UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI ECONOMIA "GIORGIO FUÀ"

Corso di Laurea Magistrale in Data Science per l'Economia e le Imprese

Sviluppo di modelli predittivi basati su algoritmi di Deep Learning per l'analisi automatica di ticket di assistenza

Development of predictive models based on Deep Learning algorithms for the automatic analysis of support tickets

Relatore:

Prof. Alex Mircoli

Correlatore:

Domenico Potena

Tesi di laurea di:

Serena Marini

A.A. 2023 / 2024

Indice

Capitolo 1

Introduzione

Nel settore della Grande Distribuzione Organizzata (GDO), affrontare e risolvere rapidamente le problematiche operative è cruciale per mantenere alta l'efficienza e la soddisfazione del cliente. Magazzini Gabrielli SpA è un'azienda familiare che gestisce una vasta rete di superstore, supermercati e punti vendita in franchising. Dopo aver consolidato la propria presenza nelle Marche e in Abruzzo, l'azienda è in continua espansione nel centro Italia. La gestione di questa rete complessa e diversificata comporta numerose sfide quotidiane, monitorate e registrate attraverso il servizio clienti.

I ticket di supporto, creati per segnalare problemi tecnici o operativi, rappresentano un'importante risorsa per migliorare le operazioni aziendali, in quanto la loro analisi consente di identificare aspetti critici e di ottenere indicazioni utili per interventi correttivi. Tuttavia, la grande quantità di dati generati dai ticket di assistenza rappresenta una sfida significativa. La loro analisi manuale risulta onerosa in termini di tempo e risorse e si dimostra inefficace nel garantire risposte tempestive. Per affrontare questa problematica, è necessario sviluppare tecniche di analisi automatica in grado di gestire e processare efficacemente grandi volumi di dati. L'introduzione di soluzioni data-driven per

l'automazione dell'analisi dei ticket non solo ridurrà i tempi di risposta, ma permetterà anche di identificare pattern ricorrenti e aree di intervento prioritario, supportando decisioni strategiche più informate.

La presente tesi si concentra sull'analisi testuale dei ticket di supporto di Magazzini Gabrielli SpA, con l'obiettivo di raggrupparli in base ai temi trattati e identificare le principali aree problematiche. A tal fine, saranno utilizzate tecniche di Text Mining per preprocessare il testo ed estrarre le informazioni più rilevanti, e tecniche di Machine Learning per raggruppare i dati in base alla loro similarità semantica, individuando così sottoargomenti all'interno delle categorie considerate. L'obiettivo finale è ottenere una comprensione approfondita delle problematiche riscontrate e fornire raccomandazioni basate sui dati. Questo permetterà all'azienda di focalizzarsi sulle aree di intervento più rilevanti, contribuendo a migliorare le prestazioni complessive dei punti vendita, con un impatto positivo sulla soddisfazione del cliente.

Il presente elaborato si articola in diversi capitoli che esplorano i vari aspetti del processo di analisi. Il capitolo successivo definirà l'obiettivo del progetto, il contesto aziendale di Magazzini Gabrielli SpA e la metodologia di analisi adottata, offrendo così una panoramica completa di tutte le fasi del processo.

Il terzo capitolo sarà dedicato alla fase sperimentale, descrivendo gli esperimenti condotti e approfondendo le tecniche di clustering semantico utilizzate. Verranno presentate le metodologie impiegate e i risultati ottenuti, fornendo una visione dettagliata delle procedure e delle decisioni prese.

Il quarto capitolo riguarderà l'analisi dei gruppi ottenuti dal clustering, con particolare attenzione alla valutazione della similarità semantica e alla descrizione delle problematiche riscontrate.

Infine, il quinto capitolo raccoglierà le conclusioni del lavoro, riassumendo i principali risultati e valutando l’efficacia delle tecniche applicate, oltre a fornire suggerimenti per sviluppi futuri.

Capitolo 2

Metodologia

La presente tesi si propone di migliorare la gestione delle interazioni con il servizio clienti di Magazzini Gabrielli SpA attraverso un'analisi avanzata dei ticket di supporto. Le richieste di assistenza e le segnalazioni di problemi rappresentano una fonte di informazione sulle difficoltà operative dell'azienda. In un settore altamente competitivo e dinamico come quello in cui opera Magazzini Gabrielli, gestire efficacemente i ticket di supporto è cruciale per mantenere un elevato livello di soddisfazione del cliente.

In un'ottica di elaborazione dei dati sono da considerare l'elevato volume e la varietà delle problematiche riscontrabili, che rendono l'identificazione manuale dei problemi ricorrenti un lavoro eccessivamente dispendioso per l'azienda. Questo contesto evidenzia la necessità di adottare un sistema avanzato di analisi, capace di automatizzare e migliorare l'interpretazione dei dati.

