di un’altra: alcuni esempi sono ‘pallacanestro’ replica l’inglese *basket-ball*, oppure sempre dall’inglese ‘grattacielo’ è traduzione di *skyscraper*, ma anche ‘superuomo’ dal tedesco *Übermensch*.

Il linguista e glottologo Lazzeroni (1990) presenta una classificazione di modelli di mutamento semantico basandosi sui criteri tracciati da Meillet[7] e sui risultati della classificazione di Ullmann[8]. Quest’ultimo infatti sostiene che il significato di una parola è definito come la relazione reciproca e reversibile tra nome e senso; la relazione viene definita reciproca e reversibile per il fatto che chi ascolta una parola, pensa alla cosa, così come chi pronuncia una parola, pensa alla cosa.

Quando il cambiamento di senso si analizza in diacronia occorre precisare che normalmente si stabilisce una distinzione tra studio delle lingue a livello sincronico (ovvero, in un determinato momento storico, a prescindere dai fenomeni di mutamento verificatisi sino a quel momento) e a livello diacronico (ovvero, studio del mutamento linguistico). In realtà la lingua non è mai veramente statica perché i parlanti possono continuamente interpretare determinate costruzioni in maniera diversa in contesti diversi, dando avvio a mutamenti continui. Tanto da far ipotizzare che la dicotomia tra sincronia e diacronia sia in realtà artificiale. Sempre Lazzeroni (2015) sostiene che il mutamento linguistico si manifesta in momenti in cui il sistema linguistico cambia e si riorganizza (il linguista li definisce ‘momenti di crisi’), si innesca quindi un processo graduale in cui il vecchio convive col nuovo; suggerendo come la prospettiva dinamica sia la più adatta a svelare i principi che governano tali sistemi. Queste ricerche mostrano come la variazione e il mutamento procedano non solo lungo le dimensioni del tempo e dello spazio,

[7] A. Meillet (1866 – 1936), linguista francese, studente di M. Bréal e di F. de Saussure, a cui si deve la prima identificazione del fenomeno della ‘grammaticalizzazione’.
[8] S. Ullmann (1914 – 1976), linguista e filologo ungherese (The Principles of Semantics - 1951).

ma anche lungo quella della profondità socioculturale (e stratigrafica) della comunità dei parlanti. Il mutamento linguistico ha come campo di studio naturale le lingue "vive", che consentono di osservarlo in atto, ma è altrettanto vero che se si riesce ad applicare alle lingue del passato le prospettive e i metodi perfezionati nello studio di quelle del presente, allora, come ha scritto Lazzeroni: "*avremo raggiunto l'altro versante dell'insegnamento di W. Labov: capire il presente spiegando il passato*".

3.1 Approcci computazionali allo studio del fenomeno

Gli studi del cambiamento semantico dal punto di vista computazionale hanno dimostrato empiricamente l'esistenza di una divisione tra derive linguistiche (cambiamenti lenti e regolari nel significato fondamentale delle parole) e cambiamenti che potremmo definire culturali o di utilizzo (cambiamenti nelle associazioni di una data parola determinati culturalmente) [Hamilton et al., 2016a]. Nella classificazione tradizionale di Stern (1931) invece la categoria di sostituzione dello spostamento semantico descrive un cambiamento che non ha una causa linguistica, ma è dovuta al progresso tecnologico, un esempio è la parola 'auto' che ha spostato il suo significato a veicoli motorizzati.

Questo tipo di studi, rappresentano un argomento tradizionale in linguistica e hanno richiesto storicamente molto lavoro manuale: le analisi venivano solitamente eseguite su scala molto ridotta. Con la crescente quantità di testi digitalizzati che coprono un ampio periodo storico, nell'ultimo decennio i campi della linguistica computazionale e dell'informatica hanno iniziato a interessarsi alla diacronia. Piuttosto che studiare effettivamente il cambiamento semantico lessicale, la maggior parte dei lavori propone metodi e modelli per rilevarlo in modo automatico e affidabile. Hamilton già nel 2016 [Hamilton et al., 2016b] sosteneva come le immersioni di parole si mostrassero promettenti per comprendere come le parole cambiano il loro significato nel tempo, ma sosteneva anche come i dati storici sul significato delle parole fossero scarsi e rendessero le teorie difficili da sviluppare e verificare.

Gran parte dei sistemi e i metodi utilizzati in linguistica computazionale per il compito di rilevamento del cambiamento di significato, si basano sull'ipotesi distributiva (Harris, 1954) che afferma che parole semanticamente simili tendono a comparire in contesti linguistici simili. Questo assunto ha un'applicazione immediata nell'estrazione delle proprietà semantiche di una parola da corpora testuali. Storicamente, i primi approcci alla modellazione diacronica erano basati sulle frequenze relative delle parole e sulle somiglianze distributive (Hilpert, 2006). L'uso di sistemi basati sull'immersione (*embedding*) di parole è più recente ma ha rivestito grande interesse negli ultimi anni, con la pubblicazione di numerosi articoli dedicati e una rassegna della letteratura di settore [Kutuzov, Øvrelid, Szymanski, & Velldal, 2018; Tahmasebi et al., 2018; Tang, 2018].

L'adozione di tecniche di linguistica computazionale dello studio del significato delle parole in diacronia, ha comportato una diversificazione in relazione alla comunità scientifica che lo affronta o del periodo temporale indagato. Il campo di studio si è allargato a questioni di cambiamento semantico; cambio di lingua; spostamento diacronico; deriva semantica; cambiamento concettuale diacronico e quasi tutte le possibili combinazioni di queste parole. La letteratura riporta diversi compiti e terminologia diversa: Tahmasebi et al (2018) definiscono il campo di ricerca come 'rilevamento del cambiamento semantico lessicale'; definizione seguita anche da Schlechtweg et al. (2020) e Basile et al. (2020). Kutuzov et al. (2018), invece, formalizzano il compito come 'rilevamento dei cambiamenti semantici'. Del Tredici, et al. (2019) usano una definizione più ristretta: 'cambiamento di significato a breve termine'. Sempre partendo dalla stessa esigenza di specificare il contesto ristretto, Gonen et al. (2020) hanno definito il compito come 'rilevamento del cambiamento di utilizzo'.

Nel lavoro di tesi, parlando delle teorie e degli approcci dei vari studi passati in rassegna si è spesso utilizzato il termine più neutro e generale di LSC, riservando poi alle sperimentazioni sui dati una classificazione più specifica. Come già indicato in precedenza (§ 2), quando si descrive una prospettiva

ampia del cambiamento di significato, la distinzione proposta da Gonen non è necessaria, ma si vedrà che in casi come quello dei corpora dell'italiano, l'interpretazione risponde meglio alle caratteristiche veramente indagate.

Lo sviluppo di nuove metodologie per lo studio dei ruoli semantici lessicali nella linguistica generale [Traugott, 2017] è strettamente legato alla maggiore disponibilità di ampi corpora. Se ne sono giovati anche i lavori sulla semantica dell'uso. Un presupposto chiave in gran parte di questi lavori è che i cambiamenti nei modelli/schemi collocazionali di una parola riflettono i cambiamenti nel significato della parola [Hilpert, 2008], fornendo così una causa/ragione della semantica basata sull'uso [Gries, 1999]. Questa visione si allinea bene con le ipotesi alla base dell'approccio semantico distributivo [Firth, 1957], spesso impiegato nelle attività di NLP (*Natural Language Processing*).

Sono molte le recenti pubblicazioni volte a tracciare i cambiamenti temporali nella semantica lessicale utilizzando metodi distributivi, in particolare modelli di *word embeddings* (immersioni di parole) basati sulla previsione. Lo stato attuale della ricerca scientifica relativa alle immersioni diacroniche di parole e al rilevamento degli spostamenti semantici è molto variegato e ha prodotto metodi diversi che possono essere confrontati per individuare le principali sfide di questo sottocampo emergente [Tahmasebi et al., 2019]. Lo sviluppo della semantica computazionale ha dato origine a una serie di iniziative di ricerca che cercano di catturare i cambiamenti semantici diacronici in modo guidato dai dati. Da tempo i word embeddings sono una rappresentazione di input ampiamente utilizzata per questo compito. Le ricerche più attuali studiano i cambiamenti semantici usando modelli distributivi di word embeddings, dunque rappresentando il significato lessicale con vettori densi prodotti da dati di co-occorrenza.

Studi recenti [Giulianelli et al., 2020; Martinc et al., 2020a] mostrano che il raggruppamento (più comunemente definito *clustering*) di immersioni contestuali potrebbe essere un *proxy* per il cambiamento dell'uso delle parole. In generale il clustering è una tecnica di apprendimento non supervisionato

che esplora i dati alla ricerca di possibili relazioni e raggruppamenti o cluster. In questo specifico contesto possiamo pensarlo come un metodo che cerca di raggruppare parole secondo criteri di somiglianza (di significato/uso). Se questi raggruppamenti, che in teoria catturano usi distinti delle parole, producono cluster nei quali le parole si distribuiscono diversamente in periodi di tempo differenti, è ragionevole dedurne un possibile cambiamento del 'contesto' della parola o addirittura la perdita di un senso o il segnale di uno nuovo.

Tutti questi studi affermano come l'approccio basato sul clustering offra una interpretazione più intuitiva del cambiamento nell'uso delle parole rispetto ai metodi alternativi. Per esempio quelli che, per interpretare il cambiamento, esaminano i 'vicini' di una parola in ogni periodo di tempo, trascurando il fatto che una parola possa avere più di un significato. Il principale limite dei metodi basati sul clustering è la scalabilità in termini di consumo di memoria e tempo di elaborazione: il clustering viene applicato a ciascuna rappresentazione della parola separatamente e tutte le sue occorrenze nel corpus devono essere aggregate in cluster. In grandi corpora con ampi vocabolari alcune parole possono apparire milioni di volte e l'uso di questi metodi ne risulta fortemente limitato. In questo contesto internazionale di studi riveste un interesse particolare capire se e come i nuovi modelli di linguaggio BERT possono offrire una valida alternativa a tecniche popolari comeWord2Vec[9], GloVe[10], ELMo (§ 4.1), FastText[11], per l'immersione di parole. Tali metodi producono embeddings calcolati come la media di tutte le immersioni di ogni parola indagata, che si configurano come i vettori dei diversi usi osservati delle parole nell'intero corpus. Fondamentalmente, un

[9] Mikolov, Chen, Corrado e Dean (2013) hanno proposto il framework Word2Vec con due algoritmi, Continuous Bag of Words (CBOW) e Skip-Gram.

[10] Pennington, Socher e Manning (2014) propongono Glove, un sistema che si basa sulla fattorizzazione della matrice di co-occorrenza parola-contesto per costruire immersioni di parole. Invece di prendere le probabilità di co-occorrenza grezze per codificare il significato delle parole, usano il rapporto delle probabilità di co-occorrenza di tre parole.

[11] Bojanowski, Grave, Joulin e Mikolov (2017) rilasciano FastText, un algoritmo che supera il problema delle parole fuori vocabolario ricorrendo a sotto-parole come componente minimo, anziché parole complete, per costruire immersioni.

word embedding non solo converte la parola in una rappresentazione opportuna, ma identifica anche la semantica e la sintassi della parola per costruire una rappresentazione vettoriale di queste informazioni. Tuttavia gli embeddings prodotti con sistemi tipo Word2Vec non sono sensibili al contesto e quindi incapaci di catturare direttamente la polisemia. Il motivo risiede nella necessità per tali sistemi di rappresentare una parola con un singolo vettore di immersione, come media di tutti quelli estratti per quella data parola. Questa rappresentazione univoca e statica risulta essere un limite nel caso di studi diacronici, poiché spesso il cambiamento del contesto di una parola rispecchia un cambiamento nel suo significato o uso. È più realistico addestrare immersioni di parole che possano seguire i cambiamenti del loro utilizzo in un corpus. Le innovazioni prodotte dalle ultime ricerche hanno cambiato le tecniche di word embedding dotandole della capacità di 'valutare il contesto' della parola, ovvero attraverso l'utilizzo delle informazioni dalle parole adiacenti ad essa. Queste promettenti innovazioni hanno portato alla decisione di adottarli nel lavoro di tesi.

Esistono diversi metodi per addestrare tali immersioni, che possono essere suddivisi in apprendimento indipendente delle immersioni per ogni intervallo di tempo e strategie di apprendimento dipendente, che utilizzano le informazioni dell'intero periodo di studio. Dallo studio della letteratura di riferimento, pur avendo questi modelli raggiunto prestazioni e risultati allo stato dell'arte in quasi tutti i campi di NLP, tuttavia nello specifico contesto del cambiamento semantico sembrano avere un'efficacia minore, rispetto alle tecniche più consolidate sopra indicate. Le ragioni possono essere molte e si è ritenuto importante indagarle per capire la natura del problema. Dalle sperimentazioni condotte non sono stati individuati indizi di carenze strutturali, è più probabile che ad oggi i modelli risentano ancora di una limitata esperienza di utilizzo in questo tipo di analisi.

Capitolo 4

Rassegna dello stato dell'arte

La letteratura di riferimento mostra che lo studio del cambiamento semantico con strumenti informatici è molto attivo sulla lingua inglese, ma fa fatica a diffondersi in altre lingue. Nonostante le potenzialità del campo, la sua analisi presenta svariati studi orientati alla confutazione di ipotesi proposte in linguistica teorica, mentre gli approcci più operativi risentono di questioni fondamentali ancora non risolte: la necessità di corpora diacronici per lingue diverse dall'inglese; la necessità di dati annotati per la valutazione; il confronto e la costruzione di approcci mirati alla caratterizzazione diacronica del senso delle parole e alla modellazione del cambiamento. Grande importanza riveste anche l'analisi sistematica delle tecniche di visualizzazione dei dati per la giustificazione delle ipotesi [Tang, 2018]. Gli articoli più recenti che indagano i cambiamenti nella semantica lessicale in diacronia utilizzano principalmente metodi distributivi, in particolare modelli di word embeddings basati sulla previsione, ovvero prevedere la parola corrente dato il contesto, oppure prevedere le parole di contesto dalla parola corrente. Questo tipo di rappresentazioni sono diventate un input ampiamente utilizzato per questo compito, le troviamo in molti articoli successivi al 2011 [Mikolov et al., 2013].

Il campo di studi è altamente eterogeneo, si trovano almeno tre diverse comunità di ricerca interessate: elaborazione del linguaggio naturale (e linguistica computazionale), information retrieval (e informatica in generale) e scienze politiche. Per gli obiettivi di questo lavoro l'indagine è ristretta a

quelle ricerche che tracciano i cambiamenti semantici usando modelli distributivi di word embeddings e che rappresentano il significato lessicale con vettori densi prodotti da dati di co-occorrenza. Prima dell'ampia adozione di tali modelli per tracciare spostamenti semantici o altri tipi di cambiamento linguistico era abbastanza comune utilizzare il cambiamento nelle frequenze delle parole nel corpus, per esempio Hilpert e Gries (2009), o Michel et al. (2011), o più recentemente Lijffijt et al. (2012). Il cambiamento della frequenza di una parola però non è necessariamente indicazione di cambiamento semantico. Un approccio in grado di modellare più direttamente il significato delle parole risulta superiore ai metodi basati sulla frequenza. Numerose pubblicazioni recenti hanno mostrato che le rappresentazioni di parole attraverso vettori sparsi o densi (embeddings) forniscono un modo efficiente per svolgere task come LSC [Turney et al., 2010; Baroni et al., 2014]. In effetti Kulkarni et al., (2015) hanno dimostrato esplicitamente che i modelli distributivi superano i metodi basati sulla frequenza nel rilevare gli spostamenti semantici.

Jurgens e Stevens (2009)[12] hanno utilizzato l'algoritmo *Random Indexing* (RI)[13] per creare vettori di parole. Sempre negli stessi anni altri hanno utilizzato modelli espliciti basati sul conteggio, costituiti da matrici sparse di co-occorrenza pesate da *Local Mutual Information* [Gulordava et al., 2011], mentre Sagi et al. (2011) si sono rivolti al *Latent Semantic Analysis*[14]. In Basile et al. (2014), è stata proposta un'estensione del RI denominata *Temporal Random Indexing* (TRI). In quest'ultimo caso si tratta di pochi esempi selezionati manualmente basati sui testi italiani del Progetto Gutenberg, manca una valutazione quantitativa di questo approccio, quindi non è semplice capire se il rilevamento dello spostamento semantico sia un compito per il quale TRI possa essere considerato migliore di altri modelli

[12] Gli autori hanno proposto la concettualizzazione di un modello distributivo nel tempo: si tratta cioè di vettori semantici (temporali) di ogni parola per ogni intervallo di tempo in analisi. Questo approccio ha favorito il confronto quantitativo delle parole relativamente al loro significato, ma anche quello dello sviluppo del significato delle parole nel tempo.
[13] Si rimanda la descrizione all'articolo di Kanerva et al., 2000.
[14] Se ne trova una descrizione in Deerwester et al., 1990.

distributivi. Con un approccio diverso Mitra et al. (2014) hanno analizzato i risultati del clustering grafico di parole in corpora diacronici. Il loro modello distributivo consiste in nodi lessicali dentro grafi connessi con archi pesati, i cui pesi sono calcolati in base ai conteggi delle co-occorrenze scalati da *Mutual Information* (MI). È importante sottolineare che sono stati in grado di rilevare non solo il semplice evento dato da uno spostamento semantico, ma anche il suo tipo: la nascita di un nuovo senso, la scissione di un vecchio senso in diversi nuovi o la fusione di più sensi in uno. Per le sue caratteristiche, tale lavoro entra in una classe molto meno rappresentata di approcci "a grana fine" del rilevamento del fenomeno. Gli autori trattano nativamente la questione delle parole polisemiche, mettendo la questione dei sensi delle parole nella giusta considerazione.

Il lavoro di Kim et al. (2014) è probabilmente il primo ad impiegare modelli di word embedding basati sulla previsione per tracciare spostamenti semantici diacronici, usando *Continuous Skipgram with Negative Sampling* (SGNS) [Mikolov et al., 2013] invece di *Continuous Bag-of-Words* (CBOW), scelta alternativa per l'apprendimento dei vettori semantici. Sempre in questo filone di studi Levy e Goldberg (2014) dimostrano che Skip-Gram di Word2Vec fattorizza implicitamente una matrice *Pointwise Mutual Information* (PMI) di co-occorrenze delle parole. Continuando questi studi Hamilton et al. (2016b) hanno mostrato la superiorità di SGNS rispetto ai modelli distributivi espliciti basati su PPMI (*Positive Pointwise Mutual Information*), sebbene abbiano notato che le approssimazioni SVD[15] (*Singular Value Decomposition*) di basso rango possono funzionare alla pari con SGNS, specialmente su set di dati di dimensioni contenute, come già indicato da Bullinaria et al. (2007). Da allora, la maggior parte delle pubblicazioni nel campo ha iniziato a utilizzare rappresentazioni di parole dense: o sotto forma di matrici PPMI fattorizzate da SVD, o sotto forma di modelli neurali basati sulla previsione. Di approccio diverso sono le ricerche di Freemann e Lapata (2016) che si sono ispirate a modelli tematici per creare un modello bayesiano dinamico del cambiamento

[15] Particolare fattorizzazione di una matrice basata sull'uso di autovalori e autovettori.

del significato diacronico, in cui il significato delle parole è modellato come l'insieme dei sensi, che vengono tracciati su una sequenza di intervalli di tempo contigui.

Resta il problema di fondo del confronto tra modelli diversi, per esempio come confrontare calcoli del coseno tra le immersioni di una stessa parola in due modelli diversi? Processi di apprendimento distinti, anche quando addestrati sugli stessi dati, possono produrre vettori numerici diversi a causa degli algoritmi di word embedding stocastici/statistici sottostanti. Questo fenomeno si apprezza di più su corpora diversi perché anche se il significato della parola è completamente stabile, la similarità diretta tra vettori di diversi periodi di tempo può rimanere relativamente bassa, semplicemente perché le inizializzazioni casuali dei due modelli erano diverse, a maggior ragione se viene utilizzata una metrica quale la distanza del coseno, che comprime i valori sulla superficie di un ipersfera di raggio 1 ed è quindi meno fine. Per ovviare a questo, Kulkarni et al. (2015) hanno suggerito che prima di calcolare le similarità, si dovrebbe aver precedentemente allineato i modelli, per adattarli in uno spazio vettoriale unico, utilizzando trasformazioni lineari che ne preservano la struttura generale. La mole dei dati che ora è possibile trattare ha comunque condizionato le analisi del fenomeno, molti studi infatti hanno optato per la visualizzazione grafica dei dati per arrivare ad una migliore valutazione delle regolarità del cambiamento semantico; lo hanno fatto Hilpert et al. (2015), Michel et al. (2011) e Rohrdantz et al. (2011).

Nel 2019 è stato tenuto un tutorial sull'argomento [Eisenstein 2019] e durante la Conferenza Annuale dell'ACL (*Association for Computational Linguistics*) 2019, si è tenuto il primo workshop internazionale sugli approcci computazionali al cambiamento linguistico storico (LChange'19) [Tahmasebi et al. 2019]. L'evento ha visto la partecipazione di molte comunità che lavorano in questo campo, ne è dimostrazione il fatto che più del 30% degli articoli era dedicata a questo tema.

Le immersioni contestuali, come ELMo [Peters et al. 2018], GPT-2 [Radford et al. 2019] e BERT [Devlin et al. 2019] basate sull'architettura

Transformer [Vaswani et al. 2017], rappresentano gli ultimi sviluppi. Sono approcci che utilizzano un 'modello di linguaggio mascherato' MLM (*Masked Language Model*) come obiettivo di pre-apprendimento. In particolare i modelli neurali basati su BERT stanno ottenendo risultati all'avanguardia in molte applicazioni NLP. Nel 2020 SemEval (*Semantic Evaluation*) presenta il primo task sul rilevamento del cambiamento semantico lessicale senza supervisione, su quattro lingue [Schlechtweg et al. 2020], seguito poi dal task di semantica lessicale diacronica (DIACR-Ita) sull'italiano, ad EVALITA 2020 [Basile et al. 2020].

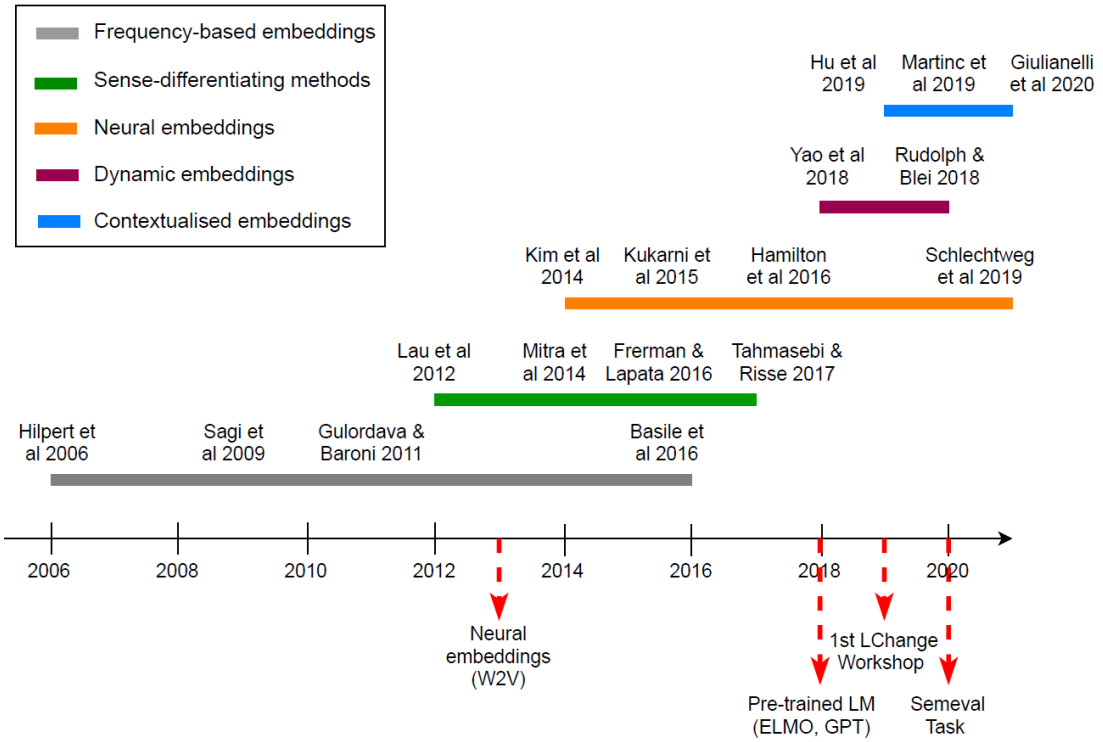

Figura 1: sintesi grafica dello stato dell'arte realizzata da S. Montariol in 'Models of diachronic semantic change using word embeddings'

L'evoluzione dello studio del cambiamento semantico e il grande sviluppo che ha visto il moltiplicarsi di iniziative in anni recenti, si comprende bene nell'immagine che ne propone Syrielle Montariol nel suo '*Models of diachronic semantic change using word embeddings*' (figura 1).

4.1 SemEval e DIACR-Ita overview

Per individuare il metodo più corretto per studiare il cambiamento semantico utilizzando i modelli BERT, è stato utile analizzare in dettaglio i risultati ottenuti in SemEval 2020: una serie di workshop di ricerca internazionali sull'elaborazione del linguaggio naturale, che mettono a confronto lo stato dell'arte in vari compiti di analisi semantica. Ogni anno i workshop presentano una raccolta di compiti di analisi semantica computazionale su cui si cimentano sistemi progettati da diversi gruppi. Nel 2020 è stato presentato lo specifico task: "*SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection*" Gli organizzatori del task hanno cercato di affrontare i problemi che ostacolano il progresso nel rilevamento del cambiamento semantico lessicale, ossia l'individuazione di metriche di valutazione e la mancanza di sufficienti gold standard. L'evento del 2020 è stato infatti il primo compito condiviso che ha affrontato questa lacuna fornendo ai ricercatori un quadro di valutazione e alcuni set di dati di alta qualità annotati manualmente per inglese, tedesco, latino e svedese[16]. Presentando il task gli organizzatori si auspicavano ricadute a lungo termine nello studio del cambiamento di significato delle parole e ne incoraggiavano lo sviluppo per altre lingue, in particolare per quelle con risorse limitate. Osservazioni che l'esperienza di questa tesi porta a condividere.

Lo stadio attuale degli studi rappresenta un'occasione di fondamentale importanza per confrontare approcci e strategie, per la prima volta infatti, è stato possibile confrontare i sistemi proposti dai partecipanti su basi relativamente solide e tra lingue diverse, e analizzare le conclusioni raggiunte. Allo stesso tempo ottenere risposte a questioni riguardanti le prestazioni di diversi tipi di rappresentazioni semantiche (come immersioni di token rispetto a immersioni di tipi e modelli di argomenti rispetto a modelli di spazio

[16] Ai partecipanti erano forniti un gold standard LSC multilingue (inglese, tedesco, latino, svedese) di alta qualità, basato su circa 100.000 istanze prodotte da giudizi di esperti.

vettoriale), metodi di allineamento e misure di cambiamento semantico. Un'analisi approfondita dei risultati, ha consentito di mettere in luce le tendenze nell'utilizzo dei modelli, e fornire ispirazione per eventuali miglioramenti. Operativamente il task si basava sul confronto di due corpora C1 e C2 relativi a due periodi di tempo diversi. Sebbene ciò semplificasse il problema del rilevamento del fenomeno LSC, riconducendolo all'analisi ad un arco temporale stabilito, presentava due vantaggi principali: (i) ridurre il numero di periodi di tempo per i quali i dati dovevano essere annotati manualmente, in modo da poter annotare campioni di corpus più grandi e quindi rappresentare in modo più affidabile le distribuzioni dei sensi delle parole target; (ii) ridurre la complessità del compito, consentendo l'applicazione di diverse architetture e modelli, ampliando la gamma di possibili partecipanti. Il task era organizzato in due sotto-attività:

- Classificazione binaria: per un insieme di parole target, decidere se e quali parole avessero perso o acquisito senso(i) tra C1 e C2.
- Ranking: classificazione di un insieme di parole target in base al loro grado di LSC tra C1 e C2.

Dopo uno studio generale dei contributi presentati per i due compiti correlati, si è deciso di indirizzare l'analisi sui lavori che avevano adottato i modelli BERT. Tra questi troviamo specializzazioni come DiaSense[17], sistema che si basa su 'bert-as-service' [Xiao, 2018][18], una libreria Python che utilizza il modello BERT di Google come codificatore di frasi, ospitandolo come servizio tramite ZeroMQ4. Un'altra specializzazione era CMCE[19] (*Clustering on Manifolds of Contextualized Embeddings*), un sistema che utilizza words embeddings contestualizzate MBERT, la cui dimensionalità è ridotta attraverso un autoencoder e l'algoritmo UMAP[20]. Gli autori ne proponevano

[17] C. Beck, Modeling sense change via pre-trained BERT embeddings. SemEval-2020.
[18] https://bert-as-service.readthedocs.io/en/latest/
[19] D. Rother, Clustering on Manifolds of Contextualized Embeddings to Detect Historical Meaning Shifts. SemEval-2020.
[20] UMAP è un algoritmo di riduzione della dimensionalità e un potente strumento di analisi dei dati, simile a PCA (Principal Component Analysis) in termini di velocità e a tSNE per quel che riguarda la preservazione di quante più informazioni possibili sul set di dati.

l'adozione con lo scopo di poter utilizzare una gamma più ampia di algoritmi di clustering. Le sperimentazioni che hanno utilizzato BERT nella sua configurazione più classica, erano più attinenti a quanto si voleva realizzare in questo lavoro di tesi, pertanto si è deciso di dedicarvi più dettaglio.

Kutuzov e Giulianelli (2020) hanno usato i contextualized word embeddings per il rilevamento del cambiamento semantico lessicale, ma si sono dedicati prevalentemente alla seconda sotto-attività, ovvero provando a classificare le parole in base al grado di cambiamento/deriva semantica nel tempo. In particolare hanno analizzato le prestazioni di due modelli (BERT e ELMo) e applicato tre diversi algoritmi per il rilevamento del cambiamento semantico. Partendo dalle matrici di embedding relative alle estrazioni dai due archi temporali hanno applicato: *cosine similarity* inverso applicato a 'prototipi' (~embedding medi) di parole (PRT); Distanza media del coseno a coppie tra token embedding (APD); Divergenza Jensen-Shannon (JSD). Gli autori affermano che nelle sperimentazioni il modello ELMo è superato da BERT nella maggior parte delle attività NLP, ma la sua architettura più leggera, formata da un LSTM bidirezionale a due strati sopra uno strato convoluzionale e un numero inferiore di parametri, consentono un training e un'inferenza più rapidi. Inoltre hanno ottenuto i migliori risultati nel task adottando, come metriche di rilevamento di LSC, la somiglianza del coseno degli embeddings contestualizzati medi e la distanza media a coppie tra gli embeddings contestualizzati. I risultati da loro ottenuti hanno superato le baselines[21] di un ampio margine.

Martinc et al. (2020b) hanno ripreso il lavoro di Giulianelli et al. (2019) considerando più vantaggiosi i metodi basati su embeddings contestuali e clustering per il rilevamento del cambiamento semantico lessicale. Gli autori hanno ritenuto che questa strategia consentisse un'analisi "a grana fine" del cambiamento nell'uso delle parole, un vantaggio interpretativo che le aggregazioni delle immersioni riflettono sui diversi usi della parola.

[21] Gli autori affermano che nella fase post-valutazione, ottengono la migliore sottomissione per il Sotto-task 2.

In questo contesto di studi gli algoritmi di clustering prendono in input matrici formate da tutti gli embedding estratti. Tali matrici possono essere distinte per ognuno dei corpora o confluire in un unico insieme, composto da tutti i corpora da analizzare. Il clustering crea dei raggruppamenti (cluster) che corrispondono ad una similarità trovata dagli algoritmi rispetto al tipo di 'distanza' che implementano. Il problema del clustering per questo tipo di attività resta la scalabilità in termini di consumo di memoria e tempo di calcolo (§ 6.4), caratteristica che ne ha limitato l'uso in attività esplorative aperte, e ne ha consentito l'uso solo per un piccolo numero di parole target preventivamente selezionate.

Studi recenti [Kutuzov e Giulianelli 2020; Martinc et al., 2020a] mostrano che il raggruppamento di immersioni contestuali potrebbe essere un proxy per il cambiamento dell'uso delle parole. Il loro assunto è che se i cluster, che in teoria catturano usi distinti delle parole, si distribuiscono in modo diverso in periodi di tempo differenti, è ragionevole dedurne un cambiamento di senso della parola. L'approccio basato su cluster offre generalmente un'interpretazione più intuitiva del cambiamento nell'uso delle parole rispetto ai metodi che invece analizzano i 'vicini' più prossimi di una parola per ogni periodo di tempo in esame [Gonen et al., 2020; Martinc et al., 2020b]. Per grandi corpora, con ampi vocabolari, alcune parole possono apparire milioni di volte, se il clustering viene applicato a ciascuna parola, l'aggregazione in cluster di tutte le sue occorrenze è un'operazione che rende l'uso di questi metodi proibitivo. Uno scenario di questo tipo richiede che vengano conservati in memoria grandi matrici di embedding da sottoporre all'algoritmo di clustering. Il problema è quindi duplice: è richiesto tanto spazio di memoria per conservare i dati durante il processo di estrazione e serve tanta capacità di elaborazione affinché il processo di clustering possa trattare dati così grandi.

Un modo largamente usato per risolvere il problema della scalabilità, utilizzando immersioni contestuali, consiste nel fare la media dell'insieme di rappresentazioni contestuali per ogni parola producendo una singola rappresentazione [Martinc et al., 2020b]. La media, sebbene scalabile, perde

molto in interpretabilità, poiché gli usi delle parole sono fusi in un'unica rappresentazione. Per ovviare a questo Montariol et al. (2021) per ogni parola, generano un insieme di immersioni contestuali utilizzando BERT. Queste rappresentazioni sono raggruppate utilizzando K-mean[22] e/o Propagation Affinity[23] (PA) [Martinc, 2020a] e le distribuzioni di cluster derivate vengono confrontate su intervalli di tempo utilizzando la divergenza di Jensen-Shannon (JSD) [Lin, 1991] o la distanza di Wasserstein (WD) [Solomon, 2018]. La strategia adottata dagli autori consiste nel creare aggregazioni di embeddings 'vicini' in termini della similarità del coseno, mano a mano che questi sono generati. Arrivati a una quota empiricamente definita (nel caso specifico 200) si memorizza il precedente e si genera un nuovo aggregato procedendo sempre nello stesso modo. Al termine si esegue il clustering di queste aggregazioni con l'obiettivo di estrarre la distribuzione dell'uso della parola in ogni periodo. Gli autori sostengono che il tipo di raggruppamento prodotto dal clustering risulta solitamente distorto: un numero limitato di grandi cluster accompagnato da molti cluster costituiti solo da un paio di istanze. Nel confronto dei due algoritmi, a parte la differenza sul numero di cluster che K-means richiede come parametro iniziale, gli autori sostengono poi che PA consente di individuare i sensi centrali di una parola, mentre K-mean tende a produrre cluster più uniformi.

La produzione di piccoli cluster, che contengono solo poche istanze, e che quindi non rappresentano un senso o un uso specifico della parola, è relativamente comune nell'uso del clustering in questo contesto, infatti alcuni studi hanno mostrato come BERT sia sensibile alla sintassi e alla pragmatica, che in generale non rappresentano indicatori significativi per il rilevamento di

[22] Algoritmo di clustering la cui strategia è minimizzare la varianza totale intra-gruppo e in cui ogni gruppo viene identificato mediante un centroide. In realtà viene usata una sua versione semplificata, conosciuta come algoritmo di Lloyd [Lloyd, 1982], che risolve K-means in modo approssimato, a vantaggio della velocità di elaborazione.

[23] Algoritmo di clustering utilizzato prevalentemente in statistica e data mining, basato sul concetto di "passaggio di messaggi" tra punti. Diversamente da altri algoritmi quali il k-means, affinity propagation non richiede che sia definito a priori il numero di cluster. In modo simile a k-medoids cerca membri "rappresentativi" dell'insieme di input, individuando rappresentanti di singoli cluster.

cambiamento semantico. Per le analisi e l'individuazione dei cambiamenti diacronici applicano ai cluster la teoria del trasporto ottimale[24]. Tale teoria tratta l'omonimo problema, che, in matematica, consiste nel capire come trasportare una distribuzione di massa da un posto ad un altro, appunto, "in maniera ottimale". Prendendo come funzione di costo una misura di distanza e come masse le frequenze relative è possibile applicare l'algoritmo al confronto tra distribuzioni di probabilità definite in uno spazio geometrico, come è quello degli embeddings di 768 dimensioni creati da BERT. Un algoritmo che risolva questo tipo di problemi in tempi ragionevoli al crescere dei punti e delle dimensioni dello spazio è stato reso possibile solamente di recente, tramite tecniche di calcolo numerico. Costruendo un unico corpus diacronico e indicando come sezioni proprie i sotto-corpora sincronici, corrispondenti ai relativi intervalli temporali, vengono presi gli embeddings di una parola in due sezioni del corpus e applicato l'algoritmo di clustering. In questo modo si possono ottenere, per ciascuna delle sezioni di testo, le frequenze relative nei cluster e la posizione dei centroidi. Come funzione di costo viene usata la matrice delle distanze del coseno tra i centroidi dei cluster delle occorrenze della prima sezione e quelli della seconda. Come massa di partenza e di arrivo si considerano le due distribuzioni di frequenza. L'uso della frequenza relativa è necessario perché la massa totale deve rimanere costante. Il risultato ottenuto in questo modo tiene conto contemporaneamente delle distribuzioni nei singoli cluster e della loro posizione nello spazio semantico.

Karnysheva et al. (2020) hanno invece descritto un approccio che partendo da modello ELMo prevedeva il raggruppamento di immersioni utilizzando gli algoritmi Dbscan[25] (*Density-Based Spatial Clustering of*

[24] L'esistenza di un trasporto ottimo nel caso in cui il costo fosse uguale alla distanza era, sino agli anni '80, ancora un problema aperto e fu risolto solamente alla fine degli anni '90 quando Yann Brenier, mentre studiava questioni di meccanica dei fluidi (le equazioni di Eulero per fluidi incomprimibili), si trovò a valutare il problema di Monge-Kantorovich: in cui però, il costo di trasporto non era proporzionale alla distanza, bensì al quadrato della stessa (una sorta di energia cinetica). Brenier dimostrò che, in questo caso, il modo di trasportare materia minimizzando i costi esiste ed è unico.

[25] Metodo di clustering proposto nel 1996 da Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. È basato sulla densità perché connette regioni di punti con densità

Applications with Noise) e K-means. Per risolvere il problema dell'inizializzazione del clustering di K-means (cioè trovare il numero k di cluster) hanno utilizzato un approccio dinamico. I possibili valori che k può assumere sono stati impostati attraverso un intervallo. Nel loro modello, il limite superiore dell'intervallo era posto a 10, il che significa che il clustering doveva essere eseguito ripetutamente con valori da 1 a 10 e convergesse quando l'errore al quadrato della distanza tra i punti dati e il loro centro del cluster raggiungeva un minimo locale: approccio definito come "metodo del gomito"[26]. Quanto all'adozione del modello ELMo le autrici hanno affermato essere dovuta alla sua caratteristica principale, ovvero l'utilizzo di tutti i livelli di rappresentazione interni per calcolare la rappresentazione finale degli embeddings. Questa peculiarità era indicata dalle autrici come determinante nel permettere di acquisire più informazioni a diversi livelli linguistici (semantica, sintassi, ecc.), rispetto alle semplici immersioni di parole [Peters et al., 2018]. Karnysheva e Schwarz, sempre giustificando l'adozione del modello ELMo, hanno inoltre definito un 'difetto' il fatto che le immersioni pre-addestrate di BERT lavorassero su corpora non lemmatizzati, a differenza dei corpora forniti dal task che invece lo erano. Nel loro sistema il numero e la dimensione dei cluster determinano se una parola mostra un cambiamento di senso o meno e hanno quindi calcolato il grado di variazione di una parola con la distanza di JSD applicata ai vettori che rappresentano le distribuzioni di probabilità dei cluster.

Per il task era infatti disponibile una griglia di riferimento per valutare il grado di shift semantico che era così definita: perché una parola abbia acquisito un senso, il senso deve apparire un massimo di k volte in C1 e almeno n volte in C2, perché una parola abbia perso un senso, questo deve apparire un massimo di k volte in C2 e almeno n volte in C1. I valori per k e n erano stati impostati in modo diverso per il latino (k = 0, n = 1) poiché il campione dei dati

sufficientemente alta. L'algoritmo stima la densità attorno a ciascun punto (contando il numero di punti in un intorno ε (o eps) specificato dall'utente, ed applica delle soglie chiamate minPts per identificare i punti "core", "border" e "noise".

[26] https://en.wikipedia.org/wiki/Elbow_method_(clustering)

aveva una dimensione inferiore rispetto alle altre tre lingue, per le quali erano indicati k = 2 e n = 5. I loro risultati hanno superato le baselines per entrambe le sotto-attività.

Un altro lavoro [Kanjirangat et al. 2020] impiegava anch'esso il clustering con K-means, suddividendo l'approccio in due metodi diversi:

- metodo 1: raggruppamento di vettori di entrambi i corpora;

- metodo 2: clustering separato dei due corpora.

Entrambi i metodi erano basati su raggruppamenti degli embeddings associati alle parole target, ma mantenevano distinte le sperimentazioni tra singolo corpus e unione dei due. Per la selezione del numero ottimale di k e l'inizializzazione sulla migliore posizione dei centroidi [Rousseeuw, 1987], gli autori sono ricorsi al metodo *silhouette score*[27]: dato un valore di k, si calcolano nel cluster corrispondente, le medie sia della 'nearest-inter cluster distance' che della 'nearest-cluster distance'. La differenza tra queste due grandezze, normalizzate dal massimo delle due, è stata utilizzata come punteggio idoneo da massimizzare per selezionare il valore ottimale di k. Lo stesso approccio è stato utilizzato per determinare i centroidi iniziali. Gli autori avevano previsto anche una robusta elaborazione grafica in grado di fornire informazioni sulla qualità dei cluster prodotti.

Le lingue trattate in SemEval non comprendevano l'italiano, quindi prove dirette sui dati del task non erano possibili. Di recente Basile et al. (2020) hanno organizzato un compito di rilevamento del cambiamento semantico lessicale per l'italiano chiamato DIACR-Ita, una riproposizione del sotto-task 1 di SemEval, che riguardava come stabilire se un insieme di parole target avesse cambiato significato in due periodi, t1 e t2, dove t1 precede t2. Sempre con l'obiettivo di condurre un'analisi della letteratura per la lingua trattata e i

[27] Il silhouette score (o coeff. di Silhouette) è una misura di quanto un oggetto sia simile al proprio cluster (coesione) rispetto ad altri cluster (separazione). Il suo valore varia da -1 a +1, dove un valore alto indica che l'oggetto è ben abbinato al proprio cluster e scarsamente abbinato ai cluster vicini. Se la maggior parte degli oggetti ha un valore elevato, la configurazione del cluster è appropriata. Se molti punti hanno un valore basso o negativo, la configurazione del cluster potrebbe avere troppi o troppo pochi cluster.

dati di cui si dispone, si sono studiati anche i lavori presentati al task, soprattutto quelli che hanno utilizzato i modelli BERT.

Il metodo messo a punto da Laicher et al. (2020) utilizzava la distanza media a coppie di BERT embeddings basati su estrazione di token contestualizzati da periodi temporali diversi. Gli autori sostenevano di non comprendere come non fossero riusciti a conseguire gli stessi risultati ottenuti sul set di dati inglese del task 1 di SemEval 2020, dove avevano raggiunto prestazioni elevate. Sul set di dati DIACR-Ita per l'italiano invece avevano chiuso il compito con una precisione di 0.72. Tra le possibili ragioni indicavano il mancato fine-tuning del modello sull'italiano, anche se dalla loro esperienza su set di dati in inglese, avevano già sperimentato la possibilità di raggiungere prestazioni eccezionalmente elevate anche senza messa a punto. Il loro approccio utilizzava la distanza del coseno tra vettori di embeddings contestualizzati estratti da due periodi di tempo t1 e t2 e quindi la *Average Pairwise Distance* (APD) per misurare la loro distanza a coppie [Sagi et al., 2009; Schlechtweg et al., 2018; Beck, 2020; Kutuzov e Giulianelli, 2020]. Il punteggio che indicava il cambiamento semantico lessicale della parola era quindi configurato come la distanza media di tutti i confronti. La scelta di APD è stata raggiunta dopo aver testato diverse misure, ed è stata adottata come la più adatta al compito poiché ha prodotto sui dati i risultati migliori. Gli autori hanno anche testato l'estrazione di embeddings di diversi livelli per valutarne il migliore adattamento al compito.

Wang et al. (2020) hanno invece formalizzato l'attività come l'individuazione di una misura di distanza per due insiemi di vettori di dimensioni flessibili. Hanno quindi utilizzato varie metriche di distanza: distanza euclidea media, distanza media di Canberra, distanza di Hausdorff (HD), hanno anche sperimentato strategie di clustering basate su K-means e sul modello Gaussian Mixture[28] sui quali hanno applicato la divergenza di

[28] I modelli di aggregazione Gaussian Mixture (GMM) presuppongono che vi sia un certo numero di distribuzioni gaussiane e ciascuna di queste distribuzioni rappresenti un cluster. Quindi, un modello di miscela gaussiana tende a raggruppare i punti dati appartenenti a una singola distribuzione.

Jensen-Shannon per valutare le distribuzioni dei cluster. Gli autori hanno ottenuto con questo metodo prestazioni in linea con la baseline prevista dal task.

Capitolo 5

I corpora diacronici: ILC-Ita

Il problema del reperimento di corpora di addestramento per varietà storiche della lingua italiana è reale. I materiali testuali digitali si trovano ma, anche se già acquisiti da sistemi di OCR (*Optical Characters Recognition*), spesso sono stati prodotti senza una consapevolezza del dato digitale: l'interpretazione del formato digitale rimane un problema di fondo anche dopo l'avvento del formato XML[29] TEI[30]. Non solo nei primi tentativi di digitalizzazione, anche più recentemente, si è spesso interpretata l'operazione di digitalizzazione come una "copia" di quanto presente nel formato cartaceo. Le scelte di codifica non sono uniformi, spesso dettate da aspetti contingenti legati allo strumento o al progetto che le ha prodotte. Anche i tentativi di attenersi a regole di redazione, quando l'intervento è stato prodotto in epoche in cui mancava una norma di riferimento, hanno poi prodotto formati particolari, spesso senza un'adeguata

[29] L'XML che conosciamo oggi nasce come progetto alla fine del 1996, nell'ambito della SGML Activity del W3C. Per il suo sviluppo fu creato dal W3C un gruppo di lavoro, XML Working Group, composto da esperti mondiali delle tecnologie SGML, ed una commissione, XML Editorial Review Board, deputata alla redazione delle specifiche del progetto. Nel Febbraio del 1998, le specifiche sono divenute una raccomandazione ufficiale, con il nome *Extensible Mark-up Language* (XML).

[30] Progetto internazionale promosso nel 1988 dalle maggiori associazioni professionali di informatica linguistica ACL, ACH (*Association for Computers and Humanities*) e ALLC (*Association for Literary and Linguistic Computing*), per lo sviluppo di un sistema di codifica che riordinasse le diverse rappresentazioni dei testi in formato elettronico. Nel 1991 sono state pubblicate le Guidelines for *Electronic Text Encoding and Interchange* (TEI P1), contenenti le prime specifiche provvisorie della codifica TEI. Oggi è divenuto il formato standard di rappresentazione per diverse tipologie di materiali testuali: https://tei-c.org/

documentazione a supporto delle scelte fatte. In generale nei testi che traducono articoli di giornale questo aspetto ha prodotto una bassa qualità dei dati. Come spesso accade per altre lingue, anche per l'italiano, la scelta di questo tipo di testi è quasi obbligata poiché sono quelli di buona disponibilità e maggiore quantità. Il grande corpus testuale sul quale è stato pre-addestrato il modello BERT per l'italiano ne contiene anch'esso in buona misura.

Altro aspetto importante riguarda propriamente l'indagine sul cambiamento di senso, che risulta ulteriormente complicata dal fatto che, una variazione osservata nella forma lessicale tra diversi materiali testuali, pur provenendo da periodi di tempo diversi, può non essere affatto dovuta a cause diacroniche. I linguisti sanno bene che anche guardandola come un'entità sincronica, la lingua è piena di variazioni a tutti i livelli linguistici, quindi mantenere la specificità del tipo di dato diventa essenziale.

Con queste premesse è stata fatta una rassegna dei corpora testuali disponibili a contenuto giornalistico-informativo per la lingua italiana. Lo spirito dell'indagine era capire quali corpora fossero disponibili per la varietà di italiano che interessava, partendo dalla letteratura scientifica sullo stesso tema. Alcune ricerche indagano il fenomeno ma utilizzano risorse linguistiche specializzate. In particolare Giovanni Semeraro e P. Basile dell'Università di Bari, hanno presentato alla conferenza CLIC 2014 la costruzione di un framework, chiamato *Temporal Random Indexing* [Semeraro et al., 2014]. Lo stesso gruppo di Bari ha sviluppato AlBERTo, un modello BERT per la lingua italiana moderna particolarmente focalizzato sul linguaggio utilizzato nei social network, nello specifico su Twitter, utilizzando come corpora di addestramento TWITA [Polignano et al., 2019].

Risorse più adatte alla sperimentazione in oggetto sono sembrate quelle offerte dal corpus PAROLE (*Preparatory Action for Linguistic Resources Organization for Language Engineering*) [Marinelli et al. 2003]. Il progetto europeo omonimo era finalizzato alla costruzione di banche dati testuali e lessicali, per le principali lingue europee, ampie, generiche e riutilizzabili, strutturate in modo uniforme. È stato uno dei principali progetti lanciati dalla

CE nei primi anni 2000 per la costruzione di risorse linguistiche per la lingua scritta. Già allora era evidente che la mancanza di grandi lessici computazionali e la non omogeneità delle risorse esistenti rappresentasse un ostacolo al progresso delle applicazioni NLP. La composizione della parte in lingua italiana, conformemente al resto del corpus, comprende libri in percentuale del 17,91%, articoli di giornale (49.68%), periodici (24,58%) e miscellanea per il restante (7,83%). Il consorzio di soggetti coinvolti nel progetto concordò che, per ciascuna delle lingue partecipanti il Corpus complessivo, di almeno 20 milioni di parole, fosse disponibile all'interrogazione presso la sede del partner di riferimento, mentre fu reso disponibile per la distribuzione alla comunità esterna un sotto-corpus di circa 3 milioni di token[31].

Un'alternativa era rappresentata dal corpus UNITA (A Diachronic Italian Corpus based on "L'Unità"). Il corpus copre 67 anni della storia del giornale omonimo, dal 1948 al 2014. Le dimensioni del corpus sono di poco superiori ai 4 milioni di token. Il task DIACR-Ita sfrutta porzioni di questo corpus [Basile et al., 2020]. La ricerca è proseguita con corpora di riferimento aggiuntivi, per esempio il corpus dei documenti pubblici di Alcide de Gasperi [Tonelli et al., 2019], che comprende 1.762 documenti (articoli di giornale, documenti di propaganda, lettere ufficiali, discorsi parlamentari, per un totale di 3.000.000 di tokens) scritti dal politico italiano Alcide De Gasperi e pubblicati tra il 1901 e il 1954[32]. Esistono poi altre risorse descritte ma non disponibili allo scopo[33] come DiaCORS, corpus diacronico che comprende testi prodotti tra il 1861 e il 2001. Si è valutato anche di utilizzare parte del corpus VoDIM[34] (Vocabolario dinamico dell'italiano moderno), in questo caso la selezione d'interesse riguardava il sotto-corpus realizzato dall'Università degli Studi di Milano. In

[31] Il corpus originario di 3 milioni circa di parole si localizza temporalmente tra il 1970 e il 1997, ne esiste anche una versione più estesa di 4 milioni di token, costituita da ulteriori articoli di giornale che vanno dal 1997 al 2005.
[32] Disponibile alla URL: https://github.com/StefanoMenini/De-Gasperi-s-Corpus
[33] Disponibili per l'interrogazione alla URL: http://corpora.dslo.unibo.it/DiaCORIS/
[34] Il progetto, finalizzato alla compilazione di un dizionario dell'italiano post-unitario basato su corpora, è un'impresa diretta a livello nazionale dal presidente dell'Accademia della Crusca, Claudio Marazzini, che a partire dal 2014, coordina i numerosi gruppi di ricerca coinvolti.

particolare dal gruppo di ricerca del progetto PRIN 2012: "Corpus di riferimento per un Nuovo Vocabolario dell'Italiano moderno e contemporaneo. Fonti documentarie, retrodatazioni, innovazioni" guidato da Ilaria Bonomi. Nel progetto il sotto-corpus è etichettato come 'giornali'[35] e presenta una sezione più storica (tardo ottocentesca) di questo tipo di materiali.

Tra gli altri materiali individuati sono stati analizzati "I Periodici Milanesi[36]". Un corpus rivelatosi da subito interessante, non solo perché il più antico e di epoca pre-unitaria, ma soprattutto perché interamente formato da articoli di giornale, relativi a 58 testate giornalistiche del periodo che va dal 1800 al 1847. I testi furono lemmatizzati con il contributo tecnologico dell'Istituto di Linguistica Computazionale (ILC) del CNR nei primi anni '80, attraverso procedure all'avanguardia per quel tempo e in modo semi-automatico. Gli articoli del corpus coprono generi testuali diversi[37]: informazione politica; riviste letterarie; di varietà; riviste tecniche; periodici teatrali; almanacchi e strenne. Un corpus di dimensioni di poco inferiori al milione di token che rappresenta uno spaccato prezioso dell'italiano pre-unitario non letterario. La reperibilità di materiale digitale ottocentesco di contenuto informativo è infatti molto carente e anche quando ne è stata conservata copia, le caratteristiche dei formati, spesso obsoleti, ne impediscono un pieno sfruttamento.

---

[35] Per il dettaglio sui dati di contenuto giornalistico si rimanda alla pagina dedicata nel sito del VoDIM: http://vodim.accademiadellacrusca.org/testi?g=UNIMI

[36] Il corpus dei materiali testuali è costituito dai testi lemmatizzati relativi ai periodici milanesi, utilizzati per la realizzazione a stampa dei 5 volumi "*La stampa periodica milanese della prima metà dell'Ottocento: testi e concordanze*"; editi da Giardini (Pisa) nel 1984 e curati da Stefania De Stefanis Ciccone, Ilaria Bonomi, Andrea Masini.

[37] I generi, a volte suddivisi in sottogeneri, sono: Politica; Teatro Letteratura; Belle Arti; Narrativa; Aneddotica; Cronaca; Divulgazione storico-geografica; Biografia; Osservazioni sui costumi contemporanei; Moda; Pubblicità; Agricoltura e Botanica; Chimica e Fisica; Invenzioni e scoperte; Economia (industria, artigianato e commercio); Economia domestica; Giurisprudenza; Medicina; Zoologia e zootecnia.

5.1 Caratteristiche di ILC-Ita

Le ragioni che hanno guidato la scelta di quale corpus fosse adattato alle sperimentazioni del progetto sono prevalentemente due. La prima è di tipo pratico e riguarda: l'accessibilità completa con possibilità di segmentazione interna dei dati; il periodo cronologico di riferimento; la tipologia dei contenuti; la qualità e dimensione dei dati. Quelle elencate sono condizioni da prediligere in un progetto sperimentale che necessita ad ogni passo di verifiche e di prove incrociate. La possibilità di ristrutturare il dato secondo le esigenze rende la flessibilità degli esperimenti molto efficace. Il secondo ordine di ragioni riguarda la possibilità di indagare il cambiamento semantico in varietà storiche della lingua, analizzare il fenomeno in una diacronia marcata. Nel contesto di ricerca che 'guarda indietro', verso varietà della lingua lontane nel tempo, mancano esperienze consolidate sullo studio del cambiamento di significato, soprattutto per la mancanza di dati sui quali sperimentare le soluzioni e *gold standard* sui quali verificare le ipotesi. Come affermato da diversi studi in letteratura la digitalizzazione e l'accesso a gradi quantità di dati 'storici' è una realtà che sta crescendo, ma gli strumenti tecnologici in grado di trattare quelle risorse sono ancora carenti.

Con le premesse appena descritte si sono scaricati il corpus UNITA, PAROLE, I Periodici Milanesi e il sotto-corpus VoDIM. L'analisi fatta sui dati scaricati ha evidenziato problemi di immediata usabilità per ognuno dei corpora:

- per il corpus UNITA nella versione testo (non annotata) sono legati prevalentemente alla presenza non trascurabile di errori prodotti dal sistema di OCR utilizzato, condizione che rende l'errore non strutturale e non eliminabile automaticamente. Spesso salta il concetto di frase, decisamente importante per il modello che vogliamo utilizzare.
- Il corpus PAROLE presenta un codifica non uniforme (spesso mancante) delle parti strutturali delle trascrizioni dei giornali dai quali è composto: titolo, sottotitolo, occhiello, firma, ecc.; con conseguenti problematiche di

parsing con strumenti software. In secondo luogo questi fenomeni sono stati codificati corredando il testo con codici di annotazione proprietari di cui si è persa la memoria esatta. Tale caratteristica è impattante sulle procedure di acquisizione, soprattutto per quanto riguarda la codifica adottata per tabelle, elenchi, grafici, punti elenco annidati, ecc. Nella stragrande maggioranza dei casi però si tratta di errori strutturali e ripetitivi, che possono essere corretti attraverso l'uso di espressioni regolari. Nei casi più problematici, di annidamento di più codifiche, il rischio di rendere incomprensibili intere frasi ha reso necessario l'eliminazione di intere parti del testo.

Elementi ricondotti alla forma grafica separata		**Elementi ricondotti alla forma contratta più comune**	
altra_volta	altra volta	in_vece	invece
buon_senso	buon senso	in_oltre	inoltre
con_tutto_ciò	con tutto ciò	più_tosto	piuttosto
di_dietro	di dietro	sopra_tutto	soprattutto
di_fronte	di fronte	fin_che	finché
di_fuori	di fuori	in_fine	infine
di_sopra	di sopra	pur_troppo	purtroppo
lo_che	lo che	e_pure	eppure
non_che	non che		
nulla_ostante	nulla ostante		
per_anco	per anco		
tutto_al	tutto al		
via_meglio	via meglio		

Tabella 1: esempi di interventi manuali operati sui testi del corpus PM

- Il corpus de "I Periodici Milanesi" (PM) presentava testi in versione lemmatizzata (ogni parola corredata da annotazione morfo-sintattica), condizione che ha reso necessaria una conversione dei dati. Fattore ulteriore di complessità nel loro trattamento è data dal fatto che la fase di lemmatizzazione fu eseguita a mano da studiosi che annotarono una serie di fenomeni, secondo criteri linguistici teorici. Questi interventi hanno legato in un'unica forma alcune sequenze, come locuzioni e avverbi, che lasciate in questa forma renderebbero meno efficace la loro usabilità per l'addestramento di uno strumento automatico. Tali ragioni hanno portato verso la normalizzazione di queste grafie, che, grazie alla consulenza di

linguisti dell'ILC, ha prodotto una maggiore uniformità di alcune locuzioni e avverbi molto comuni.

- Il sotto-corpus del VoDIM preso in esame riguardava i materiali a contenuto giornalistico e di recente costituzione e revisione. I testi presentavano infatti un formato XML TEI e hanno quindi richiesto un intervento minimo per la loro piena usabilità. L'unico problema ha riguardato l'errata codifica di alcuni caratteri, che si verifica quando si opera componendo documenti senza consapevolezza del formato dei dati. È infatti comune la pratica del "copia e incolla" da documenti che hanno formati diversi (ANSI, invece che UTF-8) e che provoca l'errata interpretazione di diversi caratteri accentati. La porzione estratta ha riguardato esclusivamente la parte dei testi ottocenteschi poiché il resto dei testi era coevo del corpus PAROLE ed essendo molto più piccolo non è stato acquisito.
- I restanti testi, sebbene sulla carta sembrassero promettenti, nei fatti si sono dimostrati non privi di problemi e differenze, non ultimo una comoda accessibilità e la possibilità di scomporli in parti.

Il lavoro necessario a rendere i corpora pienamente utilizzabili ha riguardato soprattutto i dati più antichi e acquisiti digitalmente alla fine degli anni '80 dello scorso secolo: non solo formati digitali obsoleti ma anche annotazione manuale. Per ottenere il formato testo desiderato, dalla versione annotata morfo-sintatticamente del corpus PM, è stato necessario implementare un parser software *ad hoc* per l'estrazione del testo. L'analisi dei risultati estratti ha evidenziato una serie di errori di vario tipo, in gran parte dovuti alla redazione del formato originale (a record di dimensione fissa), prodotto con l'ausilio di software profondamente condizionato dalle limitate capacità di memoria dell'informatica del tempo. Le soluzioni individuate per risolvere tali errori ne hanno ridotto il volume. Anche il corpus PAROLE ha subito una fase di ripulitura, questa volta implementando con espressioni regolari, regole in grado di catturare gli elementi da correggere: elementi di struttura costruiti come sequenze di caratteri; individuazione della differenza tra punti di abbreviazione e punti fermi, necessari alla chiusura delle frasi; segmentazione del file in righe contenenti un'unica frase.

Con queste premesse si è costruito il corpus denominato ILC-Ita, nella sua versione completa e costituito da tutti i testi indicati in precedenza e mostrati in tabella 2 a pag. 25. Il corpus è quindi l'unione di più corpora di epoche diverse, ma dal quale possono essere estratti più sotto-corpora sincronici. Dall'analisi fatta è sensato estrarre tre corpus:

- ILC-Ita1 (materiali post 1990)
- ILC-Ita2 (materiali post 1980)
- ILC-Ita3 (articoli di giornale 1800)

Materiali	Testi	anni di rif.	n. parole	Somme parziali
Quotidiani post 1990	La Repubblica	1995-2005	2.083.500,00	
	ZeroUno	1990?	101.395,00	
	Il Sole 24 Ore	1995-2002	2.303.100,00	
	Unione Sarda	1996	533.781,00	
				5.021.776,00
Periodici post 1980	Casaviva	1985-1988	114.849,00	
	Cento Cose	1985-1988	116.473,00	
	Epoca	1985-1988	120.405,00	
	Espansione	1985-1988	78.581,00	
	Grazia	1985-1988	111.196,00	
	Panorama	1985-1988	82.403,00	
	Starbene	1985-1988	119.799,00	
	Storia illustrata	1985-1988	114.154,00	
	Zerouno	1985-1988	101.395,00	
				959.255,00
Giornali '800	Periodici Milanesi	1800-1847	690.200,00	
Gionali VoDIM '800	Il Corriere	1886-1996	345.500,00	
	Il Nuovo Corriere	1867-1887	99.200,00	
	Gazzetta Piemontese	1867-1887	31.370,00	
				1.166.270,00
			Totale	**7.147.301,00**

Tabella 2: dimensioni e caratteristiche del corpus ILC-Ita

Un esame più approfondito individua un ulteriore suddivisione interna al corpus ILC-Ita3. Infatti "I Periodici Milanesi" sono precedenti all'unità d'Italia, mentre gli altri sono post-unitari e questo si evidenzia sia nella costruzione delle frasi che nell'uso di forme verbali e aggettivi. Per l'analisi di varietà storiche dell'italiano avrebbe senso quindi dividere in ILC-Ita31 (Periodici

Milanesi) e ILC-Ita32 (parte VoDIM) ma il numero di token in ILC-Ita32 è risultato troppo esiguo per essere trattato singolarmente.

Capitolo 6

Metodo

6.1 Il metodo implementato

L'ipotesi che inquadra questo lavoro di tesi parte dall'assunzione che i word embedding siano in grado di rappresentare il senso delle parole e che dunque parole di significato simile si ritrovino vicine nello spazio vettoriale degli embeddings. Occorre però considerare che una parola può avere significati multipli che si attestano in frasi diverse, gli embeddings vanno quindi considerati nel contesto delle frasi in cui occorrono. In questo scenario significati distinti dovrebbero apparire in zone distinte dello spazio, e ciascun senso risultare 'vicino' a parole semanticamente simili. In modo conseguente variazioni di significato nel tempo dovrebbero riflettersi in vettori dislocati diversamente nello spazio. Come già osservato nel cap. 2, non è semplice individuare queste variazioni, la questione è complessa e dipende da più fattori concorrenti: dati, modello, strumento d'indagine. Se non se ne comprende la portata si arriva a conclusioni simili a quelle di Laicher et al. (2020), che affermano che gli embeddings contestuali ricavati dai transformer sono meno efficaci dei word embedding tradizionali ottenuti con Word2Vec.