Una soluzione a questa esigenza è rappresentata dal clustering semantico, una tecnica avanzata di analisi di dati che permette di raggruppare testi, come i ticket di supporto, in base alla loro similarità semantica. L'obiettivo è individuare gruppi omogenei, ovvero cluster composti da ticket riferiti allo stesso argomento. Questo

approccio prevede la trasformazione dei ticket in rappresentazioni vettoriali, che vengono utilizzate per il clustering e permettono di identificare relazioni e temi comuni. Una volta ottenuti i risultati, sarà possibile analizzarli per scoprire pattern e tendenze, fornendo informazioni preziose per la gestione e il miglioramento dei servizi offerti da Magazzini Gabrielli.

Nel dettaglio, la metodologia di analisi adottata si sviluppa attraverso diverse fasi, che vanno dalla preparazione dei dati alla valutazione dei gruppi, come illustrato in Figura 1.

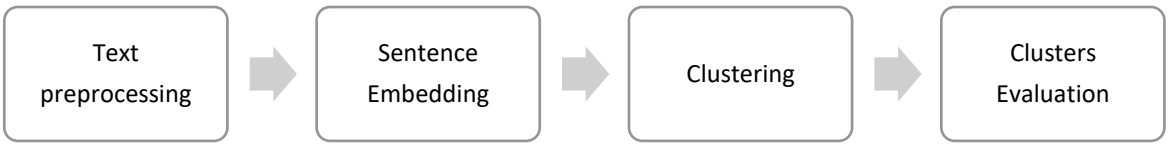

Figura 1: Metodologia di analisi adottata

I dati analizzati sono stati ottenuti tramite un precedente collegamento con il database dell'azienda Magazzini Gabrielli SpA, che offre un'ampia raccolta di ticket di supporto, arricchita da informazioni aggiuntive. L'analisi si focalizza sul contenuto testuale dei messaggi, che contengono le descrizioni dei problemi segnalati e le risposte degli operatori del servizio, fornendo dettagli

potenzialmente preziosi per comprendere le richieste e le problematiche riscontrate.

Tuttavia, i ticket presentano una varietà di elementi che non sono direttamente rilevanti per l'obiettivo descritto e che, anzi, possono complicare notevolmente il processo di estrazione di informazioni utili. Ad esempio, URL, hashtag e link a immagini, sebbene possano avere un valore aggiunto per il personale interno o per la documentazione, non sono necessari per l'analisi del problema. Tali elementi, se non gestiti adeguatamente, possono compromettere l'accuratezza dei modelli e ostacolare l'individuazione delle informazioni rilevanti secondo l'obiettivo specifico di analisi.

Per migliorare la qualità dei dati, viene implementata una fase di pre-processing dei ticket mediante tecniche di Natural Language Processing (NLP). Questo passaggio è cruciale per trasformare il testo grezzo in un formato più strutturato e pulito, eliminando elementi non informativi e normalizzando i dati.

Sono svolte diverse attività standard di pulizia del testo, tra cui la rimozione di segni di punteggiatura, caratteri numerici e stopword, ovvero parole rilevate di frequente ma con scarso apporto informativo. Sono inoltre eseguite operazioni specifiche legate alla struttura dei ticket di supporto: sono eliminati contatti del mittente e del destinatario, insieme a simboli e abbreviazioni derivanti dai canali

di comunicazione. Allo stesso modo, vengono rimossi simboli di formattazione e riferimenti esterni, come tag HTML e CSS, URL e hashtag. Infine, attraverso un'analisi manuale dei messaggi degli utenti, sono individuate e rimosse le espressioni di saluto e ringraziamento più frequentemente utilizzate.

Un ulteriore step è la lemmatizzazione, processo che consiste nella riduzione delle singole parole alla loro forma base o lemma. L'eliminazione di flessioni grammaticali, come plurali, tempi e modi verbali, contribuisce a semplificare e uniformare il testo. La gestione di molteplici forme della stessa parola, infatti, può aumentare la complessità dei modelli di analisi e influire negativamente sull'accuratezza dei risultati. La lemmatizzazione, dunque, facilita l'individuazione dei concetti principali, fornendo una rappresentazione più chiara e coesa dei dati testuali.

Dopo il pre-processing, la fase successiva sarebbe stata il raggruppamento dei ticket in gruppi semanticamente omogenei, con l'obiettivo di identificare insiemi di ticket che condividono problematiche o richieste simili, facilitando così una gestione più efficiente delle risorse. Tuttavia, una delle sfide principali riscontrate è stata legata alla natura testuale dei dati. La maggior parte degli algoritmi di Clustering, infatti, è progettata per operare su dati numerici e non può elaborare

direttamente contenuti testuali. Pertanto, si è ritenuto necessario convertire le frasi in rappresentazioni numeriche adatte all'elaborazione di tali algoritmi.