Con queste premesse la tesi ha esaminato diversi approcci per risolvere i cambiamenti di significato, sia tecniche che misurano direttamente la distanza (ossia cercano una vicinanza) tra embeddings, sia quelle che utilizzando tecniche di clustering per individuare gruppi di sensi/usi di parole. La tesi ha

esplorato diverse tecniche di clustering e diverse misure di distanza da usare nel clustering, utilizzando contextual embeddings ottenuti da reti di Transformers, ma anche diverse metriche di distanza da applicare direttamente allo spazio degli stessi contextual embeddings. Nelle sperimentazioni condotte il modello di linguaggio è stato inoltre messo a punto (fine-tuned) su corpora di periodi temporali diversi e sono state svolti numerosi esperimenti in diverse configurazioni sui corpora italiani al fine di mettere a punto l'approccio. Il modello migliore è stato poi applicato anche ai dati forniti per DIACR-Ita, con l'obiettivo di confrontarsi con lo stato dell'arte. I risultati ottenuti hanno superato i migliori risultati ufficiali ottenuti nella competizione, mostrando l'efficacia della tecnica proposta e smentendo quindi il risultato negativo citato da Laicher et al. (2020).

Come è emerso dai vari contributi presentati a SemEval e DIACR-Ita, la strategia migliore è risultata l'utilizzo di più strumenti d'indagine sui dati. Più tecniche che lavorano sugli stessi dati e che concorrono alla produzione del migliore risultato. La combinazione di metodologie modulari, che si integrano per migliorare i risultati consente di valutare a fondo il fenomeno e soprattutto di adattare lo studio alle caratteristiche dei dati: contenuti, dimensioni, due o più intervalli temporali. Tale aspetto di flessibilità non deve mancare nell'implementazione di un metodo idoneo a questo compito. Le attività principali che hanno concorso alla realizzazione del metodo si possono riassumere in alcuni passi essenziali:

(i) *Analisi dei dati*: numero dei corpora, dimensioni, differenze dimensionali tra corpora, caratteristiche dei testi, formato. Dagli esperimenti fatti è risultato molto importante condurre un'analisi preventiva dei dati, in ragione del tipo di studio. Naturalmente questo richiede tempo e ha un impatto non trascurabile sull'organizzazione degli esperimenti, ma è un tempo speso bene se può aiutare a prendere le decisioni migliori. Per esempio nel Task DIACR-Ita l'analisi dei corpora ha mostrato come i corpora, rielaborati dal formato UDPipe[38] (utilizzando il modello ISDT-UD v2.5), presentassero

[38] http://lindat.mff.cuni.cz/services/udpipe/run.php

caratteristiche diverse dal corpus ILC-Ita1, pur essendo coevi (§ 6.1.3.2). Tali caratteristiche hanno reso consigliabile un fine-tuning del modello generale anche a parità di intervallo temporale. Una scelta che si è rivelata giusta visti poi i risultati.

(ii) *Pre-processing sui corpora*. Per esempio, per eseguire il task di DIACR-Ita e nelle altre sperimentazioni condotte nel lavoro di tesi, sarebbe stato sicuramente possibile fare come Martinc et al. (2020b), cioè scorrere tutto il corpus per estrarre le parole dalle frasi così come si presentavano nel corpus e, nel caso dell'impostazione di una selezione delle parole, inserire un filtro in grado di estrarre solo gli embeddings delle parole desiderate. Tuttavia esistono strategie più efficienti. Per esempio sono disponibili linguaggi di programmazione, ambienti di sviluppo e procedure consolidate che lavorano più rapidamente con i dati testuali e sono in grado di produrre estrazioni molto raffinate dei dati [Picchi et al., 2010], [Sassolini et al., 2010].

(iii) *Scelta del modello BERT più adatto agli obiettivi del lavoro*. Come hanno affermato a vario titolo i partecipanti a SemEval 2020 non esiste un modello specifico per l'attività di rilevamento semantico, le scelte sono lasciate all'esperienza e all'intuito del singolo. Come Martinc et al. (2020b) si è optato per il modello sviluppato per l'attività MLM, più adatto a studi della lingua. Per capire come questa sia una questione importante, basti pensare che il modello ELECTRA è finalizzato proprio a migliorare il metodo MLM di pre-allenamento.

(iv) *Partire dal modello pre-trained più grande*. È vero che i modelli multilingue sono più leggeri e consentono di produrre risultati ottimi per molte attività, ma l'applicazione di modelli BERT allo studio del cambiamento semantico è relativamente recente, non esiste ancora la consapevolezza di quale sia la strategia migliore per trattare il fenomeno e quale impatto abbiano i materiali di partenza sullo studio che si vuole condurre. In ragione di questo si è ritenuto un'attività importante condurre un fine-tuning su tutti i corpora da trattare. Non è servita una messa a punto spinta/profonda, è stato sufficiente, come sostenuto da Martinc et al. (2020b), condurre almeno 4/5

cicli completi (epochs) su tutti i dati, senza un ulteriore perfezionamento. Nel caso di dati di piccole dimensioni si è infatti osservato che questo può portare a problemi di poca 'generalizzazione', con conseguente rischio di overfitting.

(v) *Flessibilità della configurazione del metodo*. Solo dopo aver condotto le analisi sui dati è stato possibile scegliere gli approcci più idonei a studiare il fenomeno. Per esempio, per compiti come quello di DIACR-Ita, valutate le dimensioni delle attestazioni delle parole target in entrambi i corpora, era più opportuna una metodologia che tenesse conto delle dimensioni contenute dei dati. In generale l'approccio più efficace si è ottenuto studiando i risultati dell'applicazione di diverse configurazioni del metodo, date dalla composizione di diversi moduli software, e infine provando a produrre automaticamente quella più adatta al trattamento dei dati (§ 7).

(vi) *Generalizzazione del metodo*. Questo è un aspetto importante che ha visto interrogarsi molti degli autori che sono stati citati in questo lavoro di tesi. In realtà solo Martinc et al. (2020a), Montariol et al. (2021) e Giulianelli et al. (2020) hanno sollevato la questione nei loro articoli, ma non hanno saputo rispondere in modo esauriente alla questione. Martinc et al. definiscono il lavoro di Giulianelli sul clustering un proxy per il cambiamento dell'uso delle parole, ma poi ne sottolineano la mancanza di scalabilità. Salvo poi loro stessi usare un algoritmo di clustering come Affinity Propagation che utilizza la distanza tra grafi e che prevede un processo iterativo non facilmente prevedibile né intuitivo, in cui i punti si scambiano 'messaggi' fino a quando non si ottiene un insieme di rappresentanti di alta qualità. Un approccio dimostratosi efficace in dati di dimensioni contenute, ma non propriamente scalabile. Mediare tra scalabilità e interpretabilità del metodo è una caratteristica comune agli studi condotti su dati di grandi dimensioni. È probabile che con lo sviluppo di hardware sempre più potenti sia possibile scalare anche metodi complessi, è già stato dimostrato in vari campi di ricerca. Tuttavia, quando i fenomeni da studiare sono complessi e pieni di varabili come il cambiamento semantico, è più sensato adottare metodi semplici, la cui logica sia facilmente comprensibile, basati su algoritmi scalabili, di uso

consolidato e per i quali esiste una vasta gamma di personalizzazioni ed esperienze.

(vii) *Automatizzare i processi.* Come sostenuto da Tahmasebi et al. (2018), provare a rilevare automaticamente il cambiamento semantico è un passo importante, non solo perché è l'unico modo per saggiare la scalabilità dell'approccio, ma anche perché consente di evidenziare i punti critici, di valutare pro e contro delle scelte fatte. Questo è vero soprattutto in relazione alla necessità di adattare l'approccio alle proprie capacità di svilup-patore/utente. È infatti emerso dalle sperimentazioni fatte che l'applicazione di funzioni e strumenti già disponibili, senza nessuna customizzazione, è molto rapida, ma non sempre efficiente. Inoltre data l'ottimizzazione delle procedure messe a disposizione delle librerie software, nel momento in cui sorga l'esigenza di apportare modifiche, l'intervento può risultare pesante, a volte addirittura rischioso.

Nel lavoro di tesi si è quindi deciso di modulare il metodo proposto in strategie multiple e concorrenti, con l'obiettivo di rendere ogni modulo indipendente e utilizzabile in autonomia, ma consentirne l'integrazione a seconda delle necessità. I metodi individuati comprendono quindi:

1. *metodo senza aggregazioni (metodo 1)*; basato sul calcolo di metriche di distanza tra vettori, quali distanza del coseno, distanza euclidea, distanza di Hausdorff, APD con distanza del coseno e APD con distanza euclidea. Il metodo può ulteriormente caratterizzarsi in:
   1.1. metodo che utilizza gli embedding di tutte le occorrenze delle parole nei due corpora;
   1.2. metodo che utilizza la media degli embedding di ogni parola nei due corpora. Strategia riservata prevalentemente a dati di grandi dimensioni;
2. *metodo di clustering (metodo 2);* che utilizza gli algoritmi K-means e Dbscan. Anche questo metodo può ulteriormente suddividersi in:
   2.1. metodo che unisce i corpora in una sola analisi (C1+C2), che sarà definito '*metodo 2.1*', rifacendosi alla definizione data da Kanjirangat

et al. (2020). L'approccio è simile ma la procedura è modificata per riuscire a renderla scalabile. Per l'attribuzione di un nuovo significato, il conteggio di tutte le etichette di tutti i cluster proposto dagli autori, è stato sostituito da un controllo che impone una condizione di uscita nel caso si incontri un cluster con riferimenti di un solo corpus. La condizione è infatti quella minima per confermare la presenza di un nuovo senso;

2.2. metodo che esegue il clustering separatamente per C1 e C2. Come ulteriore strumento d'indagine si può aggiungere al metodo il modulo software che proietta poi i risultati del clustering applicato su un corpus su quelli del clustering applicato all'altro. Sempre seguendo quanto implementato dagli autori indicati al punto 2.1, ci si riferirà al metodo come 'metodo 2.2'. Questo tipo di configurazione è stata utile nei casi in cui l'applicazione degli altri metodi aveva prodotto risultati ambigui ed era necessaria un'analisi grafica più dettagliata; ne saranno mostrati uso e utilità nel capp. 7.4 e 8.

6.1.1 Il modello BERT per l'italiano

Seguendo i passi indicati da Tang et al. (2018) (§ 2), dopo i corpora e la definizione di un metodo generale per lo studio del fenomeno, si è passati alla scelta del modello di linguaggio (LM) alla base del metodo. Tra i modelli BERT disponibili si sono individuati quelli con le caratteristiche più adatte allo scopo: obiettivo del pre-addestramento ampio (non classificazione, question answering, sentiment, ecc.); modello pre-addestrato su dataset di maggiori dimensioni e di tipo 'large general-domain', (non linguaggio dei social, dominio medico, ecc.). Le scelte possibili da adottare per la lingua italiana erano quindi principalmente due: il modello BERT standard oppure quello ELECTRA[39]. La modellazione del linguaggio 'mascherato' propria di BERT è

[39] ELECTRA (*Efficiently Lgain an Encoder that Classizes Token Replacements Accurately*) è un nuovo approccio pre-allenamento che mira a eguagliare o superare le prestazioni a valle di

ampiamente utilizzata nell'elaborazione del linguaggio naturale per l'apprendimento delle rappresentazioni testuali e per le attività di modellazione del linguaggio. I metodi di pre-addestramento definiscono una sorta di mascheramento parziale delle parole per migliorare l'apprendimento. In pratica corrompono l'input sostituendo alcuni token con 'maschere' e quindi addestrano un modello per ricostruire i token originali. Questo approccio produce buoni risultati quando gli addestramenti vengono trasferiti in attività NLP a valle, ma generalmente richiede grandi quantità di calcolo per essere efficace. Un'alternativa è rappresentata dal modello ELECTRA [Clark et al., 2020] in cui è proposta un'attività di pre-addestramento più efficiente chiamata "*replaced token detection*". Invece di mascherare l'input, l'approccio lo corrompe sostituendo alcuni token con alternative plausibili, campionate da una piccola rete di generatori. Quindi, invece di addestrare un modello affinché predica le identità originali dei token corrotti, si addestra un modello discriminativo, in grado di predire se ogni token nell'input danneggiato è stato sostituito o meno da un campione del generatore. I creatori di ELECTRA affermano che le rappresentazioni contestuali apprese dal loro approccio superano sostanzialmente quelle apprese da BERT.

Dalle analisi condotte in letteratura vi è una predominanza nell'uso dei modelli BERT rispetto a ELECTRA, anche se si riscontra comunque una quantità minore di LM pre-addestrati sull'italiano, rispetto all'inglese per il quale la scelta è molto più ampia. Nel lavoro di tesi si è deciso quindi di partire dai primi, nel caso specifico, è stato eseguito l'addestramento sui dati mettendo a punto BERT per l'attività MLM, ossia specificamente il modello di linguaggio mascherato. L'obiettivo era utilizzare BERT come *feature extractor*: per qualsiasi sequenza di testo, le *features* estratte da BERT per la modellazione del linguaggio, possono essere utilizzate come rappresentazione contestualizzata dei token di input. Tra quelli pre-addestrati per la lingua italiana è stato scelto quello case sensitive (ossia nella configurazione che

un modello pre-addestrato BERT MLM utilizzando risorse dalla fase di pre-formazione. L'attività di pre-addestramento in ELECTRA si basa sul rilevamento dei token sostituiti nella sequenza di input. ELECTRA mira a migliorare il metodo MLM di pre-allenamento.

distingue tra lettere maiuscole e minuscole). La scelta della configurazione *~cased* non è condivisa da tutti, probabilmente ciò è dovuto al fatto che la stragrande maggioranza delle sperimentazioni utilizza la versione *~uncased*. È però da considerare che queste esperienze si riferiscono ad altre attività (classificazione, *summarization, question answering*, ecc.) e soprattutto riguardano prevalentemente la lingua inglese.

I modelli sono scaricabili dal repository dei dataset di Hugginface[40], nel caso di studio si è utilizzato il modello '*dbmdz/bert-base-italian-xxl-cased*', che ad oggi è quello di maggiori dimensioni. I creatori, il team MDZ Digital Library (dbmdz) presso la Bavarian State Library, hanno utilizzato un recente dump di Wikipedia e vari testi della raccolta di OPUS e li hanno estesi con i dati della parte italiana del corpus OSCAR. Pertanto il corpus di finale ha una dimensione di 81 GB e 13.138.379.147 di token[41].

6.1.2 L'ambiente software

Gli strumenti hardware e software utilizzati per lavorare con questi modelli sono stati un computer con GPU (*Graphic Processing Unit*) e un server linux con alcune unità GPU, utilizzato prevalentemente per le fasi di fine-tuning. Una GPU è un circuito elettronico nato per accelerare la creazione di immagini su un dispositivo di visualizzazione, ma oggi sono utilizzate ampiamente in questo campo. Le moderne GPU discendono dai chip grafici ma oggi, l'alto parallelismo supportato, permette di svolgere funzioni computazionali in modo più efficiente delle CPU. L'incremento del loro uso è dovuto alla crescente necessità di elaborare in modo parallelo grandi quantità di dati, necessità che caratterizza molti settori d'impiego come il *deep learning* e l'elaborazione di immagini, sino alle applicazioni utilizzate per la gestione del mercato azionario in borsa. Lo sviluppo di queste applicazioni richiede un ambiente di programmazione affidabile con librerie specifiche altamente

[40] https://huggingface.co/
[41] https://huggingface.co/dbmdz/electra-base-italian-xxl-cased-generator

ottimizzate. Le librerie CUDA[42] (*Compute Unified Device Architecture*) sono tra le più adatte perché costituite da strumenti e tecnologie che offrono prestazioni notevolmente superiori rispetto alle alternative in diversi ambiti applicativi, dall'intelligenza artificiale al calcolo ad alte prestazioni.

Volendo poi utilizzare il linguaggio Python, che rappresenta l'uso prevalente nello sviluppo di queste applicazioni, si è deciso di configurare un ambiente Jupyter Notebook per l'esecuzione degli applicativi. Jupyter Notebook è un'applicazione Web open source che permette di creare codice sorgente eseguibile lavorando in modalità client dal proprio computer. Il software sviluppato in python viene eseguito sul server, dove risiede e dove sono disponibili maggiori capacità di calcolo grazie alla GPU[43]. Con queste premesse si sono installate le librerie *Transformers*[44] con cui sviluppare i modelli e implementare i moduli software per condurre i vari esperimenti. Sono disponibili diverse implementazioni di tali reti Transformers: nella versione originale sviluppata in PyTorch[45] oppure nella versione TesorFlow[46]. Nel lavoro di tesi si è optato per la versione sviluppata in PyTorch. L'elemento caratterizzante di questo framework sono i tensori, matrici multi-dimensionali di numeri, su cui si fondano anche le librerie NumPy e buona parte del calcolo scientifico in Python. Ciascuno di questi framework offre agli utenti gli elementi costitutivi per la progettazione, l'addestramento e la validazione di reti neurali profonde attraverso un'interfaccia di programmazione di alto livello. PyTorch è diventato un framework molto popolare perché possiede due caratteristiche chiave. In primo luogo, è ottimo nel calcolo e gestione dei tensori con GPU. In secondo luogo, PyTorch supporta nativamente CUDA, a

[42] Si tratta di un'architettura hardware per l'elaborazione parallela creata da NVIDIA produttore di GPU. Tramite l'uso delle librerie software disponibili per l'uso di CUDA, i programmatori di software possono implementare applicazioni capaci di eseguire calcolo parallelo sulle GPU delle schede video NVIDIA.
[43] Jupyter Notebook è anche configurabile sul proprio computer attraverso l'uso di piattaforme con Anaconda che simulano l'interazione client – server.
[44] https://huggingface.co/docs/transformers/index
[45] PyTorch è un framework DL open source Python
[46] https://www.tensorflow.org/resources/models-datasets

differenza di TensorFlow che consente di addestrare i modelli utilizzando le librerie Keras e un metodo di adattamento.

6.1.3 Strategie di *fine-tuning*

L'analisi della letteratura (§ 4) ha evidenziato una situazione molto articolata che testimonia l'esistenza di approcci diversi, non solo legati a attività a valle diverse ma, anche riguardo alla stessa attività, esistono implementazioni differenti, che non facilitano lo scelta delle migliori strategie d'uso di questi modelli. Erano presenti in SemEval interpretazioni del fine-tuning molto specializzate che si rifacevano al lavoro di Arase e Tsujii (2019). Il loro lavoro mirava ad aumentare la conoscenza semantica in BERT, mettendo a punto il modello pre-addestrato sui dati di parafrasi e poi perfezionando nuovamente il modello per le attività correlate d'identificazione della parafrasi e valutazione dell'equivalenza semantica. Gli autori riportavano risultati che dimostravano prestazioni migliorate rispetto a un modello non esposto ai dati di parafrasi. Tra i lavori descritti molti non hanno ritenuto fondamentale questa pratica e hanno utilizzato prevalentemente modelli pre-addestrati multilingue. Kutuzov e Giulianelli (2020) hanno invece dedicato tempo a questa attività e articolato gli interventi a seconda della lingua. Per inglese, tedesco e svedese, hanno utilizzato i rispettivi modelli specifici per lingua: *bert-base-uncased*, *bert-base-german-cased* e *af-ai-center/bert-base-swedish-uncased*. Per il latino invece hanno utilizzato bert-base-*multilingual-cased*, poiché non ancora disponibile un BERT specifico per il latino. Hanno poi eseguito il fine-tuning con la configurazione standard di BERT per due epochs mentre per l'inglese l'addestramento ha raggiunto le cinque epochs. Martinc et al. (2020b) per ogni lingua hanno condotto fine-tuning del modello per cinque epochs sui corpora del task, come da loro sperimentato in precedenza [Martinc et al., 2020a]. Hanno utilizzato la configurazione *Masked Language Model* nella fase di messa a punto [Devlin et al. 2019], perfezionando il modello per aumentare la qualità delle rappresentazioni contestualizzate, ma non hanno condotto alcuna

messa a punto diacronica del modello. Gli autori hanno affermato che questo passaggio non fosse necessario a causa della natura contestuale delle immersioni generate dal modello.

Per quanto riguarda la lingua italiana, in riferimento a quanto utilizzato in DIACR-Ita, nella maggioranza dei casi i partecipanti al task hanno ritenuto che non servisse un fine-tuning o messa a punto per i modelli pre-addestrati. Wang et al. (2020) hanno lamentato che, nell'attività di rilevamento del cambiamento semantico, la messa a punto nelle attività a valle è attualmente impossibile perché le etichette annotate non sono sufficienti a questo scopo. Invece Laicher et al. (2020) hanno ottimizzato i parametri e il sistema sul set di dati per l'inglese utilizzato in SemEval, concludendo, dai loro esperimenti, che fosse possibile raggiungere prestazioni elevate anche senza messa a punto. Hanno quindi evitato il fine-tuning del modello pre-trained sui corpora del task, ritenendola un'attività che necessitava di tempo e i cui benefici non erano assodati. Analizzando i risultati non ottimali prodotti per l'italiano da questi modelli, si è ritenuto invece importante produrre un fine-tuning su ogni corpora utilizzato per il lavoro di tesi. Anche le analisi post-task che i diversi gruppi hanno fatto, confermano risultati inferiori alle aspettative. Come Martinc et al. (2020a) per la messa a punto è stata utilizzato BERT per l'attività MLM.

L'uso dei modelli BERT richiede la disponibilità di un hardware in grado di supportare il calcolo richiesto. Per come sono concepiti si tratta di gestire porzioni ampie di dati organizzate in strutture dati (tensori ~ matrici) molto impegnative dal punto di vista dell'elaborazione. Il motivo è legato all'architettura dei Transformers; il blocco *multi-head attention* è la principale innovazione alla base di tali modelli e permette alla rete di organizzarsi e scegliere su quali parti del testo concentrarsi. Tutte le parole di una frase vengono elaborate nel Transformer contemporaneamente, superando il limite di elaborazione sequenziale dei dati, presente nei modelli precedenti di reti neurali ricorrenti (RNN), come *Long Short-Term Memory* (LSTM) o *Gated Recurrent Units* (GRU). Per poter tener conto della posizione dei termini in

una frase, i transformer aggiungono un vettore di codifica della posizione agli embeddings di ciascun token. I vantaggi del loro uso includono la modellazione più efficace delle dipendenze a lungo termine tra i token in una sequenza temporale, e l'addestramento più efficiente del modello in generale, eliminando la dipendenza sequenziale dai token precedenti.

Come indicato nel cap. 6.1.1 siamo partiti da un modello pre-addestrato per l'italiano che utilizza un enorme set di dati e consente di evitare di partire con l'addestramento da zero. Il modello pre-addestrato è già in grado di trattare la lingua italiana e possiede molte conoscenze su di essa. È lecito quindi domandarsi se sia necessario un ulteriore perfezionamento sui dati sottoposti ad indagine. In generale si compie questo passaggio quando si vuole perfezionare il modello su un'attività specifica, con l'obiettivo di rendere più facile l'adattamento. Leggendo i diversi contributi dei partecipanti al task DICR-Ita non è emersa una posizione univoca in merito, ma come verrà illustrato nel capp. 6.2.3.1. e 6.2.3.2, ragioni legate alla peculiarità dei dati ne hanno suggerito l'uso. Il fine-tuning o messa a punto del modello è una tecnica che mira ad ottimizzare i modelli pre-addestrati. Si tratta in genere dell'ulteriore perfezionamento del modello pre-addestrato, condotto per alcuni cicli completi (epochs) sul set di dati specifici per l'attività a valle desiderata. Sebbene la messa a punto di BERT sia relativamente semplice in teoria, può richiedere molto tempo e può riservare incognite quando si ottimizza un modello su piccoli set di dati

Produrre l'addestramento dei dati è quindi un'operazione complessa che è resa facile solo dalla disponibilità di funzioni apposite della stessa libreria che rilascia il software. La fase di addestramento consiste così nel preparare i dati nel formato atteso e settare gli iperparametri della funzione dedicata. Di norma un addestramento con supervisione dei risultati prevede l'utilizzo di una porzione di dati riservata per la valutazione (una porzione che tipicamente va dal 20% al 30% dei dati) e un meccanismo di arresto per evitare l'overfitting dei dati. Esiste anche la possibilità di produrre un addestramento in cui tutti i dati sono sottomessi al modello senza alcun controllo sui risultati. Nel caso

specifico del lavoro di testi si è trattato di condurre due diverse fasi di fine-tuning: per condurre il task DIACR-Ita (§ 7.3.1), per lo stesso esperimento con i dati ILC-Ita (§ 7.4).

6.1.3.1 *Fine-tuning* con i dati ILC-Ita

Il fine-tuning nel caso del corpus ILC-Ita aveva l'intenzione di permettere al grande modello pre-addestrato sull'italiano recente, di acquisire conoscenze su una varietà della lingua relativamente lontana nel tempo. Come descritto nel cap. 5.1 la parte più antica (ILC-Ita3) del corpus risale in gran parte al periodo pre-unitario, quando l'italiano non era ancora una lingua condivisa e 'I Periodici Milanesi' da cui è estratto il corpus ne risentono. Fu infatti solo dopo l'unità che nacque la volontà di riconoscersi in una lingua condivisa, ne discusse anche Luigi Settembrini in 'Ricordanze della mia vita' (ed. a cura di M. Themelly, Milano 1961), che la collegava 'al ridestarsi di un comune sentimento identitario, a séguito di tre secoli di servaggio a garanzia dello straniero dominatore'. È dello stesso periodo la proposta di Manzoni di ricorrere al solo fiorentino colto, per ottenere l'unità linguistica. Queste ragioni hanno portato alla scelta di un'ottimizzazione del modello su quei dati, costituita da una messa a punto di poche epochs ma sufficiente a rendere il modello in grado di 'addattarsi' alle peculiarità dei dati.

La messa a punto di modelli linguistici pre-addestrati basati su BERT è diventata una pratica comune in vari compiti di NLP e nonostante le buone performance, rimane un processo instabile: addestrare lo stesso modello con impostazioni casuali può comportare una grande variazione delle prestazioni. La letteratura [Devlin et al., 2019; Lee et al., 2020; Dodge et al., 2020] ha identificato due potenziali ragioni per l'instabilità osservata: 'dimenticanza catastrofica'[47] e piccole dimensioni dei set di dati di fine-tuning. Mosbach et

[47] Per utilizzare questi modelli in domini con dati di addestramento limitati, uno degli approcci più efficaci consiste nel pre-addestrarli prima su dati di grandi dimensioni fuori dominio e quindi ottimizzarli con i dati target limitati. Il problema è che dopo la messa a punto dei

al. (2021) affermano che l'instabilità osservata è causata dalla difficoltà di ottimizzazione, che porta a gradienti evanescenti. Inoltre, sostengono che la variazione delle prestazioni dell'attività a valle può essere attribuita a differenze di 'generalizzazione', condizione in cui ci si trova quando si hanno modelli perfezionati con stessi valori della *loss*[48] di training, ma con prestazioni sui dati di test notevolmente diverse.

Il corpus che ha richiesto un addestramento più articolato è stato ILC-Ita3, ossia quello più antico cronologicamente. Questo milione circa di token era certamente il più distante dal testo sul quale è stato pre-addestrato il modello BERT. Per questo motivo sono stati condotti sui dati due distinti addestramenti: uno utilizzando l'API (*Application Programming Interface*) Trainer messa a disposizione dalle librerie Transformers, l'altro implementando in proprio la procedura di training. L'API Trainer è ottimizzata e configurabile impostando gli iperparametri desiderati, ma non c'è modo di 'sindacare' sulle modalità di training.

L'obiettivo dell'ottimizzazione del modello era quella di renderlo in grado di gestire/comprendere una varietà d'italiano particolare. L'attività a valle di questo compito era infatti la specializzazione della lingua in una prospettiva storica. L'adattabilità cercata non era facilmente misurabile ma è stato possibile osservare quanto l'apprendimento sui dati ILC-Ita migliorasse, andando a valutare la performance della procedura, ovvero i valori di loss misurati dalla procedura di training. Le varie prove fatte e le configurazioni sperimentate hanno fissato i migliori risultati di *validation_loss* (loss misurata sul dataset di valutazione) a 2.139, con un valore di Perplexity finale 8.65[49]. La

modelli questi tendono a funzionare male nel dominio di origine, fenomeno noto come 'dimenticanza catastrofica'.

[48] In modo sintetico con loss (perdita) si indica quanto sia distante la previsione del modello dai dati. Se la previsione del modello è perfetta, la loss è zero. L'obiettivo dell'addestramento di un modello è trovare un insieme di pesi che abbiano in media una bassa loss su tutti i dati.

[49] La *perplexity* PP di un modello linguistico su un test set è l'inverso della probabilità media (geometrica) assegnata a ciascuna parola nel test set dal modello. Questo indice è teoricamente elegante poiché il suo logaritmo è un limite superiore del numero di bit per parola previsto nella compressione del testo (nel dominio) utilizzando il modello misurato. Sfortunatamente, mentre i modelli linguistici con *perplessity* inferiori tendono ad avere tassi di errore inferiori,

validation_loss è una misura più precisa dell'accuratezza in casi come questo, infatti la seconda è più indicata quando è necessario determinare una soglia che fissa l'appartenenza ad una classe o meno. Se si cerca di prevedere la risposta corretta, ma con meno fiducia, la validation_loss la catturerà, mentre l'accuratezza no. Con l'API Trainer e rispetto agli altri corpora addestrati, il risultato ottenuto su ILC-Ita3 ha prodotto valori della performance inferiori. Per esperienza nell'uso di questo tipo di strumenti, valori bassi di loss garantiscono un buon risultato, ma dalla letteratura disponibile non si comprende quale valore sia accettabile, si lascia all'utilizzatore l'onere di capire se, per il peculiare scopo, si può ritenere soddisfatto. Esistono infatti molte variabili che possono produrre cambiamenti significativi nelle prestazioni del sistema.

Nel nostro caso, la necessità di capire come funzionava il modello e provare ad "aprire la scatola", ha spinto le sperimentazioni verso un'implementazione in proprio della procedura di addestramento. Il problema principale era capire come calibrare il tasso di apprendimento (*learning rate*) e le condizioni di uscita (tecnica *earlystopping* + *patience*)[50]. La procedura implementata in proprio si è rivelata più flessibile, potendo impostare combinazioni di learning rate progressivo, dimensioni di batch e numero di epochs più artigianali. Infatti nella procedura prevista dall'API Trainer la funzionalità di arresto automatico permette di impostare l'uscita con validation_loss non decrescente, ma è legata all'iterazione degli *step* (i cicli più interni del processo di training), rendendo la gestione per *epochs* (i cicli esterni, sull'intero set di dati) meno controllabile. La procedura realizzata, intenzionalmente più lenta (ha richiesto più iterazioni ma aveva anche una

esistono numerosi esempi in letteratura in cui i modelli linguistici che forniscono un grande miglioramento di questo indice rispetto a un modello di base, hanno poi prodotto un miglioramento minimo o nullo nell'errore.

[50] Questa tecnica è comunemente utilizzata nell'addestramento di reti neurali, che prevede di fissare una condizione di uscita dal ciclo di addestramento. Ovvero l'addestramento termina quando le prestazioni sul valore di loss misurato sul set di validazione non diminuisce più. Il valore *patience* indica il numero di iterazioni senza miglioramento che si attendono prima di terminare l'addestramento.

finestra di batch diversa) e con un learning rate flessibile, ha prodotto un modello in cui si è raggiunto un valore inferiore di validation_loss.

Epochs	Learning rate	time	Training loss	Validation loss	average prediction
1/6	4.2e-05	366s	2.3097	2.1262	53.15%
2/6	3.3e-05	728s	2.1258	2.1237	53.09%
3/6	2.5e-05	1093s	2.1121	2.1236	53.09%
4/6	1.7e-05	1455s	2.1012	2.1252	53.15%
5/6	8.3e-06	1819s	2.0930	2.1263	53.15%

Tabella 3: risultati del training del corpus M1.1 con procedura ad hoc

Il fatto che il modello così addestrato abbia prodotto una performance migliore non è di per sé una garanzia di un migliore adattamento. Per questo motivo sono state condotte prove di diverso tipo sull'uso del modello che vanno dalla prova sul campo dei vari metodi, allo studio sul pooling dei livelli (§ 6.1.4). La differenza tra i valori finali di val_loss tra il training con procedura standard e quelli registrati dalla procedura ad hoc dicono qualcosa di significativo sulla peculiarità del dato trattato. In altri termini sembra confermato quanto affermano Mosbach, Andriushchenko e Klakow (2021) che la strategia predefinita di regolazione 'fine'[51] raccomandata da Devlin et al. (2019), non funziona bene. Gli autori hanno osservato che l'addestramento su un set di dati di dimensioni contenute influisce effettivamente sulla varianza della regolazione 'fine', in particolare risultano molte più esecuzioni non riuscite.

In un secondo momento sono stati prodotti gli altri modelli fine-tuning (ILC-Ita1, EuroP) necessari per le varie prove implementate e per le analisi e i confronti sulle scelte chiave. Per questi dati, in cui è presente un italiano attuale, l'addestramento prodotto con le funzioni standard della libreria ha dato subito buoni risultati, non si è quindi utilizzata la procedura *ad hoc* di

[51] Ossia perfezionare BERT utilizzando un tasso di apprendimento fissato al valore 2e−5, tale tasso viene aumentato linearmente da 0 a 2e−5 per il primo 10% delle iterazioni, noto come 'riscaldamento', e successivamente ridotto linearmente a 0.

training. In totale, oltre al modello pre-trained di partenza sono stati prodotti i seguenti modelli:

- M1.1: fine-tuning utilizzando il corpus ILC-Ita3 nella versione "my_train" (implementazione della funzione di *train* in proprio)
- M1.2: sempre fine-tuning col corpus ILC-Ita3 ma nella versione "API Trainer" (con uso delle librerie Hugginface per il train)
- M2: fine-tuning utilizzando il corpus ILC-Ita1 nella versione "API Trainer" (con uso delle librerie Hugginface per il train)
- M3: fine-tuning utilizzando il corpus europarl-v7 (EuroP) di dimensioni confrontabili con gli altri corpora, utilizzato per ulteriori confronti e analisi.

Corpus ILC-Ita3 (M1.2)

Step	Training Loss	Validation Loss
100	2.391400	2.188614
200	2.338700	2.161887
300	2.226100	2.180398
400	2.214500	2.138744
500	2.166300	2.166889
600	2.142900	2.169407
700	2.130200	2.187521

Perplexity = 8.65

Corpus ILC-Ita1 (M2)

Step	Training Loss	Validation Loss
500	1.726300	1.643276
1000	1.715900	1.653849
1500	1.662800	1.654732
2000	1.654200	1.642167
2500	1.602100	1.631311
3000	1.584900	1.613770
3500	1.570300	1.615069

Perplexity = 5.00

Corpus EuroP (M3)

Step	Training Loss	Validation Loss
500	1.375600	1.301990
1000	1.293200	1.298312
1500	1.229700	1.261826
2000	1.234600	1.254122
2500	1.183000	1.236977

Perplexity = 3.44

Figura 2: schermate delle fasi di addestramento dei relativi corpora

In figura 2 sono mostrate le sintesi delle sessioni di fine-tuning di M1.2, M2 e M3, esecuzioni fatte utilizzando la stessa procedura mutuata dall'API Trainer, ma con aggiustamenti e piccole variazioni nei valori degli iperparametri. In realtà come visto nei lavori descritti (§ 4.1) non era necessario condurre una validazione dell'addestramento, soprattutto per le varietà di italiano più recenti come in M2 e M3. Tuttavia si è ritenuta una controprova significativa delle difficoltà del perfezionamento dell'ultimo strato di questi modelli, provando a confutare le affermazioni di Mosbach riportate in precedenza.

6.1.3.2 *Fine-tuning* con i dati DIACR-Ita

Per condurre la sperimentazione del task si è deciso di produrre un fine-tuning anche per i due corpora C1 e C2 messi a disposizione dagli organizzatori. In questo caso si trattava di selezioni ragionate e sincroniche estratte dal corpus UNITA. La particolarità risiedeva nel fatto che, essendo ormai concluso ufficialmente il task, è stato possibile reperire solo i corrispondenti corpora annotati. Questo ha richiesto un pre-processing dei dati, costituito da un applicativo software di parsing che ricostruisse il testo *flat* a partire dalla versione annotata.

	Corpus	Period	#Tokens
C1	L'Unità	1948-1970	52,287,734
C2	L'Unità	1990-2014	196,539,403

Tabella 4: dati ufficiali sui corpora C1 e C2 con n. di token

Entrambi i testi ottenuti sono risultati però diversi dall'originale poiché l'annotazione linguistica eseguita con UDPipe ne ha di fatto sciolto tutte le preposizioni articolate e le enclitiche, producendo frasi del tipo: "*Reazioni contenute in Spagna, che a il contrario di la Gran Bretagna…*" oppure "*… da il presidente il sistema di le « picconate », considerando lo l' unico idoneo…*". Tale caratteristica avrebbe potuto rendere l'adozione del solo modello pre-trained rischiosa, per tali ragioni e in considerazione di quanto già fatto per i corpora ILC-Ita si è prodotto un fine tuning sia per C1 (costituito da 2,754,329 frasi) che per C2 (6,960,632 frasi). I dettagli sulle dimensioni dei corpora sono mostrati in tabella 4. Date le dimensioni notevoli di C2 si è deciso di optare solo per una messa a punto di sole 3 epochs ed esclusivamente con le funzionalità ottimizzate dell'API Trainer. L'addestramento del modello su C1 è stato impostato in modo più flessibile, sempre utilizzando API Trainer, ma impostando un numero di epochs maggiore, valutando ad ogni ciclo completo quanto migliorassero i valori di val_loss e infine provando in parallelo l'uso del modello addestrato nelle sperimentazioni. Questa differente attenzione era dovuta sia alla presenza di un intervallo di tempo più lontano nel tempo, sia

alla necessità di comprendere come questa differenza nella sintassi, avrebbe influito nell'apprendimento da parte del modello. Si registra un peggioramento dell'apprendimento che però, con fasi lunghe di training, migliora. La sintesi del processo è mostrata in figura 3.

Fine-tuning Corpus C1

Step	Training Loss	Validation Loss
500	2.618400	2.435795
1000	2.511600	2.384766
1500	2.511200	2.352230
2000	2.458700	2.334169
2500	2.450700	2.313931
3000	2.429200	2.290524
....		
7000	2.328500	2.216005
7500	2.329100	2.214739
8000	2.331500	2.205575
8500	2.305800	2.194465
9000	2.297900	2.192404
9500	2.309100	2.185804

Figura 3: schermata dell'addestramento del corpus C1 di DIACR-Ita

6.1.4 Strategia di pooling per la scelta dei livelli

Il taglio sperimentale del lavoro di tesi ha imposto di sottoporre ogni decisione ad una fase di testing. Avendo un numero di variabili da controllare non trascurabile, è sempre buona norma non affidarsi a decisioni d'istinto, anche se spesso le intuizioni posso rivelarsi giuste è necessario sottoporle a vaglio, prima di considerarle buone. Si è deciso di eseguire tutte le sperimentazioni estraendo sia l'ultimo livello di hidden, che la media degli ultimi quattro livelli. Una sessione dedicata di esperimenti è stata dedicata a questo specifico problema. Si è trattato di:

- individuare un insieme $W_n = \{w_1,...w_n\}$ di parole target da analizzare.
- Reperire un insieme di sinonimi per ogni parola target (vedi tabella 5):
  - Produrre l'estrazione degli embeddings in 'contesto vuoto' da più modelli:
    - M1 = M1.1, cioè modello addestrato con procedura implementata ad hoc su dati ILC-Ita3;
    - M2 = il modello addestrato con dati ILC-Ita1;

- M3 = il modello addestrato con dati EuroP;
- M4 = M1.2, sempre dati ILC-Ita3 ma con fine-tuning condotto con API Trainer

Parola target	*Sinonimi*
applicazione	trattamento, elaborazione, programma, attuazione, realizzazione, cura, attenzione, ornamento, decorazione, apposizione
mercato	commercio, vendita, scambio, prezzo, baratto, traffico, domanda, offerta, contrattazione, confusione, mercimonio
disco	cerchio, anello, tondo, oggetto, astronave, vinile, registrazione, incisione, dischetto
registrazione	rilevazione, inventario, censimento, incisione, riproduzione, regolazione, taratura, iscrizione

Tabella 5: set di parole e sinonimi sui quali è stato condotto lo studio

– Individuazione di una metrica di confronto per gli embeddings:
  - Distanza euclidea. Altre distanze possono essere utilizzate come: Average Geometric Distance [Kutuzov et al., 2020], Canberra distance [Lance and Williams, 1966] o la Hausdorff distance [Rockafellar et al., 2009], La scelta non è neutrale ma la più usata resta la distanza euclidea.
– Implementare una sperimentazione che potremmo definire 'orizzontale' in cui per ogni modello (uno alla volta) sono stati estratti gli embeddings in 'contesto vuoto' di ogni parola target e dei suoi sinonimi e sono stati analizzati. In sintesi l'iterazione più esterna è fatta sui modelli:
  - Le distanze euclidee 'interne', vale a dire sono stati confrontati fra loro gli embeddings della parola target e di ogni sinonimo del gruppo.
  - Le distanze euclidee 'esterne', cioè tutti gli embeddings di tutte le parole e sinonimi sono stati confrontati tra loro.
– È stata implementata anche un'analisi 'verticale' in cui si analizza un gruppo di sinonimi e relativa parola target in tutti i modelli. In questo caso l'iterazione più esterna è stata fatta sui set di sinonimi.

Sia l'analisi 'orizzontale' che quella 'verticale' sono state ripetute una volta estraendo gli embeddings dell'ultimo livello di hidden, un'altra estraendoli

dalla media degli ultimi quattro livelli di hidden. Dall'analisi di questi dati è emerso che i risultati migliori si sono ottenuti con la seconda configurazione.

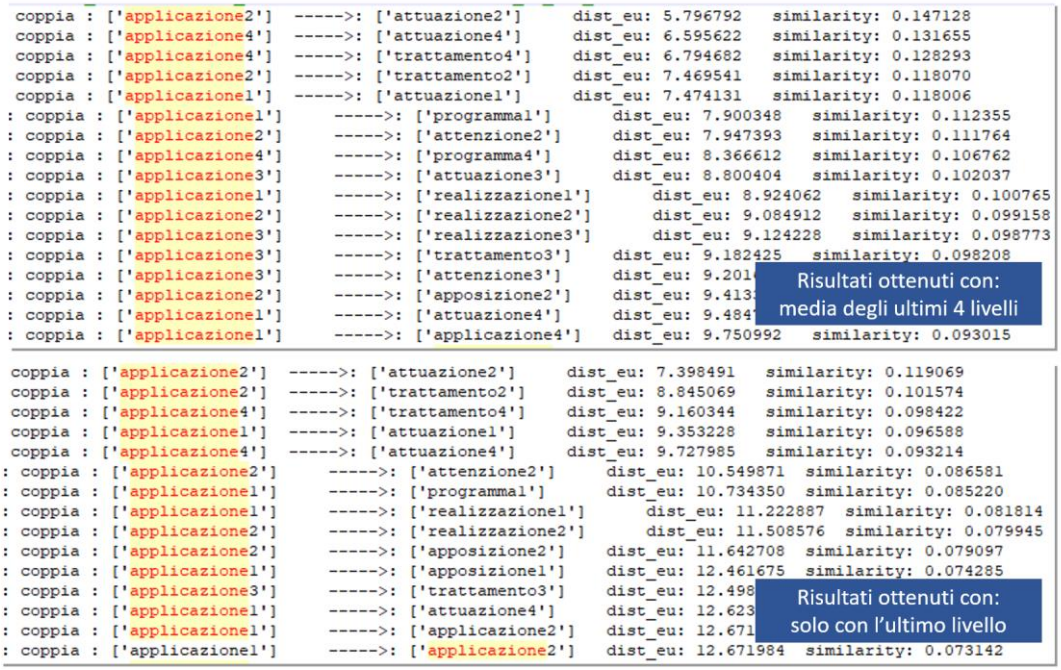

Figura 4: confronto delle distanze tra embeddings del set di sinonimi della parola 'applicazione', secondo la modalità 'verticale' e relativamente a tutti i corpora

In realtà la differenza più evidente si attesta con i dati più lontani cronologicamente. Questo ad ulteriore conferma di una complessità maggiore nel trattamento dei dati diacronicamente più lontani. Come mostrato nelle figure 4 e 5, le distanze euclidee tra sinonimi della parola *applicazione*, estratti da tutti i modelli implementati (indagine definita verticale) sono minori nei modelli che estraggono la media degli ultimi quattro livelli, indipendentemente dal corpus su cui è stato addestrato il modello. Nella figura 4 si distinguono i modelli diversi dal numero che segue ogni parola (applicazione1 = parola *applicazione* estratta dal modello M1).

Lo stesso risultato si ha per l'indagine definita orizzontale. Anche in questo caso, benché non vi sia differenza nella sequenza delle coppie di parole più simili, la distanza ha un valore minore quando si estraggono gli ultimi quattro livelli e se ne calcola la media.

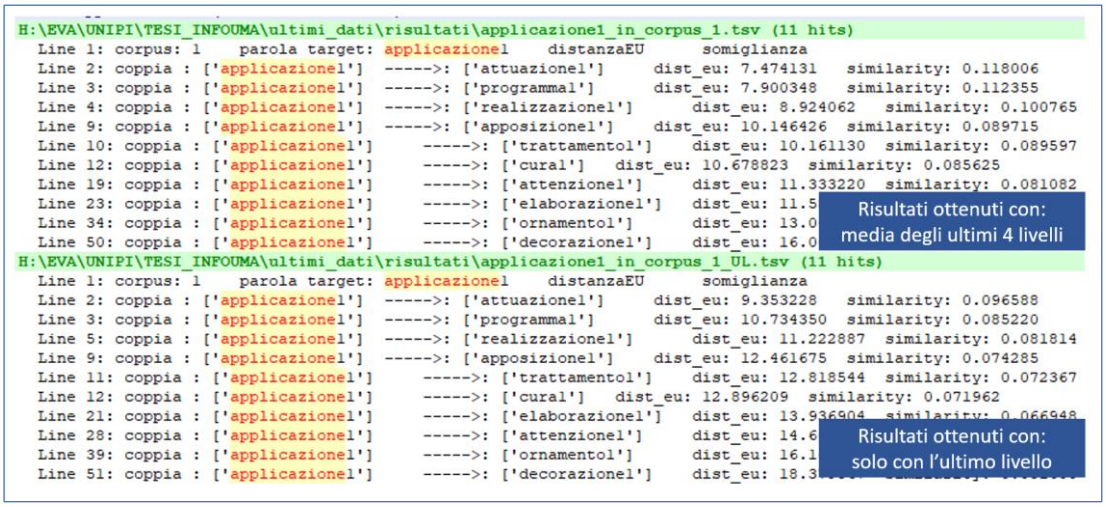

Figura 5: confronto delle distanze tra embeddings del set di sinonimi della parola 'applicazione', secondo la modalità 'orizzontale' e relativamente al corpus M1

Questo comportamento non garantisce la bontà della scelta ma suggerisce una maggiore facilità a catturare parole semanticamente vicine, soprattutto nell'ottica di un'indagine sperimentale, in cui gli elementi che evidenziano relazioni più connotate sono senz'altro più utili.

La scelta dell'estrazione dei word embeddings in contesto vuoto può rappresentare una strategia non convenzionale visto che queste rappresentazioni nei modelli BERT sono pensate proprio per l'estrazione in contesto di frase. Tuttavia quando esiste la necessità di valutare un comportamento che si potrebbe definire 'assoluto', eliminando cioè tutte le componenti casuali e contingenti, che potrebbero falsare i risultati della valutazione, è una pratica sperimentale ammessa. In questo lavoro vi si è ricorsi in più occasioni sempre con l'intento di provare ad entrare nel merito degli strumenti e del loro funzionamento. Alla luce dei risultati incrociati riproposti per tutte le parole e relativi sinonimi estratti da tutti i modelli implementati, si è deciso di utilizzare la media degli ultimi quattro livelli come strategia di pooling.

6.1.5 Il metodo in passi

I moduli che costituiscono il metodo sono di diversa natura e complessità. Le sperimentazioni condotte sono state molte e diversificate: dal fine-tuning con procedura ad hoc, alla scelta del pooling dei livelli, alla produzione di rappresentazioni grafiche dei dati. Il cuore del lavoro è costituito però dal metodo di indagine, che è stato differenziato in moduli singoli componibili e integrabili secondo le necessità degli esperimenti condotti. Con uno sforzo di sintesi si è voluto presentare tutto il metodo in un unico flusso semplificato, che, partendo dai corpora C1 e C2, conduce l'analisi di tutte le parole sottoposte a indagine. In questo modo si possono individuare i principali passi di cui si compone e che sono riassunti come segue:

- download delle librerie software utilizzate
- download dei modelli pre-addestrati M1 e M2 (*poiché è stato adottato il fine-tuning si sono creati modelli ottimizzati per ognuno dei corpora in esame*)
- recupero dei corpora di frasi pre-elaborati F_{C1} e F_{C2} (*rispettivamente estratti da C1 e C2*)
- per ogni parola target:
    - estrazione degli embeddings (di F_{C1} da M1 e rispettivamente F_{C2} da M2)
        - singolarmente di C1 e C2
        - complessivi di C1+C2
    - calcolo di tutte le metriche scelte per la misura delle distanze a coppie degli embeddings: $d = (emb_{C1}, emb_{C2})$[52]
        - dove d è: APD_{cos}, ADP_{eu}, HD, distanza euclidea
    - applicazione agli embeddings estratti dei diversi algoritmi di clustering:
        - Dbscan
            - Ricerca della configurazione ottima
            - Elaborazione degli embeddings con produzione dei cluster:
                - Separatamente su embeddings di C1 e C2
                - Unica di tutti gli embeddings (C1+C2)
        - K-means
            - Ricerca della configurazione ottima

[52] In realtà sono state misurate anche le distanze 'interne' degli embeddings, ossia si è misurata la distanza media di tutti gli embeddings in C1 e quella degli embeddings in C2 per poter confrontare i 'rappresentanti medi' per ogni intervallo di tempo.

- Elaborazione degli embeddings con produzione dei cluster
    - Separatamente su embeddings di C1 e C2
    - Unica di tutti gli embeddings (C1 + C2)
- Produzione della/e rappresentazione/i grafica/che a richiesta
- Salvataggio dei dati in vari formati

- Produzione di dati di confronto

Le operazioni indicate nel flusso non sono esaustive, ma sono sufficienti a comprendere il trattamento a cui sono stati sottoposti i dati. Molti moduli software realizzati presentano funzioni/varianti implementate in proprio per molte delle attività indicate: gestione dei corpora di frasi, estrazione degli embedding; ricerca degli iperparametri, produzione dei plot grafici, ecc.

6.2 Estrazione dei *word embeddings*

Con il termine *word embedding* (WE) si identificano rappresentazioni vettoriali dense di parole in uno spazio vettoriale multi-dimensionale. Il primo articolo che ne tratta si deve a Collobert et al. (2011), che le introduce descrivendo una rete neurale per l'apprendimento di vettori di parole, seguito poi dal modello di WE Word2vec di Mikolov et al. (2013). Da quel momento i word embedding sono stati ampiamente utilizzati per quasi tutti i task di NLP. Il loro sviluppo va di pari passo con i progressi della ricerca generale sull'apprendimento profondo, la loro efficacia sta infatti nella possibilità di modellare l'importanza semantica di una parola in forma numerica e quindi l'opportunità di eseguire operazioni matematiche su di essa. Per estrarre queste rappresentazioni dal modello occorre capire quale output viene prodotto dalla rappresentazione che implementano i Transformers di queste reti. Si tratta di un oggetto definito *hidden states* che viene restituito quando nei parametri di input si imposta il parametro *output_hidden_states*=True. Hidden states è costituito da una tupla di tensori della forma (*batch_size, sequence_length, hidden_size*), dove *batch_size* determina quante frasi

vengono elaborate in parallelo prima di eseguire l'aggiornamento interno dei pesi della rete; *sequence_length* è la massima lunghezza delle frasi ammessa dal modello, e *hidden_size* è il numero di unità nascoste/funzioni della rete. Nella configurazione del modello adottato si hanno 13 livelli (1 livello di input + 12 BERT layers), con sequence_length = 512 e hidden_size= 768. Dato quindi un token la sua rappresentazione di input viene costruita componendo *token embedding*, *segment embedding* e *position embedding*. Questa rappresentazione è chiamata *initial embedding output* e corrisponde all'indice 0 della tupla hidden states. I restanti 12 elementi nella tupla contengono l'output del livello nascosto corrispondente. Ad esempio: l'ultimo livello nascosto si trova all'indice 12, che è il 13° elemento della tupla.

Su ogni frase sottomessa al modello viene quindi eseguita la tokenizzazione BERT che si basa sul sistema WordPieces [Wu et al., 2016]. Una sequenza di immersioni (WE) viene generata per ciascuna di queste frasi/sequenze per i livelli di output dell'encoder. Una buona strategia di pooling dei livelli deve stabilire quali sono quelli più idonei alle necessità dell'attività a valle sulla quale si sta lavorando. Se L è la lunghezza della frase e H la dimensione degli embedding (768 nel caso specifico del modello BERT utilizzato), l'output sarà della dimensione L×H e verrà suddiviso lungo la prima dimensione per ottenere un embedding contestualizzato separato per ciascun token nella frase.

Il sistema di segmentazione delle parole (*WordPieces*) richiede una particolare attenzione perché il modello in output produce rappresentazioni contestualizzate di sotto-parole, mentre siamo interessati alla rappresentazione contestualizzata di parole complete. Per ottenere un vettore di una parola dall'output BERT sui wordpieces ('pezzi') in cui il tokenizzatore l'ha scomposta, vengono utilizzati vari approcci. Si possono concatenare insieme le immersioni dei wordpieces, farne la media o prendere solo la prima, che spesso contiene la maggior parte delle informazioni. La scelta adottata nel lavoro di testi è quella più condivisa, ovvero prendere la media come rappresentazione della parola [Martinc, Novak e Pollak, 2020c].

A questo punto è stato definito quale word embedding estrarre per una data parola, sia che questa sia stata trattata come parola intera dal tokenizzatore, sia che debba essere ricomposta nei suoi wordpieces. Per ottenere i vettori di singole parole è stato necessario trovare una strategia di estrazione/gestione dei 12 livelli della rete. La decisione sulla scelta del metodo di estrazione è stata oggetto di studio (§ 6.1.4) e ha portato alla scelta della media degli ultimi quattro livelli di hidden.

In molte attività specifiche si estrae solo l'ultimo livello, tuttavia quale livello o combinazione di livelli sia in grado di fornire la migliore rappresentazione non è un'informazione assodata. Esistono diverse alternative utilizzate in letteratura, tra le più usate, insieme all'ultimo livello, vi è la concatenazione degli ultimi quattro livelli, oppure la media degli ultimi quattro livelli. La differenza può essere sottile ma è stato osservato che livelli diversi di BERT codificano tipi di informazioni diverse, quindi la strategia di pooling appropriata può cambiare a seconda dello scopo del lavoro. Gli embeddings del primo livello sono privi di informazioni contestuali, ma più i livelli si spostano in profondità nella rete, più gli embeddings raccolgono informazioni contestuali, suggerendo che le caratteristiche semantiche più salienti vengono acquisite negli strati più alti [Devlin et al., 2019; Jawahar, Sagot e Seddah, 2019]. Tuttavia alcune sperimentazioni[53] concludono che quando ci si avvicina al livello finale gli embeddings tendono a raccogliere informazioni specifiche per attività quali MLM o "*Next sentence prediction*" (NSP), mentre quello che si vorrebbe sono embeddings che codifichino meglio il significato di una parola. Dallo studio della letteratura si trovano posizioni abbastanza diverse in merito: Wang et al. (2020) hanno usato solo l'ultimo livello, Kutuzov e Giulianelli et al. (2020) la somma di tutti i livelli nascosti, Laicher et al. (2020) hanno testato sia la concatenazione degli ultimi quattro livelli che quella del primo con l'ultimo, indicando quest'ultima come quella

[53] Han Xiao, bert-as-service Documentation. 2021: https://readthedocs.org/projects/bert-as-service/downloads/pdf/latest/

che aveva prodotto i migliori risultati. Martinc et al. (2020b) hanno invece optato per la somma degli ultimi quattro livelli.

6.3 Metodo 1: le metriche di distanza

Le rappresentazioni di parole di cui si voglia misurare la variabilità nel tempo, devono essere confrontate con una metrica che sia in grado di 'calcolare' tale cambiamento semantico. Nella descrizione fatta nel cap. 6.1.4 si è potuto capire come le rappresentazioni di parole siano in sostanza vettori di valori reali e come tali possano essere confrontati utilizzando misure di distanza usuali, come la distanza euclidea e la distanza del coseno. In particolare la distanza del coseno (cos) tra due vettori di embeddings, e il suo opposto, cioè la somiglianza del coseno (*cosine similarity*, che si può indicare con 1-cos), sono metriche ampiamente utilizzate nel cambiamento semantico [Shoemark et al., 2019; Schlechtweg et al., 2018]. Le distanze tra embeddings utilizzate in letteratura e adottate nella costruzione del metodo sono state anche la distanza euclidea, quella di Hausdorff e APD sia con distanza del coseno che con distanza euclidea.

Nello studio del fenomeno in diversi intervalli temporali, si impostano metriche più complesse, che fanno assunzioni più fini, distinguendo tra deriva 'incrementale' e deriva 'incettiva', ma toccano aspetti che le sperimentazioni fatte non hanno affrontato. Un altro modo per misurare la deriva temporale di una parola è calcolare la distanza tra le immersioni nel primo e dell'ultimo intervallo di tempo. Tuttavia in presenza di più archi temporali è stato dimostrato che, invece di confrontare solo il primo e l'ultimo intervallo, l'utilizzo delle serie temporali complete per calcolare la deriva aumenta le prestazioni del rilevamento del cambiamento semantico [Shoemark et al., 2019].

6.4 Metodo 2: il clustering

La messa a punto adattativa descritta al cap. 6.1.3 ha avuto l'obiettivo di adeguare il modello linguistico alle peculiarità dell'uso del linguaggio presente nei testi scritti che compongono i corpora. Indipendentemente dal livello di ottimizzazione, BERT viene quindi utilizzato come modello linguistico per ottenere rappresentazioni di parole contestualizzate (o rappresentazioni di utilizzo) per un elenco di parole di interesse. Il passaggio successivo riguarda il modo con il quale tutte queste rappresentazioni per una determinata parola, possano essere aggregate in gruppi 'interpretabili' attraverso l'uso di algoritmi di clustering. Seguendo l'intuizione fornita da Bishop (2006), possiamo pensare a un cluster come a un insieme di rappresentazioni dello stesso uso di una parola, in cui considerare la distanza tra i cluster come la ragione di un uso differente delle parole che vi sono contenute.

Come misurare queste distanze dipende dall'algoritmo di clustering, ma in generale le distanze identificano criteri di similarità tra elementi dello stesso cluster. Nel caso specifico si operano aggregazioni di embeddings contestualizzati di parole. Le librerie software che offrono funzionalità di clustering sono molte; per Python la libreria Scikit-learn propone diversi[54] algoritmi ma ognuno possiede caratteristiche che ne condizionano l'uso e che è bene conoscere. Quelli utilizzati sia in SemEval che in DIACR-Ita sono stati sperimentati tutti, con un diverso grado di approfondimento. È infatti importante non perdere mai di vista l'obiettivo dello studio e utilizzare solo quegli strumenti che meglio si prestano ai propri scopi. Con queste premesse sono state esaminate le diverse tecniche di clustering con un atteggiamento critico, valutando quali strategie soggiacessero ai diversi algoritmi.

Gaussian mixtures è un modello probabilistico che presuppone che tutti i punti dati siano generati da un insieme finito di distribuzioni gaussiane con parametri sconosciuti. Si può pensare a questi modelli come a una

[54] https://scikit-learn.org/stable/modules/clustering.html

generalizzazione del clustering di K-means, finalizzata ad incorporare informazioni sulla struttura di covarianza dei dati e sui centri delle gaussiane latenti. L'algoritmo è definito 'not scalable' dalle indicazioni fornite dalle librerie software che ne implementano le funzionalità e se ne indica l'uso per misurare stime di densità e induzione. Affiniy Propagation (AP) utilizza una graph distance (per esempio Nearest-Neighbor) per creare cluster, inviando 'messaggi' tra coppie di campioni fino alla convergenza. Dato un piccolo numero di esemplari identificati come più rappresentativi, si produce un aggiornamento iterativo dell'idoneità di ogni rappresentate di un gruppo fino alla convergenza. A quel punto sono scelti gli esemplari finali e viene fornito il raggruppamento finale. È interessante perché seleziona il numero di cluster in base ai dati forniti, ma come Gaussian mixtures, è definito 'not scalable' dalle specifiche d'uso fornite dalla libreria software. Ha due parametri importanti: preference, che controlla quanti esemplari vengono utilizzati, e damping che misura come il valore corrente viene mantenuto rispetto ai valori in ingresso, per evitare oscillazioni numeriche durante l'aggiornamento dei 'messaggi'. Lo svantaggio principale di AP è la sua complessità. L'algoritmo ha una complessità che riguarda i tempi di esecuzione di ordine quadratico $O(N2 \cdot T)$, dove N è il numero di campioni e T è il numero di iterazioni fino alla convergenza. Inoltre, la complessità della memoria è sempre dell'ordine $O(N2)$ se viene utilizzata una matrice di somiglianza densa, che può essere però ridotta se viene utilizzata una matrice di somiglianza sparsa. Ciò rende AP più appropriato per set di dati di piccole e medie dimensioni, anche a detta degli utilizzatori.

Gli algoritmi di clustering più utilizzati in letteratura sono K-means e Dbscan, infatti, con caratteristiche diverse, entrambi rispondono a compiti general-purpose in cui si possono avere un grande numero di punti e un buon numero di cluster, in più utilizzano distanze tra punti per il clustering. Il primo di default utilizza quella euclidea, l'altro considera la distanza un parametro impostabile dall'utente. K-means raggruppa i dati cercando di separare i campioni in un numero di gruppi di uguale varianza, riducendo al minimo un criterio noto come inerzia (o somma dei quadrati all'interno del cluster). Per

questo motivo richiede il numero di cluster come parametro di input. La versione dell'algoritmo implementato dalla libreria software Schikt-learn è indicato anche come algoritmo di Lloyd e utilizza una tecnica di raffinamento iterativo, che non garantisce di trovare l'ottimo, ma ne riduce la complessità, che risulta quindi essere O(n · k · m · i) dove *n* il numero di punti, *k* il numero dei cluster, *m* è il numero di attributi, e i è il numero di iterazioni sull'intero data set. L'algoritmo al primo passo sceglie i centroidi iniziali, al secondo passo assegna ogni campione al centroide più vicino. Al terzo passo crea nuovi centroidi prendendo il valore medio di tutti i campioni assegnati a ciascun centroide precedente, calcolando la differenza tra il vecchio e il nuovo centroide. L'algoritmo ripete gli ultimi due passi fino a quando lo spostamento dei centroidi non è più significativo. L'utilizzo di una funzione di distanza diversa dalla distanza euclidea (quadrata) può impedire la convergenza dell'algoritmo. Ne sono state proposte alcune modifiche, come K-medoid, per consentire l'utilizzo di altre misure di distanza, ma con un impatto sulla complessità non trascurabile, fatto per il quale è accomunato ad AP. K-means si adatta bene a un gran numero di compiti ed è stato utilizzato in campi di studio e ambiti di applicazione diversi.

L'algoritmo Dbscan interpreta i cluster come aree ad alta densità separate da aree a bassa densità. Questa caratteristica permette all'algoritmo di trovare cluster di qualsiasi forma, al contrario di K-means che presuppone che i cluster abbiano una forma convessa. La caratteristica principale di Dbscan è il concetto che campioni di base, si trovino in aree ad alta densità. L'algoritmo necessita in input dei parametri: *min_samples* (o minPts ~ campioni minimi), quantità minima di punti dati affinché un campione possa essere considerato un campione principale ed *eps* (epsilon), ossia la distanza massima tra questi punti dati, che definisce formalmente cosa si intende quando si parla di densità. Valori di *min_samples* più alti o *eps* più bassi indicano una densità maggiore: condizione necessaria per determinare un cluster. L'algoritmo prevede tre tipi di entità a) *core point* sono punti interni a un cluster, la cui densità è superiore a una soglia *min_samples*; b) *border point* ovvero punti che hanno una densità minore di *min_samples*, ma nelle cui vicinanze (a

distanza < *eps*) è presente un *core point*; c) *noise point* tutti i punti che non sono *core point* o *border point*. Dbscan esegue esattamente una invocazione per ogni punto e, se viene utilizzata una struttura indicizzata che esegue un'interrogazione dei punti vicini in O(log n), si ottiene un tempo globale di esecuzione pari a O(n · logn). Senza l'uso di strutture indicizzate, il tempo di esecuzione è pari a $O(n^2)$, è dello stesso ordine anche il tempo di elaborazione in memoria.

6.4.1 Gli iperparametri

In questo paragrafo sono affrontate le questioni chiave che riguardano l'uso degli algoritmi di clustering: gli iperparametri. È infatti esperienza consolidata di chi si misura con loro uso affrontare problemi quali: l'inizializzazione e configurazione ottima. Queste tecniche sono molto sensibili anche a piccoli cambiamenti, specialmente con dati di dimensioni contenute. Il numero di parametri iniziali è quindi una questione non trascurabile: meno ve ne sono, meno lavoro di messa a punto è richiesto. Dalla letteratura studiata è emerso evidente il desiderio di sperimentare nuovi approcci, intraprendendo strade meno battute, ma risulta altrettanto evidente la necessità di ritornare a più miti consigli nei casi in cui serve entrare nel merito dei risultati dell'algoritmo. Molti infatti utilizzano un algoritmo meno noto, sempre prevedendo l'uso parallelo anche di K-means.

Nell'esecuzione dei task di SemEval e DIACR-Ita il clustering è stato ampiamente utilizzato, in alcuni casi su tutti gli intervalli temporali congiuntamente, come hanno fatto Kutuzov e Giulianelli et al. (2020) con K-means oppure come Martinc et al., (2020b) con Propagation Affinity (PA). Karnysheva et al. (2020), pur lavorando su un unico corpus (C1+C2) come gli altri, distinguono però il trattamento tra lingue: per l'inglese e lo svedese hanno applicato K-means, mentre per il tedesco e il latino hanno utilizzato Dbscan [Ester et al., 1996]. Le autrici riportano di prestazioni diverse dei due algoritmi relativamente alle lingue del task e imputano un maggiore vantaggio

nell'uso di Dbscan, sottolineando un minore impatto della forma dei cluster rispetto K-means. Tuttavia l'algoritmo necessita di due iperparametri: *min_samples* ed *eps*, mentre K-means ne richiede uno solo: il numero k di cluster.

Per trovare tale valore k esistono in letteratura varie strategie. È possibile utilizzare il coeff. di Silhouette [Rousseeuw, 1987], che misura la densità dei cluster e la distanza tra loro, calcolando sia la media per tutti i punti, sia una combinazione della distanza media intra-cluster e quella della distanza media dal cluster più vicino. Un criterio più oggettivo (che funziona meglio per cluster sferici come quelli prodotti da K-means) e quello di calcolare il *silhouette score* come media dei *Silhouette coefficient* di ciascun pattern e scegliere quello che lo massimizza:

$$Silhouette\ coefficient = \frac{(b-a)}{\max(a,b)}$$

dove *a* è la distanza media intra cluster del pattern e *b* la distanza tra il pattern e i pattern del cluster ad esso più vicino (escluso quello di appartenenza). La funzione silhouette_score() che calcola questa metrica è presente nella libreria software Scikit-learn, risulta però computazionalmente pesante per grandi dataset. Oppure è possibile usare approcci dinamici impostando i possibili valori da assumere attraverso un intervallo. Karnysheva et al. (2020) hanno impostato il limite superiore dell'intervallo a 10, dopo prove sperimentali sui dati, eseguendo l'inizializzazione dell'algoritmo con valori da 1 a 10. L'algoritmo converge quando l'errore al quadrato della distanza tra i punti dati e il centro del loro cluster raggiunge un minimo locale. Tracciando questo valore rispetto ai diversi valori di k si ottiene una rappresentazione grafica, su un piano di due dimensioni, che raffigura una linea l_1 caratterizzata da un 'gomito' o punto di discontinuità. Per trovare esattamente tale punto, viene inserita un'altra linea l2 che collega entrambe le estremità di l_1. Il punto corrispondente alla massima distanza (l_2, l_1) localizza esattamente il gomito nel grafico.

Per quanto riguarda Dbscan la ricerca dei migliori iperparametri può avvalersi di due diversi approcci: uno che separa la ricerca dei due iperparametri e l'altro che cerca di farne una ricerca congiunta. Nel primo caso prima si cerca *eps*, calcolando la distanza di ogni punto dal suo vicino più prossimo usando l'algoritmo Nearest Neighbors e costruendo un vettore di distanze ordinato. Tale approccio trova il valore ottimale per epsilon, nel punto di massima curvatura della proiezione della curva dei valori, in uno spazio bidimensionale. Questo punto nella rappresentazione grafica corrisponde anche al numero ideale di *min_samples*. Se si opta per una ricerca incrociata con il valore di *eps* trovato, si alimenta Dbscan con vari intervalli di campioni minimi da 1 a n (più comunemente n = 10), utilizzando il coefficiente di Silhouette per stabilire il migliore *min_samples*.

Un'altra tecnica, nota in letteratura come *grid-search*, è ampiamente utilizzata nell'ambito dello sviluppo di reti neurali, per le quali sia necessario trovare una configurazione ottimale degli iperparametri. Questi ultimi vengono inseriti in una griglia, nel caso di Dbscan si tratta di un certo numero di alternative ammissibili per *eps* e *min_samples,* e si produce l'esecuzione dell'algoritmo con tutte le combinazioni di valori. Il risultato della grid-search fornisce le combinazioni migliori, la cui sintesi è possibile proiettare graficamente in una tabella, con l'uso della libreria grafica Seaborn[55].

6.4.2 Considerazioni finali sugli algoritmi di clustering

È importante sottolineare quali sono i criteri che hanno determinato una prevalenza nell'uso di un algoritmo a discapito dell'altro. Dalla letteratura non si evince un metodo standard, generalizzabile per ottenere una configurazione ottimale di nessuno degli algoritmi di clustering. Si ritiene questa caratteristica un impedimento all'uso più diffuso di queste tecniche. Sono state descritte tecniche varie e funzioni approntate in varie librerie software ma non di valore

[55] https://seaborn.pydata.org/

generale, applicabili a tutti i casi. Per ovviare a questo molti autori sono ricorsi ad algoritmi con inizializzazione predefinita, offerta dal software che li implementa, non sempre adatte ai propri dati. Altri hanno realizzato procedure di inizializzazione mutuando o combinando funzioni matematiche di misura, che però sono state testate sui loro particolari dati. Il più delle volte ne risulta una soluzione artigianale che non assicura la possibilità di essere applicata ad un altro studio. Per esempio la tecnica utilizzata da Karnysheva et al. (2020) per trovare il numero k di cluster ottimale per K-means, per quanto suggestiva, non è automatizzabile in casi di dati di dimensioni contenute, in cui le curve non presentano tali 'gomiti'.

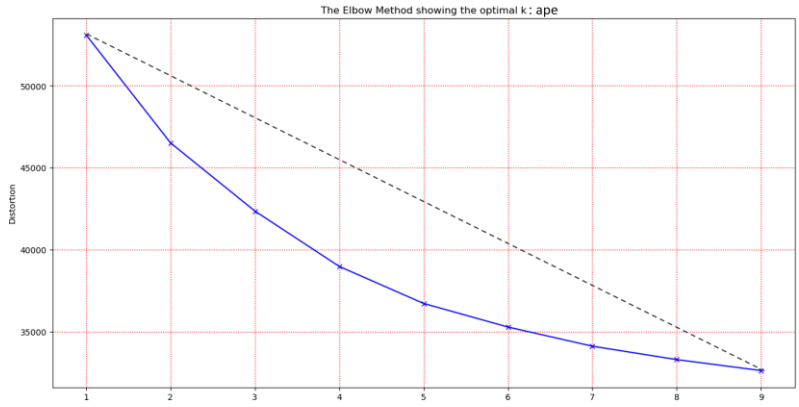

Figura 6: proiezione grafica del metodo del gomito per la parola 'ape'

In figura 6 è mostrato un esempio di applicazione della tecnica alla parola target ape del task DIACR-Ita. In modo più evidente per questa, ma anche per altre parole, è possibile comprendere come la ricerca di massimi o di punti di discontinuità sia aleatoria in casi quello raffigurato in immagine.

Dbscan aveva in teoria prospettive d'uso più promettenti. Consentiva infatti di produrre cluster di diverse forme, di gestire gli outliers di prevedere l'assenza di cluster, tutte ragioni più che valide per ritenerlo preferibile. Tuttavia dalle sperimentazioni condotte ha ottenuto risultati inferiori rispetto a K-means. I motivi di queste prestazioni vanno cercati nell'incapacità di trovare automaticamente una configurazione ottimale. I metodi per trovarla automaticamente non sono adeguati. La grid-search non solo è impegnativa

computazionalmente per griglie di valori grandi, ma soprattutto non assicura un'unica soluzione.

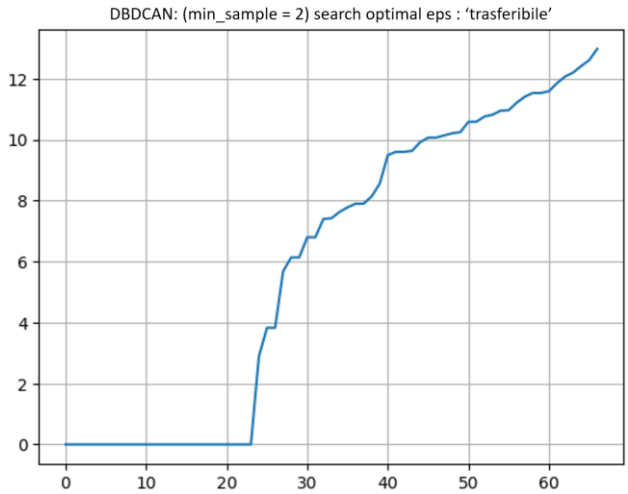

Figura 7: calcolo di optimal eps in Dbscan per la parola 'trasferibile'

L'alternativa, offerta dalla funzione che produce una curva rappresentata in un grafico di due dimensioni, è anch'essa affetta dallo stesso problema indicato per K-means in figura 6: ossia una curva in cui non sia facile ravvisare il punto di massima curvatura (figura 7).

La strategia utilizzata nelle sperimentazioni ha fatto ricorso all'incrocio tra i valori restituiti dalla grid-search e quelli della funzione con proiezione della curva. Si è registrato un sensibile disallineamento tra i risultati: spesso i valori indicati dalla proiezione grafica erano maggiori di quelli risultanti dalla grid-search, sia per *eps* che per *min_samples*. Questa difformità di risultati ha impattato in modo deciso nel tentativo di generalizzazione del metodo con questo algoritmo.

Capitolo 7

Applicazione del metodo al task DIACR-Ita

Questo capitolo è dedicato alla prova il metodo proposto, si è infatti provato a svolgere il task di DIACR-Ita come gli altri partecipanti. Questa possibilità è significativa per la verifica di ipotesi e metodi. La prova sul campo è stata fondamentale per comprendere quando e come una strategia fallisse. L'analisi dell'errore fa parte del corretto condizionamento di un problema ed è importante sviluppare un approccio che permetta di controllare gli errori e la loro propagazione.

Nel caso specifico si è trattato di capire se si trattasse di errori casuali o sistematici. I primi sono molto insidiosi e non eliminabili, solitamente sono indice di un mal condizionamento del problema o di dati non idonei, entrambe condizioni che pregiudicano una soluzione positiva del problema. Il secondo tipo di errore è quello più semplice da risolvere perché la sua analisi può mettere in luce la ragione strutturale che lo ha generato e permette di trattarlo e, quando possibile, risolverlo. Senza questa fase di validazione non si sarebbe compresa l'importanza dell'inizializzazione dei metodi, né come fosse stretto il rapporto tra dimensione dei dati e prestazione dei metodi di indagine. È stata l'occasione per poter entrare nel merito degli strumenti utilizzati, comprendendone i limiti e i margini di miglioramento possibili e in generale sperimentare nei fatti la complessità del compito.

Si è stabilito di provare l'adozione di tutti i metodi individuati nella definizione generale del metodo d'indagine (§ 6.1), non dando priorità a

nessuna configurazione: metodo senza aggregazioni o metodo di clustering. Questo non solo per fare pratica con le tecniche e gli approcci, ma soprattutto per analizzare sperimentalmente quanto descritto, cercando di comprendere le difficoltà oggettive e gli errori che si possono commettere. Dopo la messa a punto dei modelli *fine-tuned* C1 e C2 sono state riprodotte le condizioni per eseguire l'esperimento proposto dal task DIACR-Ita.

(i) Si è scaricato il set di parole che erano state proposte ai partecipanti[56]: *ape, brama, campanello, campionato, cappuccio, discriminatorio, egemonizzare, lucciola, pacchetto, palmare, pilotato, piovra, polisportiva, processare, rampante, tac, trasferibile, unico*.

(ii) Al fine di acquisire embedding contestualizzati, i corpora sono stati prima suddivisi in frasi e, per comodità di esecuzioni delle successive analisi, sono stati creati i due sotto-corpora con tutte le frasi che contenessero le occorrenze delle parole target. Le modalità di estrazione delle frasi hanno riguardato una fase di pre-processing dei corpora che ha creato i corpora: F_{C1} contenenti le frasi di C1, F_{C2} quelle di C2 e F_{C1C2} che unisce le frasi di C1 e C2. Tutti corredati da etichette/indici che ne identificassero l'appartenenza a C1 o C2. È stato oggetto di valutazione il numero massimo di frasi per parola che sono state utilizzate per il calcolo del cambiamento di senso. Le posizioni in merito non erano uniformi, i partecipanti al task avevano adottato strategie diverse: chi aveva stabilito una soglia, Laicher et al. (2020) a questo proposito avevano affermato genericamente: '*We limited the number of uses to 200 for computational efficiency reasons*'. Altri hanno operato una selezione random dai corpora di frasi, ma la cui dimensione non è stata confutata. Nel caso di questa sperimentazione la soglia è stata fissata a 700, per le parole con maggiori attestazioni

[56] Non avendo potuto aver accesso ai dati originali del task si sono riscontrate minime differenze nelle frequenze di alcune parole: 'lucciola' in C1 da task ha freq. = 64, mentre da frasi estratte ha freq. = 91; in C2 da task freq. = 226, mentre da frasi estratte freq. = 234. Per 'brama' la differenza è in C2 dove da task la freq. = 93 mentre da frasi estratte risulta = 95. Anche per la parola 'rampante' è stata registrata una minima differenza sia in C1 che in C2. L'esiguità delle differenze non ha inficiato comunque i risultati.

(*campionato*, *pacchetto* e *unico*), mentre per le altre si sono utilizzate tutte le occorrenze.

(iii) È stato utilizzato il modello pre-trained per l'italiano (§ 6.1.1) e condotto il fine-tuning su C1 e su C2, ottenendo due modelli addestrati separati per ogni intervallo temporale previsto dal task.

(iv) F_{C1}, F_{C2} (sottoinsiemi rispettivamente di C1 e C2) e F_{C1C2} (solo per l'applicazione del metodo 2.1) sono stati sottomessi ai rispettivi modelli addestrati e quindi estratti gli embeddings relativamente alla media degli ultimi quattro livelli di hidden di tutte le occorrenze delle parole target. Si è comunque ritenuto interessante estrarre anche l'ultimo livello per confrontare i risultati.

7.1 Metodo 1: distanze tra vettori

Una prima misura della distanza è stata calcolata sugli embedding contestualizzati estratti per ogni intervallo di tempo. Oltre alla distanza del coseno e quella euclidea è deciso di utilizzare la distanza Average Pairwise Distance (APD), metrica anch'essa basata sulla distanza tra i vettori degli embeddings:

$$\text{APD} = \frac{1}{n_V * n_W} \sum_{v \in V, w \in W} d(v, w)$$

dove V e W sono due insiemi di vettori e n_V e n_W denotano il numero di vettori da confrontare, mentre d(v, w) si riferisce alla metrica di distanza utilizzata (nel caso della sperimentazione erano sia la distanza del coseno che quella euclidea). L'approccio seguito si rifaceva alle soluzioni proposte sia da Wang et al. (2020) che da Laicher et al. (2020), ossia ridurre l'identificazione del cambiamento di significato di una parola, tra due distinti periodi di tempo, a un problema di classificazione binaria. Dati due insiemi di vettori di token relativi a due periodi di tempo t1 e t2, la strategia consisteva nel prendere i vettori da entrambi gli insiemi e misurare la loro distanza a coppie [Sagi et al., 2009; Schlechtweg et al., 2018; Giulianelli et al., 2020; Beck, 2020; Kutuzov e Giulianelli, 2020]. Il punteggio che misura il cambiamento semantico della

parola era quindi la distanza media di tutti i confronti. Wang e il gruppo 'University of Padova @ DIACR-Ita' hanno definito la distanza a coppie degli embeddings nella sua configurazione più generale (*Average Geometric Distance*) ma con identica formula di APD, mentre per la metrica hanno utilizzato la distanza euclidea, la distanza di Canberra [Lance e Williams, 1966)] e la distanza del coseno. Hanno sperimentato anche la distanza di Hausdorff (HD) [Rockafellar and Wets, 2009] che è generalmente utilizzata per misurare la distanza tra due set non vuoti, vale a dire:

$$\text{HD}\ (\Phi_i^{C1}, \Phi_i^{C2}) = \max(\ sup_{x \in \phi_i^{C1}} inf_{y \in \phi_i^{C2}} ||x - y||_2, sup_{x \in \phi_i^{C2}} inf_{y \in \phi_i^{C1}} ||x - y||_2)$$

Le distanze indicate sono state tutte sperimentate sui dati del task con risultati diversi. La prima osservazione che viene spontanea è la definizione della soglia per l'attribuzione della classe per la classificazione binaria. Mentre per APD con distanza del coseno (APD_{cos}) era intuitivo porre un cambiamento di significato quando la metrica superava lo 0.50, essendo il range definito tra 0 e 1; nel caso di HD (il cui calcolo si basa sulla distanza euclidea) e in APD con distanza euclidea (APD_{eu}) le cose erano più complesse. Dalla lettura delle esperienze dei partecipanti al task non si evinceva una soglia di attribuzione definita, e nemmeno un modo sicuro per calcolarla. In genere si è stabilita empiricamente operando sul campione di dati.

Nella sperimentazione condotta le soglie per APD_{eu} e HD sono state impostate come media di tutti i valori calcolati per il campione. Con questa strategia si sono ottenute le soglie $\delta_{APD} = 15.11$ e $\delta_{HD} = 12.54$. Nella figura 8 sono mostrate, per tutte le parole del task, le distanze calcolate e quali di queste[57] risultano sopra la soglia attesa e quindi indicano un cambio di significato tra C1 e C2.

[57] Per APD_{eu}, HD e APD_{cos} sono evidenziati in modo diverso i valori sopra la soglia: per HD corsivo con grassetto, per APD_{eu} grassetto sottolineato, mentre per APD_{cos} grassetto in rosso.

			Distanza euclidea				Distanza del coseno	
Parola target	Dim. frasi C1	Dim. frasi C2	Distanza euclidea embC1 embC2 medi	Dev. standard	APD con distanza euclidea	Distanza di Hausdorff	ADP distanza del coseno **(4L)**	ADP cosine similarity **(UL)**
ape *	154	260	11.12	1.66	**16.57**	***13.26***	**0.56**	0.038
brama	21	90	12.75	1.97	14.99	***13.16***	0.42	0.046
campanello	127	581	13.30	2.37	14.51	11.02	0.39	-0.0008
campionato	700	678	12.77	1.95	**16.72**	11.16	0.47	0.0172
cappuccio	72	184	14.65	2.53	**17.74**	11.71	0.49	0.0265
discriminatorio	121	233	13.27	1.22	12.87	10.84	0.34	0.0164
egemonizzare	12	38	11.70	2.14	10.70	***15.79***	0.32	-0.043
lucciola *	105	309	11.99	2.07	**15.11**	***15.04***	**0.504**	-0.193
pacchetto	317	614	13.72	1.77	**17.54**	***12.86***	0.49	0.023
palmare *	21	81	12.29	1.35	**15.982**	***14.17***	**0.508**	-0.05
pilotato *	40	261	13.10	1.74	**15.987**	***13.32***	**0.507**	-0.00019
piovra	34	563	12.89	1.63	14.43	***13.17***	0.42	-0.00…3
polisportiva	87	132	12.26	1.32	10.81	10.63	0.30	0.010
processare	45	583	13.54	2.06	14.14	12.16	0.39	-0.004
rampante *	32	406	12.98	1.89	13.11	11.39	0.38	0.023
tac *	139	436	10.24	3.03	**18.25**	***18.29***	**0.65**	0.015
trasferibile	9	58	13.32	1.75	14.22	13.91	0.40	0.114
unico	700	700	12.14	2.18	**19.09**	***15.29***	**0.56**	-0.0004

Figura 8: risultati del calcolo delle distanze tra embeddings estratti dalla media degli ultimi 4 livelli (etichetta 4L) e confronto con corrispettivo ottenuto da estrazione dell'ultimo livello di hidden (etichetta UL) per APD

La prestazione migliore si ottiene con la distanza del coseno [Salton e McGill, 1983], con la quale di ottiene un'accuracy di 0.88.

La sperimentazione ha messo in evidenza un problema di corretta classificazione delle parole target con occorrenze numerose. Per queste la misura di distanze tra embeddings, operata con tutte le metriche, produce distanze maggiori tra i relativi vettori, portando ad attribuire come nuovo senso anche nei casi in cui non c'è. Nel caso della parola unico, che ha il numero maggiore di embeddings sia in C1 che in C2 questo è molto evidente poiché nella sua attribuzione tutte le metriche sbagliano.

7.2 Metodo 1.2: distanza tra embeddings medi

Proposto da Martinc, Novak e Pollak (2020c), questo metodo calcola la media di tutte le incorporazioni contestualizzate di una parola che appare in un determinato periodo di tempo. Il metodo si avvale di questo 'rappresentate medio' per misurare l'insieme di rappresentazioni vettoriali specifiche di una parola in quel preciso intervallo di tempo. Queste incorporazioni medie possono essere confrontate usando la distanza euclidea o del coseno:

$$\text{Avg}(E_w^{(t_1)}, E_w^{(t_2)}) = d\left(\frac{\sum_{u_i \epsilon E_w^{(t_1)}} u_i}{N^{(t_1)}}, \frac{\sum_{u_i \epsilon E_w^{(t_2)}} u_i}{N^{(t_2)}}\right)$$

In modo simile Kutuzov e Giulianelli (2020) hanno misurato la variazione con la distanza del coseno tra ciascun embedding contestualizzato e un baricentro, ovvero l'immersione media di token per una data parola (~il rappresentante medio). La media di queste distanze rappresenta il coefficiente di variazione di una parola. In questo approccio la distanza tra le immersioni medie viene utilizzata come misura della deriva semantica tra intervalli di tempo. La deriva totale è la distanza tra la media delle rappresentazioni dei token tra il primo intervallo di tempo t_0 e l'ultimo t_n. L'intuizione è che per parole che hanno diversi sensi e usi diversi, la distanza dal rappresentante medio sia maggiore rispetto a parole che sono monoseme. Tuttavia a detta degli utilizzatori il metodo è poco flessibile per situazioni in cui parole acquistano o perdono i sensi e nella valutazione di parole polisemiche che comunque rimangono stabili nel tempo.

Rielaborando il metodo la distanza del coseno, metrica robusta ma meno fine rispetto alla euclidea che tiene conto della lunghezza dei vettori, è stata sostituita dalla distanza euclidea. Calcolare la distanza euclidea di ogni embeddings dal baricentro è quindi interpretata come la misura di variazione dell'insieme degli embeddings dalla loro media. Calcolando la deviazione standard di quell'insieme (campione) si può dedurre di quanto il campione si discosti dal suo baricentro. Il coefficiente di variazione è quindi la deviazione standard del campione. I risultati, sia che si considerino le medie di tutte le

distanze euclidee tra gli embedding o la distanza tra gli embedding medi, non superano la baseline.

7.3 Metodo 2: gli algoritmi clustering utilizzati

Le prove sono state condotte con tutti gli algoritmi utilizzati nella letteratura esaminata (§ 6.4) con risultati diversi. Gaussian Mixture è stato accantonato quasi subito per la sua poca scalabilità. Mentre per K-means e Dbscan si sono incontrati gli stessi problemi sollevati dagli autori che li hanno adottati, per PA si sono osservati problemi maggiori. Martinc et al. (2020a), che lo hanno utilizzato, lo ritengono migliore nei casi in cui viene analizzato il numero di usi diversi della parola, piuttosto che il numero di sensi, la cui numerosità può variare molto a seconda della parola e non riflettere necessariamente la quantità di sensi da un punto di vista lessicografico. Analizzando i risultati della sperimentazione, condotta senza nessuna inizializzazione sui parametri dell'algoritmo, si è evidenziata una produzione molto alta di cluster, che sembra legata ad una selezione più fine della somiglianza tra gli embeddings. Anche intervenendo sui parametri, soprattutto il parametro *preference* che influenza il numero di cluster prodotti (i punti con valori di preference maggiori vengono scelti come esemplari di cluster), non si è riusciti a trovare una configurazione ottimale. L'altra caratteristica riscontrata è che, contrariamente agli altri algoritmi di clustering, i risultati migliori sono stati prodotti per parole che avevano mediamente poche occorrenze. Nel complesso le analisi e le sperimentazioni condotte non hanno mostrato un vantaggio nell'uso di PA: l'interpretazione di questo gran numero di cluster prodotti non è intuitiva. Probabilmente la necessità di un uso più fine dell'elaborazioni del clustering può giustificarne l'uso, quando cioè l'indagine richiede maggiore dettaglio. A conferma di questo, gli autori che lo hanno scelto, hanno sottolineato come il modello BERT non conservi solo le informazioni semantiche per le rappresentazioni contestualizzate e sia fortemente influenzato dalla sintassi [Reif et al., 2019], inoltre hanno aggiunto che i

raggruppamenti ottenuti dalle rappresentazioni di una parola non riflettono necessariamente i diversi sensi della parola, ma spesso riflettono solo i diversi modi in cui viene utilizzata, indicandone la ragione negli intervalli temporali indagati e nel tipo dei materiali.

Per tali motivi, sebbene si sia sperimentato l'uso di PA nel lavoro di tesi, non si è ritenuto un buon candidato a costituire parte del metodo proposto, per il quale serve la maggiore versatilità possibile e il più alto grado di comprensione delle dinamiche sottostanti. Si è quindi deciso di approfondire lo studio e la sperimentazione sia di K-means che di Dbscan.

7.3.1 Applicazione del metodo ai dati DIACR-Ita

La prova sui dati C1 e C2 è stata condotta utilizzando sia Dbscan che K-means. Per entrambi la distanza utilizzata per il clustering è stata quella euclidea, per il primo provando anche la configurazione con la distanza del coseno. I dati di partenza per il loro utilizzo sono stati preparati sia nella configurazione C1+C2 che in quella che mantiene il clustering dei due corpora separato come in Kanjirangat et al. 2020. Sempre rifacendosi a quanto utilizzato in SemEval e DIACR-Ita si sono implementate diverse tecniche per la ricerca dei valori ottimi di *k* per K-mean e di *eps* e *min_samples* per Dbscan.

7.3.1.1 K-means: ricerca degli iperparametri

Per l'algoritmo K-means è stato più semplice implementare una procedura automatica che riuscisse a facilitare la ricerca del migliore valore di k. Sono disponibili librerie grafiche di vario tipo che implementano tecniche di generazione di un k ottimo, ma si tratta di funzioni già pronte all'uso, in genere non danno la garanzia di buoni risultati, infatti il loro utilizzo non è riuscito a dare un contributo significativo alle sperimentazioni. Le alternative riguardano anche funzionalità propriamente grafiche offerte dalla libreria

Yellowbrick di Scikit-yb[58], che combina varie tipologie di misure. Queste strategie hanno mostrato una estrema varietà di risposte a seconda delle dimensioni dei dati, rendendone difficile un uso generalizzato. Si è dimostrato più efficace l'uso delle funzionalità della libreria Scikit-learn, che nella maggior parte dei casi usano come base il calcolo del coefficiente di Silhouette con vari gradi di complessità. La sperimentazione migliore si è avuta usando il *silhouette score* (§ 6.4.1) ottenuto come media dei coeff. Silhouette di ciascun pattern, scegliendo poi quello che lo massimizza. Il valore di silhouette score ottenuto si è poi confrontato con i risultati della tecnica utilizzata da Kanjirangat et al. (2020). In modo simile agli autori si sono calcolati per ogni cluster: la media per tutti i punti, la sua distanza media intra-cluster e la distanza media dal cluster più vicino. La differenza tra queste due grandezze normalizzate dal massimo delle due, viene utilizzata come valore da massimizzare per selezionare il k ottimale. Lo stesso approccio viene utilizzato per determinare i centroidi iniziali. Nel caso specifico si è stabilito un numero di iterazioni massimo tenendo in considerazione la dimensione dei dati. Dopo varie sperimentazioni il numero massimo è stato confermato pari alla soglia individuata dagli autori: n= 10. Partendo quindi dal valore k ottimale trovato si esegue l'algoritmo k-means per n iterazioni per determinare i migliori centroidi. Trovato il k ottimale il clustering è applicato all'estrazione di tutti gli embeddings delle parole target, calcolati come la media degli ultimi quattro livelli di hidden della rete.

Per confronto sono stati prodotti plot grafici di appoggio per la visualizzazione dei risultati. Per il *metodo 2.1* (corpus = C1 + C2) nella produzione grafica dei risultati si distinguono i cluster dei diversi corpora dalle etichette di appartenenza: '1' per C1, '2' per C2. Nel caso del *metodo 2.2* è stato necessario definire un criterio di confronto del risultato proposto dai metodi di clustering: (i) una prima ipotesi, spesso utilizzata, si basa sul confronto tra il numero di cluster prodotti nei due intervalli temporali. In generale questa è una misura grossolana, che incontra problemi quando esiste una disparità di

[58] https://www.scikit-yb.org/en/latest/teaching.html

dimensione tra i dati di C1 e C2. In generale quando il rapporto è 1:10 la probabilità che nel corpus con le maggiori occorrenze si produca un numero maggiore di cluster è molto alta. Questo fenomeno si è presentato con ogni algoritmo di clustering sperimentato: K-means, Dbscan, AP. (ii) Un'altra strategia cerca di indagare la composizione dei cluster. In questo filone si inquadrano tutti i metodi di conteggio/controllo delle etichette associate ai cluster. Strategie poco scalabili. (iii) Un ulteriore modo per valutare una mappa tra i cluster di C1 e quelli di C2, riguarda la proiezione degli uni sugli altri. Un metodo costoso computazionalmente, che era tuttavia interessante valutare, proprio per la possibilità di proiettare su plot una rappresentazione grafica dei risultati. Xuri Tang nel suo '*A State-of-the-Art of Semantic Change Computation*' indicava come, il confronto e la costruzione di approcci alla caratterizzazione diacronica del senso delle parole e alla modellazione del cambiamento avesse necessità di avvalersi di un'ulteriore esplorazione delle tecniche di visualizzazione dei dati per la giustificazione di ipotesi.

Rifacendosi a quanto indicato da Kanjirangat et al. 2020 si è mantenuta la distanza euclidea tra i vettori come indicatore di somiglianza semantica. Per semplicità si è rappresentato ogni cluster con il suo centro di massa. Quando i due corpora avevano lo stesso numero di cluster, si è ridotta l'identificazione della mappa tra i cluster di C1 e C2, al peso minimo in un grafo bipartito completo. In questa configurazione i nodi del grafo sono associati ai due insiemi di punti e i pesi sono le distanze euclidee tra i centri di massa. Se il numero di cluster era diverso, per esempio m1 > m2, dove m1 indica il numero di cluster in C1 e m2 quello in C2, al secondo corpus la procedura aggiunge un cluster fittizio m1 - m2 e si azzerano i pesi per tutti gli archi che collegano questi elementi. Si procede allo stesso modo se m2 > m1. La corrispondenza ottimale che minimizza la somma dei pesi può essere calcolata in un ordine di tempo cubico con l'algoritmo ungherese [Kuhn, 1955; Jonker e Volgenant, 1987]. Il risultato è una corrispondenza uno a uno tra i cluster nei due corpora, non importa se corretta o fittizia. Poiché un cluster fittizio in un corpus ha una distanza zero da tutti i cluster dell'altro corpus, la corrispondenza restituita dall'algoritmo ungherese riduce al minimo la distanza tra i cluster appropriati.

In questo approccio, due veri e propri cluster nei due corpora abbinati dall'algoritmo sono intesi come rappresentativi dello stesso senso. I cluster appropriati di un corpus che puntano a un cluster fittizio sono invece considerati come un nuovo senso apparso solo nel secondo corpus, o il vecchio senso che si è verificato solo nel primo corpus. Dopo l'abbinamento dei due clustering di C1 e C2, si definisce un singolo clustering con m = max {m1, m2}. In pratica, i vettori di due cluster abbinati dall'algoritmo ungherese sono assegnati a un singolo cluster 'impuro' (misto), mentre quelli legati a cluster fittizi producono cluster 'puri' (isolati). Questo approccio è stato trasformato in una routine applicabile all'occorrenza, nel caso del task è stata poi applicata a tutti i dati. Pensando ad una generalizzazione del metodo d'indagine si ritiene che possa aver senso limitarlo ai casi in cui l'analisi dei risultati o lo studio approfondito lo dovesse richiedere (§§ 7.4 e 8).

7.3.1.2 Dbscan: ricerca degli iperparametri

Come esposto in (§ 6.4.1) l'utilizzo dell'algoritmo Dbscan ha comportato l'onere della ricerca degli iperparametri migliori da impostare per l'esecuzione dell'algoritmo. Nel caso specifico dei dati del task, si è deciso di incrociare i risultati di una grid-search con quelli di una funzione per il calcolo dell'ottimo eps implementata in proprio, basata sull'uso dell'algoritmo Nearest-Neighbor, la cui implementazione è ampiamente utilizzata in letteratura. Infatti la presenza di due parametri da ottimizzare può produrre più di una configurazione accettabile e la condizione necessita di una strategia che serva a minimizzare l'intervento manuale nella scelta. La possibilità di incrociare i dati della griglia con il punto di massima curvatura ottenuto dalla funzione implementata, rende la scelta meno arbitraria e consente di avere una verifica sul campo dei risultati della griglia. Anche per il clustering con Dbscan si è sperimentato sia il clustering separato in C1 e C2 che quello congiunto.

7.3.2 I risultati

Dalle sperimentazioni complessive condotte sui dati del task si sono ottenuti i risultati riportati nella tabella riassuntiva mostrata in tabella 6. Quanto è emerso dall'elaborazione dei dati e dall'applicazione delle varie tecniche riguarda sia un giudizio di quali siano le strategie migliori da utilizzare nel condurre un task di quel tipo, sia le basi per impostare un metodo più generale, da riproporre per lo studio del cambiamento di senso in modo strutturale.

Metodo		**Accuracy**
Apd con distanza euclidea	**(metodo 1)**	0.77
Apd con distanza del coseno		<u>0.88</u>
Distanza di Hausdorff		0.55
Clustering		
K-means	**(metodo 2)**	0.44
K-means (metodo 2.1)		<u>0.72</u>
K-means (metodo 2.2)		0.66
Dbscan con distanza euclidea		0.61
Dbscan con distanza del coseno		0.55

Tabella 6: risultati ottenuti nel task DIACR-Ita

Conducendo il task come gli altri partecipanti si aveva a disposizione solo un piccolo numero di parole annotate per testare i metodi: *matematica*, *dettagliato*, *sanità*, *senatore*, *istruzione*. Di queste solo le ultime due positive al cambio di senso. La prova degli applicativi sui questi dati ha mostrato una buona performance del metodo 1 ma non altrettanto del metodo 2. Per questo motivo, in mancanza di altri riscontri, se si fosse partecipato al task si sarebbe concluso con i risultati indicati in tabella 6. Infatti, dalle sperimentazioni condotte su tutte le parole, APD_{cos} con distanza del coseno, è risultata la metrica migliore per misurare la distanza tra vettori di embeddings su quei dati. Con lo stesso approccio utilizzato da Wang et al. (2020) e indicato nella presentazione del task Basile et.al (2020) l'accuracy è stata poi calcolata dal

confronto dei risultati dati da ADP_{cos}, applicata ad ogni parola del task, rispetto ai risultati attesi, utilizzando la formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Il metodo 1, nella configurazione d'utilizzo con la sola APD_{cos}, ha raggiunto su tutto il task un'accuracy dello 0.88. In sostanza per 16 delle 18 parole del task sono stati ottenuti gli stessi risultati del gold standard, solo le parole *rampante* (indicato dal gold come positivo al cambio di significato) e *unico* (indicato come negativo al cambio di senso) non hanno ottenuto la classificazione corretta. Gli stessi dati sottomessi ai sistemi di clustering hanno invece prodotto risultati peggiori. Solo K-means con la strategia definita 'metodo 2.1' ha superato la baseline, con un'accuracy di 0.72.

7.3.2.1 I risultati post-task

La prima valutazione riguarda l'ordine di grandezza dei corpora sui quali si è condotto lo studio. Come per i corpora ILC-Ita anche per DIACR-Ita si è evidenziata una differenza notevole di dimensioni tra il corpus più cronologicamente distante (più piccolo) e quello più recente. L'ordine di grandezza di C2 era circa il triplo rispetto a C1, mentre ILC-Ita1 era più del doppio di ILC-Ita3. Questa situazione ha impattato sui metodi di clustering perché, essendo pensati prevalentemente per analizzare grandi dimensioni di dati, generalmente non hanno una buona resa quando questi sono pochi. Un caso per tutti è l'analisi della parola *trasferibile* che aveva 9 occorrenze in C1 e 58 in C2. Tutte le prove di clustering su questi dati, separando C1 e C2 hanno riportato risultati incoerenti, anche studiando il caso in C1+C2 la sola analisi automatica non ha prodotto risultati. AP con una regolazione fine del parametro *preference* ha prodotto i migliori risultati nel clustering, tuttavia non è riuscito a confermare la mancanza di cambio di senso per quella parola. La risposta poco efficace dei metodi che producono aggregazioni è sembrata una grossa limitazione che andava indagata e compresa.

L'analisi post-task dei risultati ha mostrato come per i dati DIACR-Ita, per i casi di parole con frequenza sopra una certa soglia (>=200) per ognuno dei corpora, l'applicazione del clustering sia con K-means che con Dbscan risulti più fine e sia in grado di indagare aspetti che il mero calcolo del coseno non riesce a cogliere. Si è quindi indagato se e come fosse possibile individuare tale soglia automaticamente, è infatti importante creare le condizioni di riproducibilità del metodo. Le variabili che concorrono alla buona riuscita del clustering possono essere molte e non è detto che su dati di piccole dimensioni non possano produrre soluzioni accettabili, la ricerca del minimo numero di occorrenze per assicurarsi un buon funzionamento del clustering dipende molto dai dati. Per quelli di DIACR-Ita la soglia è stata posizionata indicativamente intorno al valore di 200 perché per valori inferiori gli algoritmi di clustering utilizzati hanno avuto comportamenti difformi: numero di cluster molto diversi tra K-means e Dbscan, con Silhouette score generalmente bassi per i cluster generati da entrambi. Infatti l'accuracy del 0.72 ottenuto dal metodo 2.1 su tutto il task, probabilmente sarebbe stato più alto se le parole avessero avuto un numero maggiore di occorrenze. A dimostrazione di questo il metodo 2.1 sulle parole *pacchetto*, *campionato* e *unico,* che hanno occorrenze superiori a 200 sia in C1 che in C2, ha correttamente individuato una negatività al cambiamento d'uso. Provando a diminuire il valore della soglia, ponendola per esempio a 100, si aggiunge ai risultati positivi del metodo la parola *campanello* (con freq. 109 in C1 e freq. 628 in C2), ma rientrerebbero nel caso anche le parole *ape* e *discriminatorio* sulle quali però lo stesso metodo fallisce. Questo per dire come sia difficile individuare un comportamento univoco riproducibile per ogni set di dati. Alcuni partecipanti al task come Laicher et al. (2020) si sono interrogati sul perché hanno ottenuto risultati inferiori alle aspettative, andando ad analizzare le frasi del corpus e ravvisando la presenza di un contesto semanticamente non correlato, che potrebbe aver giocato un ruolo significativo nell'etichettare erroneamente le parole target, ma non è certo.

Il caso della parola *unico* è emblematico della differenza tra metodo 1 (senza aggregazioni) e metodo2 (clustring).

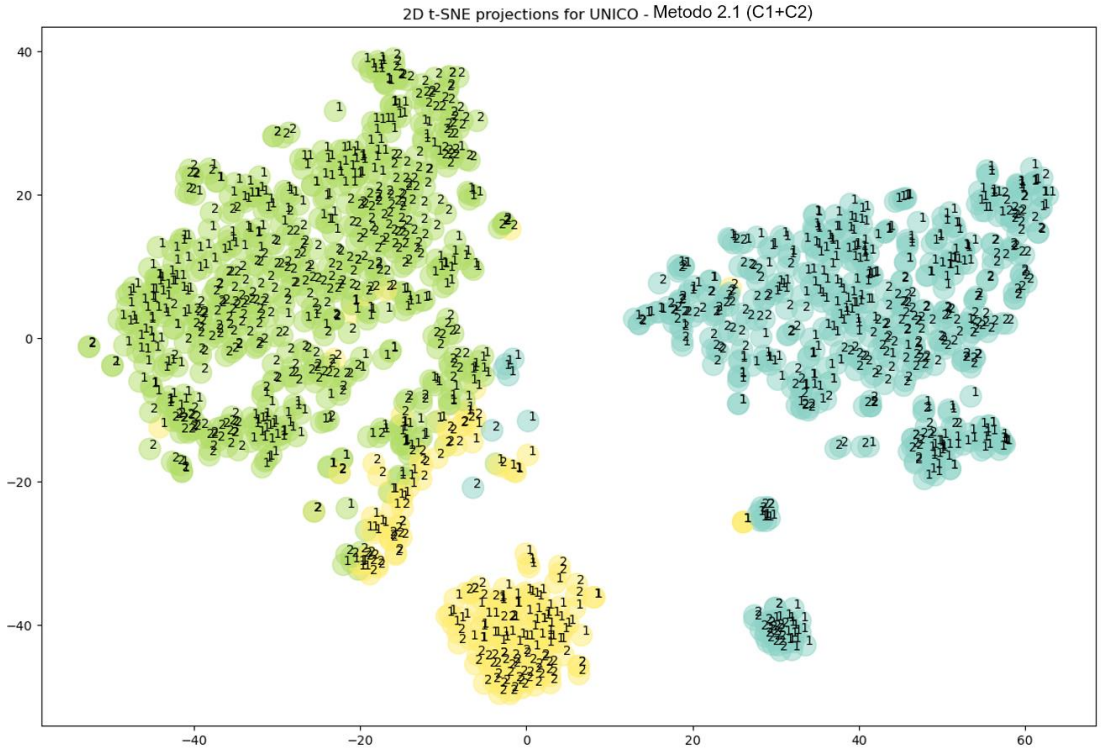

Figura 9: plot grafico della proiezione del clustering in C1+C2 della parola 'unico' (metodo 2.1)

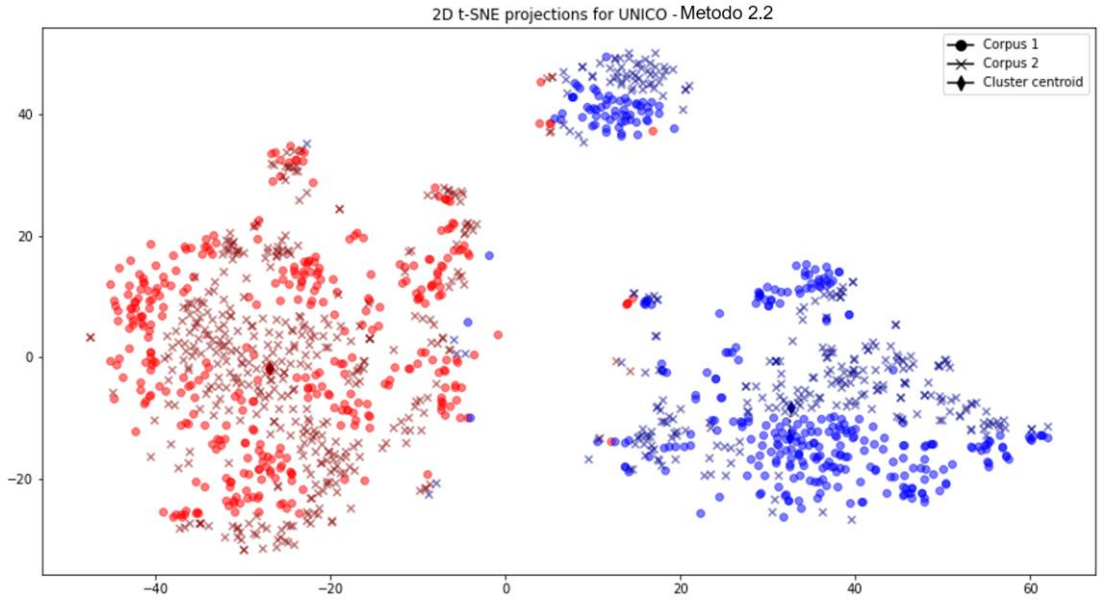

Figura 10: plot grafico della proiezione dei cluster di C2 su quelli di C1 della parola 'unico' (metodo 2.2)

Per questa parola infatti tutte le singole metriche di distanza tra vettori di embeddings (comprese nel metodo 1: senza aggregazioni) hanno indicato una positività al cambio di senso (vedi figura 8). In realtà la parola non subisce un cambio di senso, informazione che tutte le prove condotte con algoritmi di clustering, sia K-means che Dbscan, in tutte le configurazioni sperimentate (vedi figure 9 e 10), confermano all'unanimità. È sembrato quindi opportuno apportare delle modifiche alla strategia di studio del fenomeno, in particolare si è evidenziata la necessità di valutare più opzioni, modulare l'approccio, scomponendolo in sotto-attività.

Mentre per il trattamento di parole che hanno frequenze basse (sotto 100) in ognuno dei corpora in esame, risulta molto efficace utilizzare misure di distanza tra vettori, per le parole che ricorrono in modo frequente in entrambi i corpora, l'applicazione del clustering risulta uno strumento più idoneo. Con questa strategia di integrazione dei metodi anche la parola unico è stata classificata correttamente. Per confutare sperimentalmente questo risultato, ossia la negatività al cambiamento d'uso della parola, si sono indagati i dati più nel dettaglio, cioè analizzando le parole 'vicine' di quella target.

	C1				*C2*			
Parola target	**Parole vicine più prossime (contesto sinistro)**		**Parole vicine più prossime (contesto destro)**		**Parole vicine più prossime (contesto sinistro)**		**Parole vicine più prossime (contesto destro)**	
unico	**parola**	**freq.**	**parola**	**freq.**	**parola**	**freq.**	**parola**	**freq.**
	senso	24	modo	19	testo	15	modo	25
	fronte	21	mezzo	18	partito	14	paese	14
	testo	20	elemento	10	senso	10	scopo	10
	partito	18	punto	8	giudice	8	grande	8
	orario	10	classe	7	mercato	8	obiettivo	8
	candidato	7	scopo	7	contratto	7	grado	7
	prezzo	4	marcia	6	candidato	6	nazionale	7
	appare	3	superstite	6	turno	6	strumento	7
	articolo	3	ente	5	fondo	5	punto	6
	atto	3	leggi	5	sportello	5	elemento	5
	collocamento	3	obiettivo	4	forse	4	mondo	5
	contratto	3	organo	4	italia	4	risultato	5
	giuridico	3	ostacolo	4	amministratore	3	soggetto	5
	agente	2	paese	4	caso	3	candidato	4
	amministratore	2	progetto	4	collegio	3	genere	4
	aspetti	2	rappresentante	4	gruppo	3	partito	4
	blocco	2	risultato	4	lega	3	settore	4
	caso	2	scuola	4	pensiero	3	sicurezza	4

Figura 11: parole più frequenti che ricorrono nelle frasi della parola target 'unico', sia in C1 che in C2

Prendendo spunto dall'approccio di Gonen et al. (2020) si è condotta un'indagine specifica sui contesi della parola trovati nei due corpora. Si sono analizzate tutte le frasi contenenti la parola *unico* estratte sia da C1 che da C2 e si è individuato un intervallo attorno alla parola composto di sole parole 'piene' (si sono cioè eliminate le parole vuote: articoli, preposizioni, congiunzioni e interiezioni). Nell'analisi sono state prese 4 parole piene precedenti la target e 4 successive. Dall'intervallo sono state escluse anche le forme dei verbi ausiliari per aumentare il valore del contributo fornito dalle altre parole piene. L'analisi ha mostrato con evidenza quanto le stesse parole ricorressero nell'immediato intorno della parola nelle frasi di C1 e C2. La figura 11 mostra quelle di frequenza maggiore, che sono comunque sufficienti a confutare l'assenza di un significato diverso nell'arco temporale indagato.

Una considerazione ulteriore è stata fatta per quei casi, in cui si riscontra una differenza di 'omogeneità' nelle attestazioni tra C1 e C2. Con questo si intende indicare le parole che presentano frequenze molto sbilanciate/difformi tra i corpora, per esempio quando le occorrenze di una parola in C2 sono decisamente maggiori rispetto a quelle in C1. In questi casi è accaduto che sia il metodo 1 che il metodo 2.1 non abbiano colto il cambio di significato/d'uso. È il caso della parola *rampante*, di cui sarà discusso in dettaglio nel cap. 8, per la quale l'applicazione sia del metodo 1 che di quello 2.1 avevano fallito. In particolare il metodo 2.1 aveva mostrato un'unica occorrenza di C1 nel cluster n.1, unita alle 191 di C2 (vedi tabella 7), indice di un comportamento limite che meritava un'indagine più approfondita.

K-means	**C1**				**C2**				**C1+C2**		
	Metodo 2.2		**Metodo standard**		**Metodo 2.2**		**Metodo standard**		**Metodo 2.1**		
parola	***n. clu***	***Coef. Silh.***	***n. clu.***	***Coef. Silh.***	***n. clu.***	***Coef. Silh.***	***n. clu.***	***Coef. Silh.***	***n. clu.***	***Distrib. Pti in cluster***	***Coef. Silh.***
rampante	2	0.498	3	0.260	2	0.123	3	0.100	2	1: (1,191), 2: (31,215)	0.094

Tabella 7: risultato del clustering con K-means della parola 'rampante'

Per questi casi, relativamente frequenti sia in DIACR-Ita che in ILC-Ita, si è deciso di utilizzare anche il metodo 2.2, con produzione grafica dei cluster, nel tentativo di analizzare meglio la distribuzione degli embeddings nei cluster. I risultati mostrano un ulteriore miglioramento del metodo che ha confermato la positività di *rampante* al cambio d'uso. L'uso incrociato di tutti i metodi e il controllo della distribuzione degli embedding nei cluster può offrire uno strumento di analisi molto efficace e nel caso specifico del task risolutivo. Data la natura post-task di queste valutazioni non si può a ragione ritenere che senza i dati gold l'approccio generale avrebbe da solo prodotto la corretta valutazione di tutte le parole, ma si considera comunque un risultato utile a comprende come migliorare le strategie di impostazione del metodo. Una sintesi delle sperimentazioni fatte legate alla dimensione delle frequenze delle parole nei corpora e alla relativa specializzazione del metodo è stata riassunta nella tabella 8.

Casi di applicazione prevalente	Combinazione metodi
parole sia in C1 che C2 con: freq. <100	APD con distanza del coseno
parole sia in C1 che C2 con: 100=< freq.<=200	APD con distanza del coseno + K-means (metodo 2.1) + K-means (metodo 2.2)
parole sia in C1 che C2 con: freq. >200	K-means (metodo 2.1) + K-means (metodo 2.2)

Tabella 8: specializzazione del metodo in DIACR-Ita

7.3.3 Il grado di cambiamento del significato

Il metodo di clustering, essendo una misura più fine della distanza tra embeddings, è anche utilizzabile per misurarne il grado di cambiamento del significato. Purtroppo il task di DIACR-Ita non prevedeva linee guida per condurne lo studio come in SemEval. Per calcolare il grado di cambiamento si

sfruttano le informazioni sull'uso delle parole estratte dai diversi intervalli di tempo per dedurre due tipi di informazioni: (i) quanto è variato l'uso di una data parola nell'arco temporale analizzato; (ii) quali sono gli usi coinvolti e come interpretarne il cambiamento. La metrica utilizzata dipende dal tipo di informazioni estratte dalle immersioni contestualizzate per una data parola target. Possono essere matrici grezze di embedding contestualizzati $M_{emb}^{(t)}$, immersioni medie $e_{mean}^{(t)}$ per ogni intervallo di tempo (entrambi ottenibili con l'applicazione del metodo 1), o distribuzioni di cluster $c^{(t)}$. Le matrici degli embedding contestualizzati possono essere confrontate utilizzando la distanza media a coppie con APD e distanza del coseno, rivelatasi affidabile per parole di dimensioni contenute, oppure utilizzando le immersioni medie per i due intervalli di tempo, che possono essere confrontati anch'essi solo con la distanza del coseno. Per i metodi di aggregazione dei cluster, la divergenza tra le distribuzioni dei cluster può essere misurata con la divergenza di Jensen-Shannon (JSD)[59]. Nel caso specifico convertendo le distribuzioni degli embeddings nei cluster, nelle corrispondenti distribuzioni di probabilità. In generale è possibile configurare lo studio impostando la metrica $d(t_1,t_2)$ configurata come segue:

$$d(t_1,t_2) = \begin{cases} APD\ (M_{emb}^{(t^1)}, M_{emb}^{(t^2)}) \\ \cos(e_{mean}^{(t^1)}, e_{mean}^{(t^2)}) \\ JSD(c^{(t^1)}, c^{(t^2)}) \end{cases}$$

Nella particolare condizione del task, ma anche con i dati di ILC-Ita, i metodi indicati hanno ottenuto performance diverse. Come descritto in precedenza le prestazioni migliori sono state ottenute con l'approccio indicato come metodo 1, cioè utilizzando il calcolo di APD direttamente sugli embedding estratti. In

[59] Nella teoria della probabilità e nella statistica, la divergenza di Jensen-Shannon è un metodo per misurare la somiglianza tra due distribuzioni di probabilità. Si basa sulla divergenza Kullback-Leibler, con alcune differenze notevoli (e utili), incluso il fatto che è simmetrico e ha sempre un valore finito. La radice quadrata della divergenza Jensen-Shannon è una metrica spesso indicata come distanza Jensen-Shannon.

considerazione delle caratteristiche dimensionali descritte in 7.3.2. la metrica migliore è sembrata essere APD, tuttavia se ne è condizionato l'uso per calcolare il grado di cambiamento ai casi in cui anche altre misure fossero indicatori di maggiore differenza tra C1 e C2. Le condizioni imposte hanno indicato il grado di cambiamento maggiore per la parola *tac*. Infatti ha presentato la deviazione standard massima, il punteggio più alto in APD con distanza del coseno, la distanza di Hausdorff maggiore e anche calcolando APD con distanza euclidea, si è trovato per la parola un valore tra i più alti (vedi figura 12 ed estratta dalla figura 8)

					Distanza euclidea			Distanza del coseno
Parola target	Dim. frasi C1	Dim. frasi C2	Distanza euclidea embC1 embC2 medi	Dev. standard	APD con distanza euclidea	Distanza di Hausdorff	ADP con distanza del coseno **(4L)**	ADP con cosine similarity **(UL)**
tac *	139	436	10.24	3.03	**18.25**	***18.29***	**0.65**	**-0.0004**

Figura 12: metriche calcolate per la parola 'tac'

7.4 Applicazione del metodo ai dati ILC-Ita

Dopo le analisi condotte sui dati DIACR-Ita si è deciso di sperimentare lo stesso approccio anche sui dati ILC-Ita, più connotati diacronicamente, seguendo la stessa procedura implementata per il task DICAR-Ita, in particolare utilizzando ILC-Ita3 come C1 e ILC-Ita1 come C2. Sfortunatamente le stesse parole del task non si attestavano nei dati. L'esame condotto sui corpora ha mostrato come solo una minima parte delle parole del task DIACR-Ita fosse presente nel corpus ILC-Ita3: *brama*, *pilotato*, *ape*; mentre in ILC-Ita1 queste stesse parole non si attestassero, ad esclusione di *brama*, con frequenze però non significative. Questa sorta di disallineamento tra C1 e C2 è una caratteristica importante per valutare una metodologia di confronto, non solo perché non si disponeva di dati etichettati per valutare il sistema, ma anche perché la distanza linguistica dei dati rende difficile trovare una strategia efficace. Con questo obiettivo si è studiato il lavoro di De Stefanis

Ciccone et al. (1984), per trovare eventuali analisi approfondite dei dati che potessero dare indicazioni per lo studio da condurre. La natura del lavoro, che era mirato alla lemmatizzazione, non ha però offerto indicazioni utili. Si è provato anche ad indagare nel successivo lavoro di codifica dei testi in formato XML TEI. In quel caso ILC, in collaborazione con l'Accademia della Crusca di Firenze ha studiato il mapping di conversione dei dati, ma anche in questo caso il lavoro non è sceso nell'analisi delle parole.

7.4.1 La strategia per l'indagine sulle parole

Un modo per affrontare il fenomeno parte dalla scelta della parole. Una strategia possibile era partire dalle parole del dizionario che si attestavano in entrambi i corpora C1 e C2, in alternativa era possibile estrarre tutte le parole da uno dei due corpora e confrontarle con quelle dell'altro e così è stato fatto. Nel caso specifico del corpus ILC-Ita3 ne sono state estratte 28.221. Per condurre l'esperimento da queste ne sono state scelte 100 fra le più usate, un ulteriore confronto teorico con linguisti di ILC ha poi permesso di ipotizzare un primo insieme di parole che potessero servire al rilevamento positivo del cambio di significato/uso: *abuso*, *album*, *ambiente*, *amica*, *applicazione*, *arena*, *attore*, *campagna*, *campo*, *casino*, *cellulare*, *cittadino*, *diritto*, *fabbrica*, *malattia*, *mercato*, *sistema*, *sociale*, *squallore*, *tramandare*. Il set di parole era stato scelto in relazione agli aspetti d'interesse linguistico che sarebbe stato utile indagare in un approccio diacronico. La collaborazione con i linguisti ha messo in luce alcuni aspetti significativi da indagare: il cambio di significato dovuto al progresso tecnologico (*album*, *applicazione*, *cellulare*) o culturale (*abuso*, *ambiente*, *cittadino*, *diritto*, *fabbrica*, *malattia*), lo sviluppo di sensi figurati delle parole con conseguente allargamento degli usi (*arena*, *attore*, *campagna*, *campo*, *mercato*, *sistema*, *sociale*), il cambio di accezione da valore positivo a negativo (*amica*, *casino*). Di tutte queste parole si è prodotta una selezione finale che comprendesse il maggior numero di parole di tutte le tipologie sopra indicate e che fossero attestate sia in C1 che in C2. Il

set di parole sul quale è stata condotta la sperimentazione è risultato quindi: *abuso*, *album*, *amica*, *applicazione*, *arena*, *attore*, *campagna*, *campo*, *cellulare*, *cittadino*, *diritto*, *fabbrica*, *malattia*, *mercato*, *sistema*, *sociale, tramandare*. Procedendo come già fatto per DIACR-Ita, dopo la selezione delle parole target si sono estratti i WE dai relativi corpora di frasi di C1 e C2.

7.4.2 L'esperimento su dati ILC-Ita

Come per il task DIACR-Ita si sono seguiti i seguenti passi:

1. Si è adottato lo stesso modelli BERT per l'italiano ed è stato prodotto il fine-tuning sia su ILC-Ita1 (C2) che per ILC-Ita3 (C1), secondo quanto descritto nel cap. 6.1.3.
2. Con una fase di pre-processing si sono prodotti i corpora di frasi di tutte le occorrenze delle parole per entrambi i corpora, similmente a quanto fatto per il task DIACR-Ita;
3. Si sono estratti i word embedding in contesto di tutte le parole, per ognuno dei due corpora. L'estrazione ha riguardato la media degli ultimi quattro livelli di hidden della rete, rivelatesi la combinazione migliore. Si sono così prodotte due liste di vettori di immersioni che sono stati sottoposti al metodo di studio individuato.

Applicazione del metodo (1 e 2)

4. Si è implementato un confronto tra i word embedding basato su APD con distanza del coseno (metodo 1 senza aggregazioni). Secondo il criterio di valutazione delle dimensioni delle occorrenze delle parole adottato in DIACR-Ita, si è cercato il valore della soglia adeguata anche per i dati ILC-Ita. In prima istanza si è cercata una proporzionalità tra le dimensioni del corpus e quella delle occorrenze, una regola che potesse essere riproposta sistematicamente. Varie prove hanno mostrato che il limite minimo di 100 è legato all'applicazione della metrica APD_{cos} più che al rapporto tra frequenza di attestazione di una parola e dimensione del corpus. Per questo motivo anche per i dati ILC-Ita si è riservata l'applicazione del metodo 1

(senza aggregazioni) alle parole la cui occorrenza fosse minore di 100 in entrambi i corpora C1 e C2.

5. A quelle con frequenza superiore a 100 sia in C1 che in C2 si sono applicati gli algoritmi di clustering: K-means in configurazione (*metodo 2.1*), risultata la migliore, nella configurazione definita *metodo 2.2* e infine Dbscan con distanza euclidea.

7.4.3 I risultati

La sperimentazione non ha potuto avvalersi dei dati gold disponibili per il task di DIACR-Ita, motivo per il quale sui risultati è stato possibile fare solo delle considerazioni generali. Dall'applicazione dei metodi si sono prodotti i risultati mostrati nella tabella 9 riportata di seguito. Nelle colonne 2 e 3 della stessa sono riportate le frequenze delle parole nei rispettivi corpora. Nella colonna 4 è indicato il risultato dell'applicazione del metodo 1 in configurazione APD_{cos}; nella colonna 5 è invece riportato il numero di cluster prodotti dal metodo 2.2. In particolare vengono indicati con:

- *'numero cluster': (C1 e C2)* il cluster condiviso tra occorrenze della parola in C1 e C2;
- *'numero cluster': isolato*, nei casi in cui l'algoritmo ha prodotto cluster appartenenti ad uno solo dei due corpora.

Nell'ultima colonna è riportata la positività al cambiamento di significato/uso. Analizzando i risultati ottenuti si può comprendere quali metodi sono stati applicati e perché. Le frequenze di parole come: *campo*, *campagna*, *diritto*, *sistema*, *mercato*; sono maggiori delle altre ma in generale sono tutte contenute. Questa condizione ha suggerito l'utilizzo del metodo 1 (senza aggregazioni) per quasi tutte le parole (ad esclusione di *diritto*) poiché la loro frequenza nei corpora non superava le 100 occorrenze. L'esperienza maturata con il task DIACR-Ita ha però evidenziato come dimensioni molto differenti tra attestazioni in C1 rispetto a C2 rappresentasse un problema nell'uso del solo metodo 1. L'indagine si è quindi avvalsa anche del metodo 2.

I risultati mostrano un cambiamento di significato/d'uso confermato da entrambi i metodi per *amica* e *campo*. Com'era prevedibile si sono riscontrati anche risultati diversi tra metodo 1 e 2. In considerazione dei criteri di dimensionalità ridotta delle occorrenze (sotto il valore di 100), descritti in precedenza, si è confermata la positività al cambiamento individuata dal calcolo di APD_{cos} per *applicazione*, *attore* e *cellulare*, anche se il metodo 2.2 non aveva individuato per queste parole cluster isolati né per C1 né per C2. Con il criterio legato invece alla grande difformità tra dimensioni di attestazioni tra C1 e C2 (§ 7.3.2.1), si è rifiutata quella per *campagna*, che invece APD_{cos} aveva indicato. Per lo stesso criterio anche le parole *mercato* e *sistema* si sono aggiunte alla positività al cambiamento semantico. Si è cercato comunque di trovare conferma dei risultati ottenuti.

parola	**Fr. in C1**	**Fr. in C2**	**Metodo 1**	**Metodo 2.2 n. cluster**	**LSC**
abuso	25	57	0.4553	1: (C1 e C2)	0
album	7	100	0.4552	1: (C1 e C2) 1: isolato	0
amica	17	44	0.690	1: (C1 e C2) 2: isolati	1
applicazione	29	133	0.541	1: (C1 e C2)	1
arena	18	30	0.379	1: (C1 e C2)	0
attore	60	140	0.590	1: (C1 e C2)	1
campagna	49	360	0.502	1: (C1 e C2)	0
campo	99	700	0.558	1: (C1 e C2) 2: isolati	1
cellulare	6	20	0.501	1: (C1 e C2)	1
cittadino	52	100	0.410	1: (C1 e C2)	0
diritto	111	419	0.497	1: (C1 e C2)	0
fabbrica	19	81	0.457	1: (C1 e C2) 1: isolato	0
malattia	40	110	0.413	1: (C1 e C2)	0
mercato	40	683	0.465	1: (C1 e C2) 1: isolato	1
sistema	65	719	0.476	1: (C1 e C2) 1: isolato	1
sociale	43	411	0.419	1: (C1 e C2)	0
tramandare	2	1	-	-	-

Tabella 9: risultati ottenuti sul corpus ILC-Ita

La parola *amica* era stata scelta a ragion veduta, proprio perché si sapeva della diffusione a partire dal Novecento di una sua accezione di valore negativo (Biffi et al. 2022)[60]. In particolare si intende un maggior uso del senso di amico/a da positivo, legato al significato della parola amicizia, ad uno più negativo, dove *amica* si attesta in opposizione a moglie, e quindi con il significato di 'amante'. Studi lessicografici sui dizionari storici della lingua italiana hanno mostrato che, pur tenendo conto dei diversi contesti storico-culturali sui quali si modella diversamente il senso della connotazione negativa osservata, questa nuova accezione diventa più frequente nel Novecento, affiancandosi al significato neutro. Anche *attore, campo* e *mercato*, assumendo valore figurato, trovano dalla seconda metà del Novecento maggiori contesti d'uso in cui assumono significati diversi. Sono stati infatti scelti in ragione del fatto che il corpus ILC-Ita, per la sua composizione, contiene una grande quantità di articoli di tipo economico e politico, nei cui contenuti si attestano in gran quantità sensi figurati legati a queste parole: un esempio per tutti è la 'discesa in campo' di Berlusconi. Queste parole assumono aspetti figurati sia in articoli a tema politico che in quelli di ambito economico, ma sono anche largamente usate negli articoli sportivi, argomento che nei quotidiani ottocenteschi ha mostrato una presenza irrilevante.

Volendo fare un'analisi più approfondita si è provato a vedere quali embedding fossero stati raggruppati dagli algoritmi di clustering per queste parole. Tra i risultati attesi troviamo per esempio le parole *mercato* e *campo*, che erano state indicate come più probabili a nuovi usi figurati. Avendo una buona frequenza sia in C1 che in C2 gli algoritmi di clustering hanno evidenziato per queste parole, cluster presenti solo nel corpus più vicino temporalmente. Per *mercato* sono stati individuati più usi figurati che sono rimasti comunque condivisi tra C1 e C2. Per esempio in C1 troviamo: "*Il* ***mercato*** *continua fermo alle quotazioni di sabato, ma con limitata domanda.*"; "*Torino, 17 gennaio. Poco o nulla abbiamo a sognare*

[60] Biffi, Marco, Francesca De Blasi, Manuel Favaro, Elisa Guadagnini, Simonetta Montemagni, Eva Sassolini. Parole in rete / reti di parole. Possibili impieghi didattici dei grandi vocabolari storici digitalizzati. Italiano a scuola. ASLI. 2022. – L'articolo non è ancora stato pubblicato.

sull'andamento del nostro ***mercato*** *del vino nella settimana dal 10 al 16 gennaio*.". Mentre in C2 sempre nello stesso cluster sono presenti embeddings relativi a frasi quali: "*Secondo la filosofia della compagnia assicurativa tedesca, la banca d'investimento avrebbe dovuto essere una finestra sul* ***mercato*** *e una fonte di prodotti strutturati*."; "*Il* ***mercato*** *dei piccoli appalti ormai è crollato", ammette Ugo Argiroffi dell'Ance Palermo*"; "*Continua in sede tecnica il confronto a Palazzo Chigi sugli strumenti per rendere più flessibile il* ***mercato*** *del lavoro*."; "*Così come aumentano le ragazze che appendono lauree e diplomi al chiodo e decidono di investire nel* ***mercato*** *delle unghie*." Il clustering produce anche un cluster in cui mercato trova un uso legato alle tecnologie e che correttamente è stato identificato solo in C2: alcune frasi relative al cluster sono: "*Anche i giganti consolidati del settore ora si sono svegliati e premono con investimenti di milioni di marchi sul* ***mercato*** *online*."; "*Come se la necessità di stare sul* ***mercato*** *mondiale delle comunicazioni con gruppi di adeguate dimensioni non fosse una necessità dell'Italia"*."; "*Grande l'entusiasmo dei fan Apple (e non solo), fin quando l'azienda di Cupertino non ha comunicato il listino per il* ***mercato*** *europeo:…*"; "*Il* ***mercato*** *consumer è inoltre ancora in attesa di programmi (applicativi) cruciali in grado di convincere l'utenza all'acquisto*."

Lo stesso tipo di analisi è stata condotta per la parola *campo*, che vede usi figurati o meno, comunque condivisi, per esempio: "*Si crede però che alcuni intervalli di buon tempo possano lasciar* ***campo*** *di raccogliere i primi raccolti di questa stagione*;"; "*Ne' secoli immenso* ***campo*** *alle imprese marittime, molte e diverse spedizioni furono dirette verso il polo artico*"; "*Ma quando si devono abbattere gli steccati che angustiano il nobil* ***campo*** *dell' arte, non monta con che povero mezzo lo si consegua*." In C1. In C2 lo stesso cluster riporta occorrenze quali: "*Baricco, nella sua postfazione alla nuova versione dell'Iliade, lascia fuori dal* ***campo*** *di battaglia gli dèi, e così non coglie questi legami*,"; "*Adesso, il nostro interlocutore è il* ***campo*** *"*."; "*IRENE PIVETTI ha ragione quando denuncia, come ha fatto sabato a Bologna, la mancanza di regole, il vero e proprio Far West che vige nel* ***campo*** *della maternità assistita*."; "*Il* ***campo*** *della prevenzione degli infortuni e della*

sicurezza in fabbrica, infatti, è in continua evoluzione". Ma anche in questo caso il clustering ha prodotto un cluster presente solo in C2 che identifica bene l'uso della parola *campo* esclusivamente in ambito sportivo: "*Spalletti cerca di spiegare il repentino cambiamento ma anche di contentarsi: "Abbiamo giocato giovedì su un* ***campo*** *terribile ed era normale un calo.*"; "*Tragedia in* ***campo*** *durante Mirano-Monselice, match di rugby di serie B: Simone Franchini, 28 anni, mediano della squadra ospite, è morto dopo stato colpito da un arresto cardiaco.*"; "*Trapattoni 3: "Il mio è un gruppo non solido psicologicamente, voglio che cresca il dialogo, i giocatori in* ***campo*** *devono parlarsi di più.*"; "*Vista la collaborazione di società e tifosi, si esclude una stangata per l'Ascoli: pena più probabile uno o due giornate di squalifica del* ***campo*** ."

Il fatto che il metodo sia stato in grado di cogliere queste evidenze rappresenta una conferma di quanto già sperimentato nel task DIACR-Ita.

Esistono poi casi la cui analisi richiede una valutazione più fine. Per esempio, sempre per la parola *campo* è stato prodotto dal clustering un ulteriore piccolo cluster presente solo in C2 ma dal raggruppamento meno intuitivo. Il contenuto è sempre sportivo e del tutto simile a quello del cluster illustrato sopra, ma in cui ricorrono frasi lunghe del tipo: "*Dopo l'ultimo 0-0 anestetizzato dalla paura,…*"; "*All'Olimpico 1-1 nel derby: una monetina lanciata ….*"; "*E in A2 il Fasano facendo risultato…*"; "*In 61 minuti nelle due partite, 4/16…*". La separazione di questi embeddings dagli altri dello stesso argomento fa pensare al riconoscimento di un attacco di frase simile, come se per gli embeddings estratti da queste frasi si fosse prodotto uno spostamento su altre caratteristiche che vanno oltre la parola target.

Questo comportamento ha consigliato l'uso anche dell'altro metodo di clustering (metodo 2.1) per confutare il risultato. La figura 13 mostra la proiezione grafica dell'applicazione del clustering, fatto questa volta sull'intero corpus di frasi (C1 + C2). L'immagine mostra quattro cluster di cui tre condivisi tra C1 e C2 (le occorrenze di C1 nei 3 cluster sono state evidenziate con una linea rossa tratteggiata) e un solo cluster costituito di soli embeddings di C2

(quello i cui punti sono di colore verde). In questo caso il clustering sull'intero corpus non conferma un diverso uso di *campo* in C2 se non quello legato all'ambito sportivo.

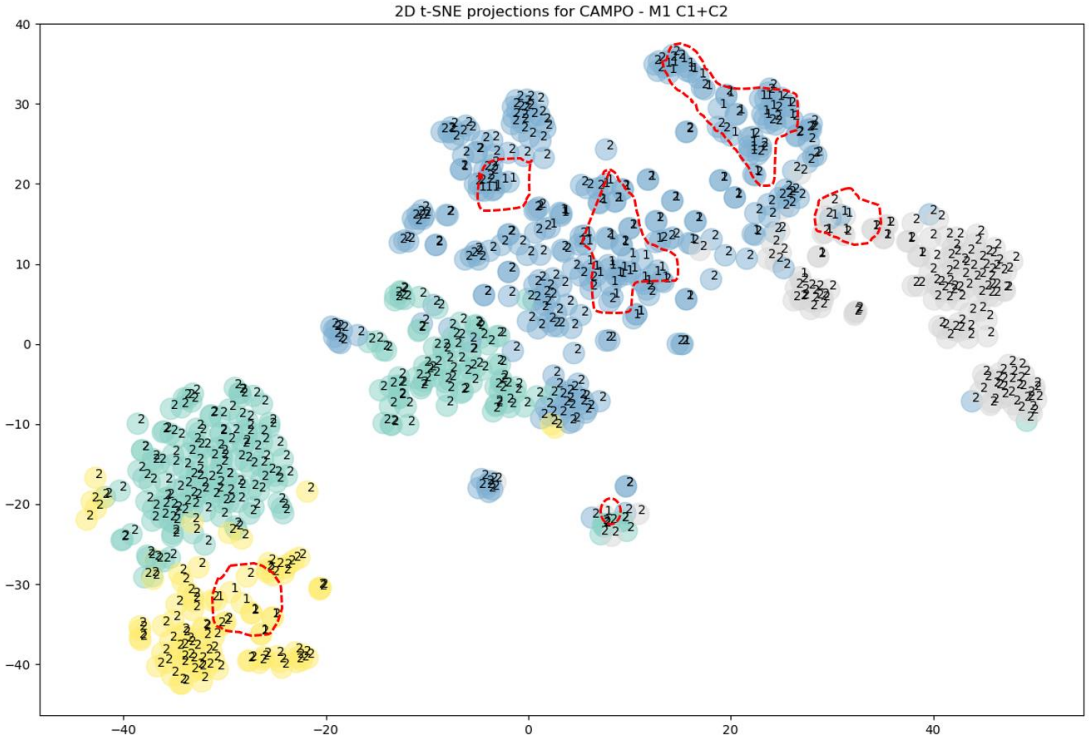

Figura 13: plot grafico della proiezione del clustering in C1+C2 della parola 'campo' (metodo 2.1)

Un'altra parola che ha dato risultati diversi dagli attesi è *cellulare* per la quale ci si sarebbe attesi un marcato cambio d'uso perché in ben 4 delle 6 frasi di C1 la parola è usata come aggettivo di 'tessuto' (... *una specie di tessuto* ***cellulare****; ...della pelle e del tessuto* ***cellulare****; ...induramento del tessuto* ***cellulare****; ..disseminati nel tessuto* ***cellulare***...), mentre in C2 ci si riferisce quasi esclusivamente al telefono cellulare (*legge sul* ***cellulare***...; *...i tasti del* ***cellulare*** *con quelle dita*...; *Chi cambia così facilmente* ***cellulare***?..; *...numero del* ***cellulare***..). L'esiguo numero di contesti (6 in C1 e 20 in C2) ha consigliato l'uso della misura $APD_{cos,}$ che ha comunque confermato la positività ma con un piccolo margine. Probabilmente una maggiore frequenza avrebbe reso applicabile il clustering e più evidente il cambio di significato, che in questo caso rappresenta il significativo passaggio da aggettivo a sostantivo.

Anche *amica* ha dato risultati diversi da quelli attesi, in C1 troviamo la parola ricorrere prevalentemente nella descrizione di trame di commedie di cui si annuncia il debutto: "*Il Poeta francamente risponde: Mercè l' ajuto d' un' **amica** mano.*"; "*una buona vedova, **amica** di casa, e che sposa, alla fine della commedia*", ecc. che sono quindi in netto contrasto con l'uso che viene fatto della parola in C2 dove ricorre prevalentemente in articoli di cronaca: *""Mafioso, mafioso", gli grida in faccia lui mentre un' **amica** se lo porta via*"; "*Piangono l'**amica** morta, mostrano le foto…*"; "*Paola mostra la borsetta dell'**amica** uccisa…*"; "*Morì in casa di un'**amica**, forse nel suo letto.*" Gli embeddings hanno quindi catturato correttamente il diverso uso e le metriche sono state in grado di misurare tale shift semantico. Tuttavia tra le frasi del corpus ILC-Ita3 era presente anche l'accezione negativa che si pensava fosse di uso più attuale, una sola frase ma comunque indicativa: "*E i miei pensieri volano a te, siccome quelli dell' amante all' **amica** lontana*;"

Questo per evidenziare come è insidioso lo studio di questo fenomeno e come la realtà che emerge dai dati richieda una valutazione attenta. Le sperimentazioni condotte in gran parte confermano quanto sostenuto dai linguisti, ossia che il cambiamento di significato delle parole nel tempo è un processo lungo, che va osservato in archi temporali molto lunghi. Quando si affrontato periodi brevi il cambiamento semantico è quasi sempre un cambio d'uso di tipo tecnologico e culturale, spesso un allargamento dell'uso dovuto nuovi contesti presenti nei testi. Le due parole *campo* e *mercato* ne sono espressione evidente. La seconda aumenta/moltiplica il suo uso nel nuovo contesto tecnologico e culturale dovuto al progresso. Per *campo* l'uso sportivo del termine probabilmente non è nuovo, ma nuovi/maggiori sono i contesti scritti in cui si attesta, rispetto al corpus ILC-Ita3 in cui gli articoli sportivi non avevano spazio.

Per quanto lo studio e la scelta delle parole siano stati mirati alla ricerca di parole con alta probabilità di positività al cambiamento di significato il lavoro non può ritenersi esaustivo. Uno studio sistematico delle parole che subiscono un cambiamento d'uso in periodi di tempo diversi è un obiettivo

interessante e auspicabile, ma senza una validazione dei metodi d'indagine diventa arduo, infatti il cambiamento semantico non è un'informazione nota a priori. Per realizzare tali metodi è necessaria la disponibilità di dati annotati: gold standard appositamente creati per la validazione dei metodi. Grazie ad iniziative come DIACR-Ita il lavoro di tesi ha potuto verificare le ipotesi proposte sui dati, cosa che si è rivelata estremamente importante. Si auspica che la comunità scientifica operi in questa direzione, solo in questo modo infatti, esperimenti promettenti potranno diventare strumenti di studio efficaci.

Capitolo 8

Analisi finali e valutazioni

In questo capitolo si analizzano i risultati ottenuti dal metodo di rilevamento del cambiamento semantico nel suo complesso, se ne discutono le caratteristiche e se ne esaminano limiti e prospettive. Si ritiene infatti che il principale vantaggio dell'approccio sia la sua semplicità e flessibilità, date da un unico tipo di rappresentazione delle parole, ma diversi algoritmi/strumenti di studio di tali rappresentazioni. Viene proposta di seguito un'analisi qualitativa del clustering con l'obiettivo di individuare un metodo applicabile al cambiamento semantico che sia configurabile a qualsiasi granularità temporale. Sia Martinc et al. (2020a) che Kutuzov e Giulianelli (2020) hanno affermato che le variazioni contestuali delle rappresentazioni BERT sono altamente sensibili e quanto tale sensibilità al contesto, seppur chiaramente positiva per lo studio del fenomeno, possa rendere più difficile ottenere aggregazioni significative di usi, soprattutto in relazione alla loro interpretazione.

8.1 Il clustering

Nell'ambito della rilevazione del cambiamento di significato di parole tra due periodi temporali, nei casi di parole a bassa frequenza, che producono un piccolo numero di embeddings, l'uso del clustering si dimostrato nei fatti poco

efficace, perché la loro elaborazione con algoritmi di clustering tende a interpretare come sensi diversi, altre caratteristiche di tipo sintattico. Ovvero quando le frequenze sono contenute gli algoritmi di clustering tendono a valutare differenze più sottili. Un caso per tutti è AP che producendo un alto numero di cluster arriva a separare embeddings che differiscono per elementi di composizione della frase. Per esempio la parola trasferibile del task DIACR-Ita ha freq. 9 in C1 e freq. 58 in C2, l'esecuzione di AP sui relativi embedding produce il seguente log dei risultati:

```
Embeddings della parola: trasferibile
Estimated number of clusters in C1: 3
n. iterations: 60
index centers
[1 3 6]
Silhouette Coefficient: 0.612
Estimated number of clusters in C2: 9
n. iterations: 67
index centers
[ 2 14 15 19 20 21 47 54 57]
Silhouette Coefficient: 0.302
```

I dati di log mostrano che per 9 embeddings vengono prodotti 3 cluster. Risulta evidente che il raggruppamento tenga conto di differenze sintattiche. Come già riportato nel cap. 7.3.2.1 su questo caso si è provato a raffinare il clustering agendo sul parametro preference. Si è potuto quindi 'sindacare' sul raggruppamento e provare a contenere il numero di cluster. Un lavoro impegnativo e non risolutivo.

Occorre inoltre considerare che alcuni algoritmi richiedono inizializzazioni più o meno raffinate. Nel caso di K-means per esempio, i criteri di selezione del numero dei cluster partono comunque da un numero minimo di due. Il metodo che lo adotta è talvolta costretto a costruire partizioni multiple per parole non ambigue, secondo caratteristiche difficilmente interpretabili. Dbscan invece, in presenza di parole non ambigue, può prevedere assenza di cluster (solo un insieme di outlier), un singolo cluster o più di uno. Può inoltre trattare i punti isolati o outliers, comportamento questo

più desiderabile in assenza di cambio di senso. Inoltre Dbscan trova cluster di qualsiasi forma, al contrario di K-mean che presuppone che i cluster abbiano una forma convessa. Per contro Dbscan richiede due parametri da impostare (§ 7.3.1.2), condizione che impone la valutazione di molte combinazioni di valori e la scelta tra più alternative possibili, ritenute ugualmente valide dal metodo di confronto. Questa caratteristica aumenta il grado di incertezza sull'esito di una procedura di ricerca completamente automatica.

L'individuazione del corretto processo di generalizzazione del metodo è uno sforzo che si scontra con queste caratteristiche peculiari dei vari approcci, ma trovare una soluzione è necessario per offrire al metodo di studio una prospettiva reale di scalabilità. Dai risultati ottenuti si può dire molto sul percorso da fare per raggiungere tale obiettivo.

8.2 Analisi complessiva dei risultati

La difficoltà più grande nel trattamento dei dati di DIACR-Ita e ILC-Ita si è riscontrata per quelle parole che avevano una frequenza maggiore in un corpus rispetto all'altro. Come già introdotto al termine della valutazione dei risultati del task DAICR-Ita, il caso della parola rampante era emblematico: parola che si sapeva essere positiva al cambio di significato, ma per la quale le occorrenze in C1 erano 32 mentre in C2 406. La disparità di attestazione aveva dimensione maggiore in rapporto ad ogni altra parola del task positiva al cambio di significato. Per quella parola la misura APD non ha prodotto una positività e da questo è stata condotta un'indagine più approfondita. Lo studio ha riguardato tutte le parole che presentavano questa caratteristica: *brama*, *cappuccio*, *pilotato*, *piovra*, *processare*, *rampante* e ha indirizzato il metodo d'indagine verso un'analisi fine, che è stata definita 'metodo 2.2' nel capitolo 6.2. Uno strumento, mutuato dall'approccio Kanjirangat et al. (2020), che permette di vedere su grafico come sono costruiti i cluster, sovrapponendo il clustering di C2 a quello di C1. La procedura disegna tutti i centroidi dei cluster, sia quelli in C1 che in C2 (indicati dal marcatore di forma romboidale). Per i

cluster uniti dall'algoritmo e che vengono quindi condivisi da punti di C1 e C2, la procedura disegna anche una linea di collegamento tra centroidi. Per cluster isolati i centroidi rimangono senza un collegamento e questo può accadere per punti che provengono solo da C1 o solo da C2. Questa tecnica articolata ma completamente automatizzata è stata resa modulabile in fasi progressive grazie alla possibilità di salvare tutti i dati di elaborazione dei grafi (etichette, centroidi, punti) in formato compresso pkl, da riutilizzare all'occorrenza.

Come mostrato dal grafico della parola *rampante* (figura 14) le procedure di clustering riescono ad individuare quattro cluster condivisi tra C1 e C2, dei quali vengono disegnate le distanze tra i centroidi. Si nota anche un cluster presente solo in C1 (con marcatori dal colore rosa), evidenziato nell'immagine da una linea tratteggiata in rosso, che non trova un match positivo in C2, per il quale viene prodotto un unico centroide.

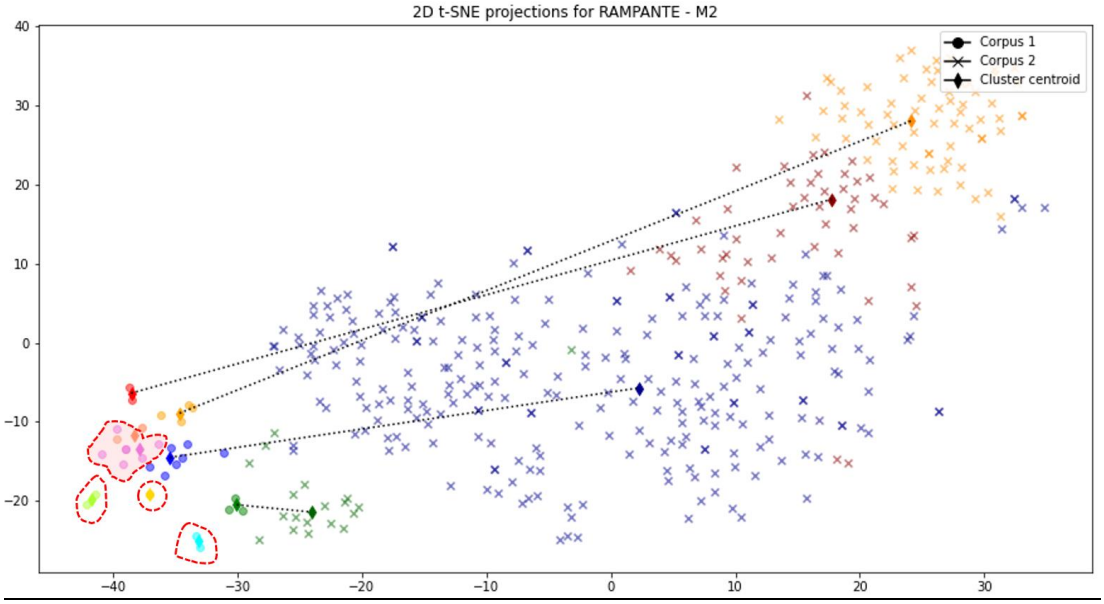

Figura 14: plot grafico della proiezione dei cluster di C2 su quelli di C1 della parola 'rampante' (metodo 2.2)

Sono evidenti nell'immagine anche piccoli cluster 'puri' di sole occorrenze di C1, una sorta di outliers, anch'essi evidenziati nell'immagine da una linea rossa

tratteggiata. Quindi il clustering ha individuato una positività al cambio di senso anche con piccoli numeri di frequenza.

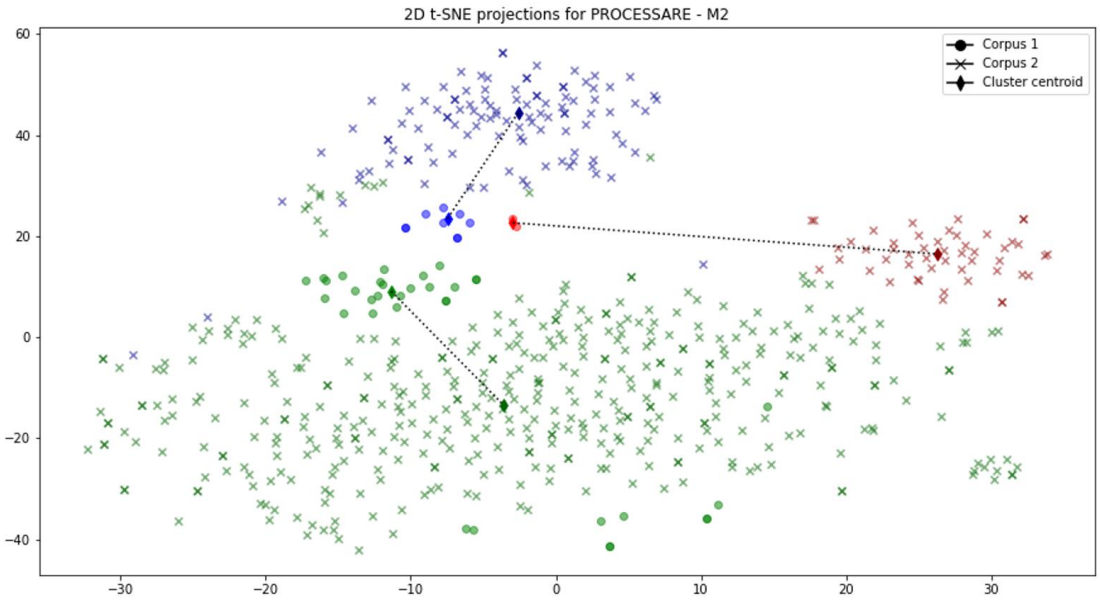

Figura 15: plot grafico della proiezione dei cluster di C2 su quelli di C1 della parola 'processare' (metodo 2.2)

La stessa corretta individuazione, questa volta riguardante però l'assenza di cambiamento semantico, si verifica per la parola *processare* mostrata in fig. 15. In questo caso le procedure di clustering (*metodo 2.2*) producono per C1 e C2 lo stesso numero di cluster (3) collegati dai rispettivi centroidi, nonostante un numero molto sbilanciato di frequenze tra i corpora.

Un'analisi dei cluster prodotti evidenzia embeddings di C1 e C2 distribuiti nei tre cluster, che evidenziano differenze tra i loro embeddings di tipo minimo. Per esempio il cluster 1: contiene embeddings relativi a frasi come: "*Arrestare e processare i responsabili*"; "*Leggi speciali per* ***processare*** *le 60 personalità…*". Mentre per C2 lo stesso cluster presenta embeddings relativi a frasi come: "***processare*** *criminali di guerra .*"; "***processare*** *i piloti , vista anche la non corrispondenza in il diritto penale*"; "***processare*** *per gestione illecita di i rifiuti , denuncia : « Abbiamo dovuto contrastare attività , ma anche sabotaggi e boicottaggi*". Il cluster numero 2 contiene embeddings di frasi di C1 del tipo: "*Si vuole* ***processare*** *l' Ideologia marxista ?*"; "*Tribunali speciali , costituiti per* ***processare*** *gli scioperanti funzionano in permanenza*

a Barcellona" e frasi di C2 del tipo: "*40 di i 55 gerarchi ricercati , ma il proposito annunciato ieri a Baghdad di* ***processare*** *chi ha commesso reati,…*"; "*A Belgrado che vuole* ***processare*** *i prigionieri…*"; "*Il Tribunale Supremo chiede di* ***processare*** *il capo di gli Esteri per una megafrode fiscale*". L'ultimo cluster sempre contenente embeddings condivisi tra C1 e C2 sembra raggruppare frasi in cui è indicato un soggetto/nome vicino alla parola target, alcuni esempi sono in C1: "*Burghiba fa* ***processare*** *comunisti e studenti*"; "*In il 1941 , infatti , Petain lo fece internare e poi* ***processare*** .."; "*L' Unita che si faceva* ***processare*** *per le « cartoline rosa »*". Mentre in C2 ricorre tra l'altro quasi tutta la storia giudiziaria di Berlusconi raccontata dall'Unità: "*anche contro soggetti che la Procura di l' Aja vorrebbe* ***processare*** *per crimini contro l' umanità*"; "*Berlusconi e Previti non vogliono far si* ***processare*** *in assoluto e mettono le mani avanti*"; "*Il premier trovi il tempo per far si* ***processare*** *così come lo ha trovato per presenziare a il compleanno di Noemi*"; "*Napolitano spiega che Berlusconi deve far si* ***processare*** *perché in la Costituzione ci sono le garanzie*".

L'analisi degli embeddings conferma una distribuzione dell'uso della parola tra C1 e C2 e questo potrebbe far pensare a un comportamento corretto per ogni parola con occorrenze sbilanciate tra C1 e C2, ma non è così. Infatti nella figura che segue (fig. 16) è mostrato come per la parola *piovra* il metodo 2.2 non sia riuscito a ricondurre da solo agli stessi 'usi' le occorrenze in C1 e C2. Dal grafico risulta evidente che le occorrenze di C1 (identificabili dal marcatore a forma di punto) non siano in numero sufficiente a confutare o meno l'analisi di un cambio di significato/uso tra i due archi temporali. È infatti presente nel grafico un grande cluster isolato di embeddings di C2, identificabile da marcatori di colore giallo, il cui centroide non trova un collegamento in nessun punto appartenente a C1. Questo comportamento di norma indicherebbe un nuovo significato/uso in C2 non trovato in C1. La presenza però di dimensioni disomogenee tra gli embeddings di C1 e quelli di C2 ha richiesto una conferma del risultato con l'applicazione anche del metodo 2.1.

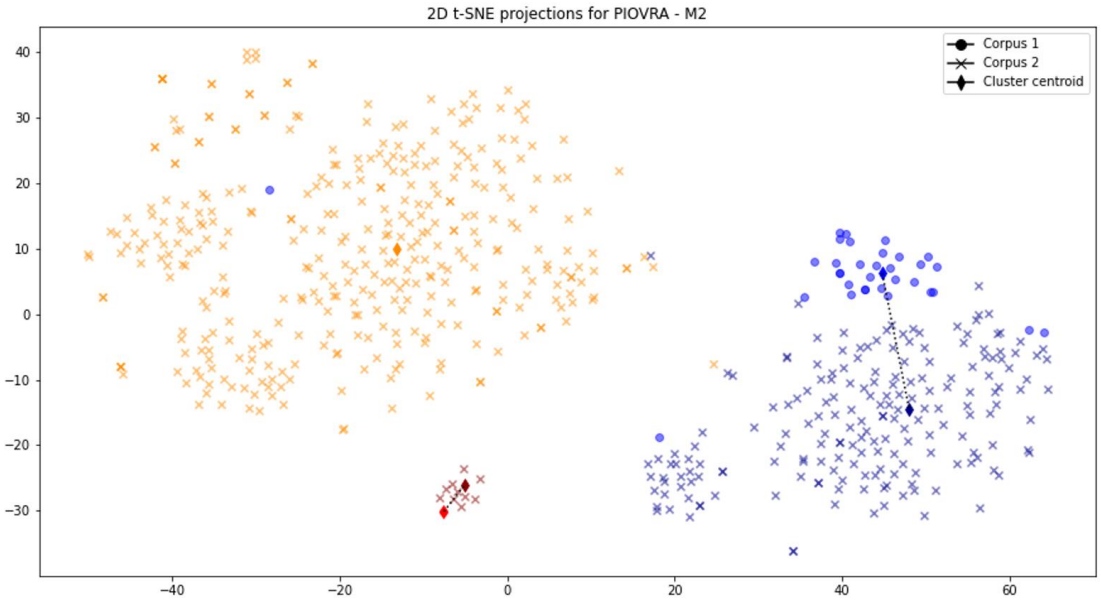

Figura 16: plot grafico della proiezione dei cluster di C2 su quelli di C1 della parola 'piovra' (metodo 2.2)

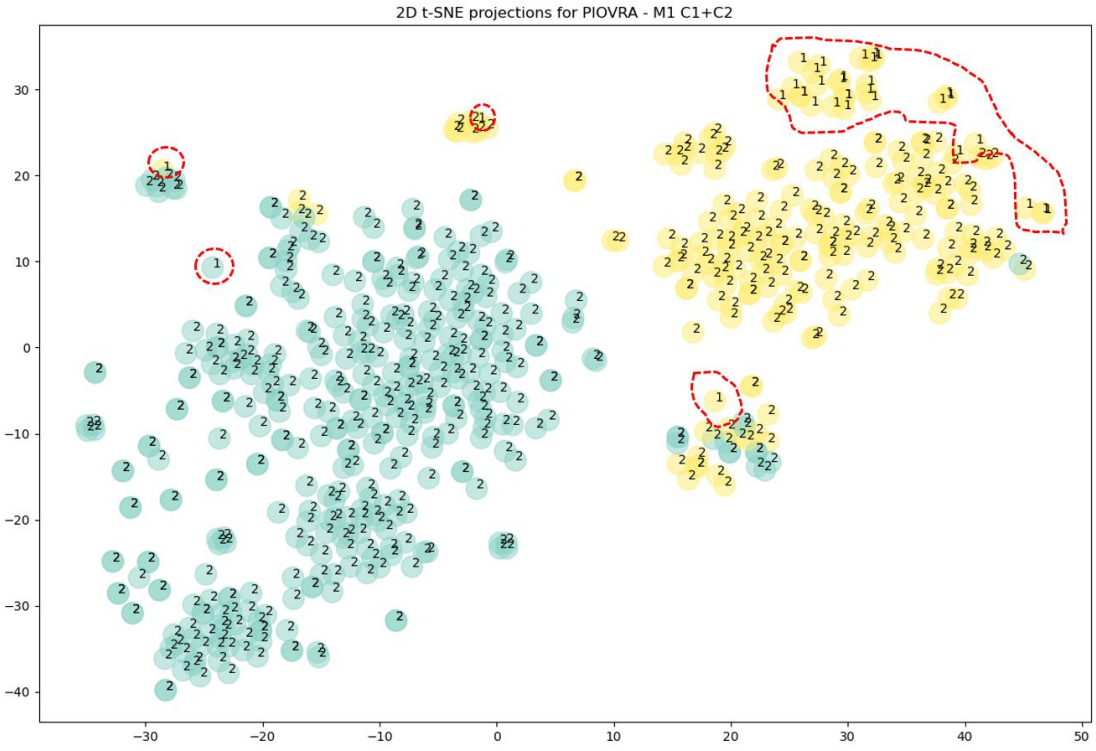

Figura 17: plot grafico della proiezione del clustering in C1+C2 della parola 'piovra' (metodo 2.1)

Per la stessa parola questo metodo, applicato sul corpus ottenuto dall'unione di C1+C2, riesce invece a rispondere negativamente al cambio di significato/uso e non a caso ha ottenuto i migliori risultati, relativamente al clustering, su tutti i dati del task. Probabilmente lavorando su dati di dimensioni maggiori in input (embeddings di C1 + C2), ha creato più verosimilmente solo 2 cluster e i vettori di embeddings della parola estratti dai due corpora si distribuiscono in entrambi.

Nella figura 17 i punti corrispondenti ai vettori delle occorrenze di C1, indicati dall'etichetta '1', sono stati evidenziati nella figura da una linea tratteggiata in rosso. La non positività al cambio di senso si deve comunque all'unico embeddings di C1 che appartiene a un cluster con solo punti appartenenti a C2, era lecito quindi domandarsi se questa condizione fosse sufficiente alla valutazione di 'uniforme distribuzione' nei due cluster.

In SemEval era stato previsto il trattamento di casi come questo, infatti si indicava chiaramente quale fosse la soglia ammessa per i punti appartenenti ai cluster: '*perché una parola abbia acquisito un senso, il senso deve apparire un massimo di k volte in C1 e almeno n volte in C2 e viceversa'* (§ 4.1). Erano state previste soglie di k e n diverse per lingue. Per il latino k = 0 e n = 1, per le altre lingue k = 2 e n = 5, proprio in ragione delle dimensioni ridotte dei dati in lingua latina. Considerazioni sulle dimensioni dei corpora che si potevano applicare anche a quelli di DIACR-Ita. Con queste premesse è risultata quindi corretta la decisione di non registrare un nuovo senso per la parola.

Volendo però analizzare la cosa in dettaglio, per capire se la condizione rispondesse effettivamente ad un uso comune in C1 e C2 si sono esaminate le frasi corrispondenti agli embeddings estratti. Come fatto per la parola *processare* anche per *piovra* si è voluto scendere nel merito del processo di aggregazione fatto dall'algoritmo di clustering e comprendere la genesi dei due cluster, chiamati A e B per comodità. L'obiettivo era verificare che l'unico punto di C1 in A fosse la registrazione corretta di uso comune ai due corpora. Si tratta infatti dell'uso della parola piovra come titolo della produzione televisiva 'La piovra'. È evidente che nel corpus C1, risalente al periodo (1948-

1970) non vi fosse traccia della fiction, ma è stata comunque correttamente riconosciuta dagli algoritmi di clustering una forma di contiguità nell'uso di quella parola nella frase: '*DI GLI SCIPIONI* ***Piovra*** *nera con D. Andrews*'[61] presente in C1, rispetto alle tante trovate in C2 : '*Vittorio Mezzogiorno in un momento di « La* ***Piovra*** *5 »...*'; '*Anche noi abbiamo lasciato La* ***piovra****, dopo quattro serie eravamo abbastanza stanchi*'; '*Bambino che in la* ***Piovra*** *9 non troviamo affatto perché sta studiando a l' estero .*'; '*Come è successo per i film su la mafia che hanno avuto inizi importanti, rappresentati con i film di Rosi, e che si sono trasformati in la saga di la* ***Piovra***.'

L'altro uso, che si attesta sia in C1 che in C2 è quello più noto utilizzato per indicare attività mafiose. Per questo uso, anche con dimensioni inferiori del corpus più distante cronologicamente, il significato è uniformemente distribuito e riconosciuto. Infatti in C1 troviamo frasi come: '*I tentacoli di la* ***piovra*** *seconds i dati ufftciali'; 'La DC e come una* ***piovra****...*'; '*... i responsabili di la sopravvivenza di le strutture feudali e di il latifondo in tanta parte di il Paese, e i beneficiari di i superprofitti di i monopoli (che sono l' altra* ***piovra*** *strutturale di l' economia italiana) ...*'; *'...i tentacoli di la* ***piovra*** *, tutt' uno ormai in il Mezzogiorno con quelli di la Democrazia cristiana'.*

Mentre in C2 troviamo: '*C era una volta il pool antimafia Il suo lavoro si basava su il tentativo di opporre a le attivtà di la* ***piovra*** *senza volto un gruppo di giudici ...*'; '*C' è una « ****piovra*** *verde » che cresce in Italia con un intreccio che vede lavorare fianco a fianco mafia...*'; '*D12 settembre Palermo marcerà contro la « ****piovra*** *» far si interpreti di una forte reazione unitaria...*'. Si può capire anche qualcosa di più da questi dati, ossia l'orientamento del giornale, ovvero L'Unità da cui sono estratti i corpora.

[61]La Piovra Nera è un film di genere poliziesco del 1958, diretto da Jacques Tourneur, con Dana Andrews e Dick Foran.

8.3 Il trattamento di varietà storiche della lingua

L'osservazione che chiude il paragrafo precedente porta a fare una valutazione più generale, che riguarda lo studio del fenomeno su corpora testuali. I linguisti hanno da tempo riconosciuto come "linguaggio" sia un termine ampio, che nella migliore delle ipotesi designa una comoda astrazione di una realtà complessa. Tuttavia, ciò non significa che qualsiasi campione linguistico debba essere considerato ugualmente rappresentativo. Soprattutto la linguistica dei corpora si è interrogata e ha profuso energie intellettuali nell'indicare come compilare campioni linguistici rappresentativi. Contesto dove è chiaro che "rappresentativo" generalmente deve essere interpretato in relazione a una specifica domanda di ricerca.

L'ambito di studio affrontato in questo lavoro di tesi indaga la lingua scritta, quindi i dati da sottoporre alle analisi sono sostanzialmente i corpora testuali, la natura e composizione dei quali ha un ruolo determinante sul tipo di studio che si può fare su di essi. Interrogarsi sulla possibilità di identificare un cambio di significato/uso di parole, sia tra periodi relativamente brevi o per archi temporali più ampi è uno studio che abbiamo visto moltiplicarsi nella letteratura analizzata anche con buoni risultati. Quando però si vuole spingere le analisi verso studi più fini, come l'individuazione del punto di cambio o la più generale 'storia delle parole', i dati di partenza diventano una questione determinante.

A maggior ragione se si vuole trattare varietà storiche della lingua come si è provato a fare in questo lavoro di tesi. Per la maggior parte dei periodi storici e per la gran parte delle lingue, sono disponibili solo testi prodotti da campioni piccoli e distorti. Tahmasebi et al. (2021) in '*Computational approaches to semantic change*' si domandano: le soglie di occorrenza nei testi storici rifletteranno fedelmente i punti di cambiamento sottostanti? La domanda posta è rilevante anche per quello che riguarda i modelli perfezionati su tali dati. Fino a che punto il fine-tuning dei modelli pre-addestrati è in grado di 'apprendere' varietà storiche della lingua? Quanto indietro si può spingere

lo studio in diacronia? Già nell'arco temporale analizzato si è visto che nell'elaborazione dei dati sono necessariamente rimaste fuori tutte le forme arcaiche delle parole che, se ricondotte alla forma normalizzata, avrebbero potuto concorrere allo studio del fenomeno. Se ne riportano di seguito solo pochi esempi: *ajuto*, per *aiuto*, *avversarjo* per *avversario*, *aggradevole* per *gradevole*, *dipositare* per *depositare*, *ismania* per *smania*, *istoria* per *storia*, *forastiero* per *forestiero*, ecc. Per non parlare dell'uso diffuso delle parole tronche che contribuisce anch'esso al proliferare delle varianti ortografiche della stessa parola. Questo per sottolineare quanto la direzione dello studio che guarda indietro, a periodi lontani nel tempo, sia piena di problemi da risolvere.

Sempre Tahamasebi afferma che il trattamento del dato in varietà storiche è una lacuna comune, situazione resa più complessa poiché solo pochi approcci propongono tecniche in grado di analizzare il cambiamento semantico in parole con poche occorrenze. Gli autori ritengono che la quantità di dati per le parole a bassa frequenza può essere insufficiente per costruire ipotesi affidabili utilizzando metodi standard. Essi sostengono però che le immersioni dinamiche sembrano offrire un'alternativa più adatta rispetto a piccoli set di dati, come dimostra questa tesi.

8.4 La valutazione del metodo

Al termine di tutte le sperimentazioni il metodo proposto è risultato idoneo allo studio del fenomeno. Anche le esigenze più particolari, legate alle caratteristiche dei dati possono essere trattate con l'impostazione di soglie di applicazione per i vari metodi (§§ 7.3 e 7.4). Si può inoltre affermare che la strategia proposta si presta alla generalizzazione, alla standardizzazione del metodo, cosa che si è testata sul campo. È ragionevole pensare che con corpora di grandi dimensioni il solo clustering possa dare ottimi risultati, e lasci indietro tutte le considerazioni sul rapporto di forze tra dimensioni delle occorrenze delle parole nei due corpora analizzati. Si possono quindi individuare scenari diversi ma tutti possibili, per l'uso del metodo:

1) C1 e C2 di grandi dimensioni in cui le disomogeneità di attestazione delle parole sono trascurabili. In questi casi, fissando la soglia delle occorrenze a 700 (massimo 1000), si lascia al clustering l'individuazione del fenomeno. In questo scenario l'uso indifferente di K-mean e Dbscan è possibile proprio perché, con tali dimensioni degli embeddings, le funzioni per il calcolo delle configurazioni ottime, sia per l'uno che per l'altro, producono risultati poco ambigui. Anche la produzione di curve grafiche in cui individuare discontinuità e massima curvatura è del tutto evidente e catturabile automaticamente.
2) C1 e C2 di dimensioni diverse, con rapporti tra dimensioni di occorrenze molto sbilanciati. In questo altro scenario è necessario imporre soglie all'uso di un metodo invece di un altro. Similmente a quanto accaduto per DIACR-Ita e ILC-Ita, si indaga il fenomeno su più fronti e si richiede una conferma incrociata per l'attribuzione di un nuovo significato nei casi definiti 'disomogenei'.
3) C1 e C2 hanno dimensioni simili ma contenute. In questo contesto l'uso del clustering può essere fuorviante se utilizzato come unico strumento d'indagine. La distanza tra vettori di embedding è in grado di trattate il fenomeno con buoni risultati, ma è possibile riservare il supporto più fine del clustering, nei casi in cui vi siano comunque sbilanciamenti tra occorrenze di parole tra C1 e C2.

Capitolo 9

Conclusioni

In questa tesi è stato presentato un metodo per il riconoscimento del cambiamento semantico nell'evoluzione temporale del linguaggio, basato sull'utilizzo di modelli transformer tipo BERT e di algoritmi di clustering. Il metodo utilizza tecniche e algoritmi a complessità relativamente bassa, in grado di supportare un cambio di scala per uno studio strutturale del fenomeno. Come indicato nel cap. 2 l'efficacia del metodo è stata verificata sia su corpora usati in letteratura, sia su corpora italiani appositamente costruiti. La sperimentazione relativamente al metodo d'indagine ha consentito di ottenere risultati con accuratezza pari allo stato dell'arte, smentendo in tal modo un precedente risultato in letteratura, secondo cui i modelli BERT non sarebbero adatti allo scopo. Oltre a questo si è cercato di indagare il legame tra embeddings contestualizzati, dati e strumento d'indagine, con l'obiettivo di offrire un contributo teorico allo studio del fenomeno e spunti interpretativi che permettano di pensare a prospettive future.

Il metodo si articola in alcune varianti, che hanno vantaggi e svantaggi, in termini di ambito di applicazione, tempo di elaborazione e interpretabilità. I contributi sperimentali rilevanti riguardano l'utilizzo efficace dei modelli BERT; la scelta del fine-tuning; la modalità di estrazione degli embeddings contestualizzati e l'utilizzo del clustering per individuare i cambiamenti. Il risultato finale è il concorso di tutte le scelte fatte e delle sperimentazioni puntuali che sono state condotte, sempre volte a confrontare il comportamento dei modelli e valutare la loro capacità nel rilevare il cambiamento semantico.

Pensando di offrire spunti utili alla creazione di metodi basati sui modelli BERT si è provato anche ad analizzare l'esperienza complessiva, che riguarda tutti coloro che intendono misurarsi da neofiti con questi modelli. Lo studio e la pratica si sono rivelati fondamentali a produrre buoni risultati, non solo perché si è imparato a lavorare con vari strumenti software, necessari al lavoro tali modelli, ma soprattutto perché si è provato a sperimentare in proprio strategie e approcci. Le esperienze di altri possono servire da spunto, possono essere considerati utili termini di paragone sui quali misurare i propri sforzi, ma non possono sostituire l'esperienza sul campo. Lo sviluppo di procedure flessibili, adattabili, richiede necessariamente la capacità di valutare quanto di utilizzabile vi sia in un certo approccio, sfrondandolo di tutti i 'constraints' dovuti all'adattamento del caso d'uso specifico per cui è stato prodotto. Non è un lavoro facile, tanto più in un ambito di ricerca in continuo divenire, ma è molto importante. Infatti lo studio in diacronia del cambiamento semantico utilizzando modelli distributivi (incluse le immersioni di parole) rimane tutt'altro che un problema risolto: il campo presenta ancora un numero considerevole di sfide aperte.

Bibliografia

Arase, Yuki and Jun'ichi Tsujii. Transfer fine-tuning: A BERT case study. 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)*. pp. 5393-5404. Hong Kong. China. Association for Computational Linguistics. 2019.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 2, Short Papers. pp. 427-431, Valencia, Spain. Association for Computational Linguistics. 2017.

Azarbonyad, Hosein, Mostafa Dehghani, Kaspar Beelen, Alexandra Arkut, Maarten Marx & Jaap Kamps. Words are malleable: Computing semantic shifts in political and media discourse. *In Proceedings of CIKM 2017*. pp. 1509-1518. Singapore. ACM. DOI: 10.1145/3132847.3132878. 2017.

Baroni Marco, Georgiana Dinu, Germán Kruszewski. *"*Don't count, predict! a systematic comparison of contextcounting vs. context-predicting semantic vectors*"*. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. vol. 1. pp. 238-247. 2014.

Basile, Valerio, Andrea Bolioli, Malvina Nissim, Viviana Patti, Paolo Rosso. Overview of the *Evalita 2014: SENTIment POLarity Classification Task*. 10.12871/clicit201429. 2014.

Basile, Pierpaolo, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti e Rossella Varvara. Overview of the EVALITA 2020 diachronic lexical semantics (DIACR-Ita) task. In Valerio Basile, Danilo Croce, Maria Di Maro e Lucia C. Passaro (eds.), *Proceedings of the 7th evaluation campaign of natural language processing and speech tools for Italian (EVALITA 2020)*. 2020.

Beck, Christin. DiaSense at SemEval-2020 Task 1: Modeling Sense Change via Pre-trained BERT Embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 50-58. Barcelona (online). International Committee for Computational Linguistics. 2020.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer. 2006.

Bloomfield, Leonard. *Language*. New York: Henry Holt. 1933.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. 5:135-146. 2017.

Bréal, Michel. *Essai de Sémantique, science des significations*. Paris. Hachette. 1921.

Bullinaria, John A., & Joe P. Levy. Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study. *Behavior Research Methods*, 39. pp. 510-526. 2007.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR 2020: Computation and Language*. DOI:10.48550/ARXIV.2003.10555. 2020.

Collobert, Ronan, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*. 12:2493–2537. 2011.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by Latent Semantic Analysis. In *Computer Scienc*. J. Am. Soc. Inf. Sci. DOI:10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9. 1990.

Del Tredici, Marco, Raquel Fernandez, and Gemma Boleda. Short-Term Meaning Shift: A Distributional Exploration. In *Proceedings of NAACL-HLT 2019* (Annual Conference of the North American Chapter of the Association for Computational Linguistics). 2019.

De Stefanis Ciccone, Stefania, Ilaria Bonomi, Andrea Masini. La stampa periodica milanese della prima metà dell'Ottocento: testi e concordanze. A cura di Remo Bindi e Eugenio Picchi. *Orientamenti linguistici*. Giardini editore. ISBN: 8842712167. 1984.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *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. Minneapolis. Minnesota. ACL. 2019.

Dodge, Jesse, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. *Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping*. arXiv preprint. arXiv:2002.06305. 2020.

Eisenstein, Jacob. Measuring and modeling language change. In *Proceedings of NAACL 2019*. Tutorials. pp. 9-14. ACL. 2019.

Ester, Martin, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings KDD-96*. AAAI (www.aaai.org). 1996.

Firth, John Rupert. A Synopsis of Linguistic Theory, 1930-1955 *Studies in Linguistic Analysis*. Oxford: Blackwell. pp. 1-32. 1957.

Freemann, Lea & Mirella Lapata. A Bayesian model of diachronic meaning change. *Transactions of the ACL* 4. 31–45. DOI: 10.1162/tacl_a_00081. 2016.

Giulianelli, Mario, Raquel Fernández, Marco Del Tredici. Contextualised word representations for lexical semantic change analysis. In *EurNLP*. London. UK. 2019.

Giulianelli, Mario, Marco Del Tredici & Raquel Fernández. Analysing lexical semantic change with contextualised word representations. In *Proceedings of ACL 2020*. pp. 3960-3973. Online: ACL. DOI: 10.18653/v1/2020.acl-main.365. 2020.

Gonen, Hila, Ganesh Jawahar, Djamé Seddah & Yoav Goldberg. Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*. pp. 538-555. 2020.

Gries, Stefan. Th. Particle movement: a cognitive and functional approach. *Cognitive linguistics*, 10(2), pp. 105-146. 1999.

Gulordava, Kristina, & Marco Baroni. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. pp. 67-71. Edinburgh, UK. 2011.

Hamilton, William L., Jure Leskovec & Dan Jurafsky. Cultural shift or linguistic drift? Comparing two computational measures of semantic change. In *Proceedings of EMNLP 2016*. pp. 2116-2121. Austin: ACL. DOI:10 .18653/v1/D16 - 1229. 2016a.

Hamilton, William L., Jure Leskovec & Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of ACL 2016* (Volume 1: Long papers). pp. 1489-1501. Berlin: ACL. DOI: 10.18653/v1/P16-1141. 2016b.

Harris, Zellig S. *Distributional structure*. Word 10(2-3). pp. 146-162. 1954.

Hedderich, Michael A., Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 2545-2568, Online. ACL. 2021.

Hilpert, Martin. Distinctive collexeme analysis and diachrony. *Corpus Linguistics and Linguistic Theory*, 2(2), pp. 243-256. DOI 10.1515/CLLT.2006.012. 2006.

Hilpert, Martin. New evidence against the modularity of grammar: Constructions, collocations, and speech perception. *Cognitive Linguistics*, 19(3). pp. 391-411. 2008.

Hilpert, Martin, Stefan Th. Gries. Assessing frequency changes in multistage diachronic corpora: Applications for historical corpus linguistics and the study of language acquisition. *Literary and Linguistic Computing* 24(4): pp. 385-401. DOI:10.1093/llc/fqn012. 2009.

Hilpert, Martin & Perek, Florent. Meaning change in a petri dish: Constructions, semantic vector spaces, and motion charts. *Linguistics Vanguard*. 1. 10.1515/lingvan-2015-0013. 2015.

Jawahar, Ganesh, Benôıt Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pp. 3651-3657. Florence. Italy. Association for Computational Linguistics. 2019.

Jurgens, David & Stevens, Keith. *Event Detection in Blogs using Temporal Random Indexing*. pp. 9-16. 2009.

Kanerva, Pentti, Jan Kristofersson, and Anders Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*. Erlbaum. 1036. 2000.

Kanjirangat, Vani, Sandra Mitrovic, Alessandro Antonucci, and Fabio Rinaldi. 2020. SST-BERT at SemEval-2020 Task 1: Semantic Shift Tracing by Clustering in BERT-based Embedding Spaces. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 214-221. Barcelona (online). International Committee for Computational Linguistics. 2020.

Karnysheva, Anna, Pia Schwarz. TUE at SemEval-2020 Task 1: Detecting semantic change by clustering contextual word embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 232-238. Barcelona (online). International Committee for Computational Linguistics. 2020.

Kim, Y., Chiu, Y.-I., Hanaki, K., Hegde, D., & Petrov, S. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. pp. 61-65. Baltimore. MD. USA. 2014.

Kuhn, Harold W. The hungarian method for the assignment problem. *Naval research logistics quarterly*. 2(1-2):83-97. 1955.

Kulkarni, Vivek, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically Significant Detection of Linguistic Change. In *Proceedings of the 24th International Conference on World Wide Web*. pp 625-635. International World Wide Web Conferences Steering Committee. 2015.

Kutuzov, Andrey, Lilja Øvrelid, Terrence Szymanski & Erik Velldal. Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of COLING 2018*. pp. 1384-1397. Santa Fe: ACL. 2018.

Kutuzov, Andrey and Mario Giulianelli. UiO-UvA at SemEval-2020 Task 1: Contextualised Embeddings for Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 126-134. Barcelona (online). International Committee for Computational Linguistics. 2020.

Laicher, Severin, Gioia Baldissin, Enrique Castañeda Dominik Schlechtweg, Sabine Schulte im Walde. CL-IMS @ DIACR-Ita: Volente o Nolente: BERT does not Outperform SGNS on Semantic Change Detection. *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*. 2020.

Lance, Godfrey N. and William T. Williams. Computer programs for hierarchical polythetic classification ("similarity analyses"). *Comput. J. 9. 1* (1966). pp. 60-64. 1966.

Lazzeroni, Romano. *La ricostruzione culturale fra comparazione lessicale e ricostruzione etimologica*. S.I. pp. 305-315. 1990.

Lazzeroni, Romano. Fra mutamento linguistico e organizzazione della memoria: la partizione del paradigma in alcune lingue indoeuropee. in M.G. Busà - S. Gesuato (eds.), *Lingue e contesti. Studi in onore di Alberto M. Mioni*. Padova. CLEUP: pp. 125-142. 2015.

Lee, Cheolhyoung, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *8th International Conference on Learning Representations. (ICLR)*. 2020.

Levy, Omer & Goldberg, Yoav. Dependency-Based Word Embeddings. 52nd Annual Meeting of the Association for Computational Linguistics. *ACL 2014 - Proceedings of the Conference*. 2. pp. 302-308. DOI:10.3115/v1/P14-2050. 2014.

Lijffijt, Jefrey, Stefan Th. Gries. Correction to Stefan Th. Gries "Dispersions and adjusted frequencies in corpora". *International Journal of Corpus Linguistics. International Journal of Corpus Linguistics* 17(1). DOI:10.1075/ijcl.17.1.08lij. 2012.

Lin, J. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*. 37 (1). pp. 145-151. 1991.

Lloyd, Stuart P. Least squares quantization. in pcm. *IEEE Transactions on Information Theory*. 28(2). pp. 129-137. 1982.

Marinelli, Rita, Lisa Biagini, Remo Bindi, Sara Goggi, Monica Monachini, Paola Orsolini, Eugenio Picchi, Sergio Rossi, Nicoletta Calzolari, Antonio Zampolli. The Italian PAROLE corpus: an overview. In: Zampolli, N. Calzolari, L. Cignoni, (eds.), *Computational Linguistics, in Pisa - Linguistica Computazionale a Pisa*. Linguistica Computazionale, Special Issue, XVI-XVII. Pisa-Roma. IEPI. Tomo I. pp. 401-421. 2003.

Martinc, Matej, Syrielle Montariol, E. Zosa and Lidia Pivovarova. Capturing evolution in word usage: just add more clusters? In *Companion Proceedings of the Web Conference 2020*. pp. 343-349. WWW '20. Taipei. Taiwan. Association for Computing Machinery. 2020a.

Martinc, Matej, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings Not Always Better than Static for Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 67-73. Barcelona (online). International Committee for Computational Linguistics. 2020b.

Martinc, Matej, Kralj Novak, Petra and Pollak, Senja. Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift. In *Proceedings of the 12th*

Language Resources and Evaluation Conference. European Language Resources Association. https://aclanthology.org/2020.lrec-1.592. pp. 4811-4819. 2020c.

Meillet, Antoine. Comment les mots changent de sens. *Année Sociologique*. 1905.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray. The Google Books Team, Joseph P. Pickett, et al. «Quantitative Analysis of Culture Using Millions of Digitized Books». *Science 331*, n. 6014 (14 gennaio 2011): 176-82. 2011.

Mikolov, Tomas, Kai Chen, Greg Corrado and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR2013*. arXiv:1301.3781v1. 2013.

Mitra, S., R. Mitra, M. Riedl, C. Biemann, A. Mukherjee, & P. Goyal. That's sick dude!: Automatic identification of word sense change across different timescales. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers). pp. 1020–1029. Baltimore. Maryland. 2014.

Montariol, Syrielle, Matej Martinc, Lidia Pivovarova. Scalable and Interpretable Semantic Change Detection. In Proceedings of NAACL. *Computer Science*. DOI:10.18653/V1/2021.NAACL-MAIN.369. 2021.

Mosbach, Marius, Andriushchenko, Maksym , Klakow, Dietrich. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. *ICLR 2021 Conference*: ICLR 2021 Poster. https://openreview.net/group?id=ICLR.cc/2021/Conference#poster-presentations. 2021.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1532-1543. Doha. Qatar. Association for Computational Linguistics. 2014.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Volume 1 (Long Papers). pp. 2227-2237. New Orleans. Louisiana. Association for Computational Linguistics. 2018.

Picchi, Eugenio, Eva Sassolini. "Text power": Tools for the cultural heritage. In: *CHC 2010 - 4-th Int. Congr. Science and Technology for the Safeguard of Cultural Heritage in the Mediterranean Basin* (2009). Proceedings vol. 1. pp. 435-439. Fondazione Roma Mediterraneo. 2010.

Polignano, Marco, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, Valerio Basile. ALBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. *IJCoL - Italian Journal of Computational Linguistics*. Accademia University Press. pp. 11-31. 2019.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language models are unsupervised multitask learners. *In Computer Science*. 2019.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., & Kim, B. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems* 32. pp. 8594-8603. 2019.

Rockafellar, R. Tyrrell and Roger J-B. Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media. 2009.

Rodina, Julia, Trofimova, Y., Kutuzov, A., & Artemova, E. ELMo and BERT in semantic change detection for Russian. *Conference - AIST 2020*. 2020

Rohrdantz, Christian & Hautli, Annette & Mayer, Thomas & Butt, Miriam & Keim, Daniel & Plank, Frans. *Towards Tracking Semantic Change by Visual Analytics*. pp. 305-310. 2011.

Rother, David, Thomas N. Haider, Steffen Eger. CMCE at SemEval-2020 Task 1: Clustering on Manifolds of Contextualized Embeddings to Detect Historical Meaning Shifts. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 187-193. Barcelona (online). International Committee for Computational Linguistics. 2020.

Rousseeuw, Peter. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*. 20. pp. 53-65. DOI:10.1016/0377-0427(87)90125-7. 1987.

Jonker, Roy and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*. 38(4):325-340. 1987.

Sagi, Eval, Stefan Kaufmann & B. Clark. Semantic density analysis: comparing word meaning across time and phonetic space. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. pp. 104-111. Athens. Greece. 2009.

Sagi, Eyal, Stefan Kaufmann & Brady Clark. Tracing semantic change with Latent Semantic Analysis. In Kathryn Allan & Justyna A. Robinson (eds.). *Current methods in historical semantics*. pp. 162-183. Berlin: De Gruyter Mouton. 2011.

Salton, Gerard and Michael J McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company. New York. 1983.

Sassolini, Eva, Eugenio Picchi. Ipotesi di sviluppo per i sistemi informativi della direzione nazionale antimafia. In *Workshop: Ipotesi di sviluppo per i Sistemi Informativi della Direzione Nazionale Antimafia* (Bologna, 28 settembre 2010). 2010.

Schlechtweg, Dominik, Schulte imWalde, S., & Eckmann, S. Diachronic usage relatedness (DURel): a framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies. Volume 2 (Short Papers) pp. 169-174. 2018.

Schlechtweg, Dominik, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky & Nina Tahmasebi. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of SemEval 2020*. pp. 1-23. Barcelona: ACL. https://www.aclweb.org/anthology/2020.semeval-1.1. 2020.

Schubert, Erich, Sander, Jörg, Ester, Martin, Kriegel, Hans Peter, Xu, Xiaowei. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst*. 42 (3): 19:1–19:21. DOI:10.1145/3068335. ISSN 0362-5915. S2CID 5156876. 2017.

Shoemark, Philippa, Farhana Ferdousi, Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. pp. 66-76. Hong Kong. China. Association for Computational Linguistics. 2019.

Solomon, J. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*. 2018.

Stern, Gustaf. *Meaning and change of meaning; with special reference to the English language*. Gothenburg: Wettergren & Kerbers. 1931.

Tahmasebi, Nina, Lars Borin, and Adam Jatowt. Survey of Computational Approaches to Lexical Semantic Change. *Computational Linguistics*, 1(1). DOI:10.5281/zenodo.5040241. ISBN-13 (15) 978-3-98554-008-2. pp. 1-91. 2018

Tahmasebi, Nina, Lars Borin, Adam Jatowt, and Yang Xu. *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics. Florence. Italy. Edition. 2019.

Tahmasebi, Nina, Borin, Lars, Jatowt, Adam, Xu, Yang & Hengchen, Simon (eds.). *Computational approaches to semantic change*. (Language Variation 6). Berlin: Language Science Press. DOI: 10.5281/zenodo.5040241. 2021.

Tang, Xuri. A state-of-the-art of semantic change computation. *Natural Language Engineering*. 24(5). pp. 649-676. https://doi.org/10.1017/s1351324918000220. 2018.

Tonelli, Sara, Rachele Sprugnoli, Giovanni Moretti, and Fondazione Bruno Kessler. Prendo la parola in questo consesso mondiale: A multi-genre 20th century corpus in the political domain. In *Proceedings CLiC-it*. 2019.

Traugott, Elizabeth Closs. Semantic change. In *Oxford research encyclopedia of linguistics*. Oxford: Oxford University Press. 2017.

Turney, Peter & Pantel, Patrick. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*. 37. DOI:10.1613/jair.2934. 2010.

Ullmann, Stephen. *The Principles of Semantics*. Blackwell. Glasgow: Jackson. 1951.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. Attention Is All You Need. *Conference - NeurIPS 2017*. 2017.

Wang Benyou, Emanuele Di Buccio and Massimo Melucci. University of Padova @ DIACR-Ita. *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*. 2020.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, (et ali.) and Dean, Jeffrey. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016.

Xiao, Han. bert-as-service. https://github.com/hanxiao/bert-as-service. 2018