La soluzione adottata coinvolte l'impiego di tecniche di Sentence Embedding, che consentono di convertire frasi, paragrafi o interi documenti in vettori numerici continui, preservandone il contenuto semantico. Attraverso questo processo, le frasi possono essere mappate in uno spazio vettoriale ad alta dimensionalità, in cui la posizione di ciascun vettore riflette le caratteristiche semantiche del testo originale. Potenzialmente testi semanticamente simili sono rappresentanti da vettori vicini tra loro, rendendo possibile un confronto diretto.

La vettorizzazione mediante Sentence Embedding consente agli algoritmi di clustering di basarsi sull'affinità semantica e, in questo caso, di raggruppare ticket riferiti allo stesso problema o a problemi simili. Per garantire risultati ottimali, sono stati sperimentati diversi modelli di Sentence Embedding, ciascuno dei quali si basa su tecnologie avanzate nel campo dell'elaborazione del linguaggio naturale (NLP). L'analisi delle performance di questi diversi modelli ha avuto come scopo quello di individuare quale fosse il più adatto per il compito specifico di clustering dei ticket. In particolare, si è tenuto conto di vari fattori, tra cui la qualità delle rappresentazioni vettoriali generate dai modelli e le prestazioni complessive in termini di coesione interna e separazione tra gruppi. La coesione interna misura

quanto i ticket all'interno di ciascun cluster siano semanticamente simili tra loro, mentre la separazione tra cluster valuta la distanza semantica tra i gruppi, garantendo che ciascun gruppo rappresenti una categoria ben distinta dalle altre. Questo approccio ha permesso di identificare il modello che meglio preservava le informazioni semantiche nei vettori, garantendo una maggiore efficacia nel processo di clustering.

Successivamente, i cluster generati sono stati confrontati e analizzati utilizzando sia un insieme di metriche specifiche per il clustering, sia informazioni specifiche relative ai ticket. Tale valutazione ha consentito di individuare il numero ottimale di gruppi e di selezionare la rappresentazione vettoriale più adeguata per rispondere all'obiettivo prefissato.

Capitolo 3

3.1 Descrizione del dataset e Text Pre-processing

Il dataset oggetto di analisi raccoglie informazioni chiave sul contenuto e le caratteristiche dei ticket di supporto, come descritto nella Tabella 1.

Variabile	Tipo	Descrizione
Ticket ID	int	Identificativo univoco del ticket
LDTEXT	testo	Testi inseriti dagli utenti all'atto di apertura del ticket
ReportedBy	testo	Operatore che ha aperto il ticket
Topic First Level	testo	Oggetto del problema riscontrato che ha portato all'apertura di un ticket
Topic Third Level	testo	Modalità di risoluzione del ticket

Tabella 1: Descrizione delle variabili del dataset

Un'analisi dettagliata della variabile "Topic First Level", che identifica la categoria del problema riscontrato all'apertura del ticket, evidenzia che le categorie più frequenti sono HW-POSTAZIONE CASSA e SW-VISUALSTORE. La prima categoria si

riferisce a problemi legati ai componenti hardware, in particolare alle postazioni cassa, che rappresentano una parte critica del sistema operativo per il punto vendita. La seconda categoria riguarda invece i problemi software relativi al gestionale Visualstore, un software utilizzato per la gestione delle attività commerciali.

La presente analisi si concentra sulla categoria di problemi hardware, che risulta essere la più frequente nel dataset. In particolare, sono stati esaminati 9.523 ticket, aperti da 372 utenti diversi. Queste richieste di assistenza sono state poi risolte utilizzando 16 diverse modalità di intervento, che sono riportate nella variabile “Topic Third Level”.

Da una prima analisi esplorativa, emerge che la lunghezza media dei ticket è di 21 parole, il che suggerisce che, nella maggior parte dei casi, gli utenti forniscono descrizioni piuttosto concise del problema riscontrato. È importante comunque sottolineare che esiste una notevole variabilità nella lunghezza delle descrizioni. Alcuni ticket contengono descrizioni estremamente brevi, composte da una sola parola, mentre altri si estendono fino a 1.444 parole, con un livello di dettaglio molto più elevato. Questa variazione è confermata dalla deviazione standard di 34,85 parole, che riflette la dispersione dei dati rispetto alla media e suggerisce che gli utenti adottano stili comunicativi diversi nella descrizione dei problemi, influenzando così la quantità di informazioni disponibili per l’analisi.

3.2 Text Pre-processing

Il processo di pre-processing, implementato in Python, inizia con la tokenizzazione del testo, passaggio per suddividere il contenuto testuale in singole parole o token. Questo è eseguito utilizzando la libreria *spaCy*, in combinazione con il modello "it_core_news_sm" specifico per l'italiano.

Successivamente, viene effettuata una pulizia approfondita del testo. In questa fase, i tag HTML e CSS vengono rimossi tramite la libreria *BeautifulSoup*, in modo da trattare testi provenienti da varie fonti ed estrarre solo il contenuto testuale rilevante. Gli hyperlink e gli hashtag sono eliminati e sostituiti con spazi bianchi utilizzando la libreria *re* per la definizione di espressioni regolari. Infine, si rimuovono simboli di formattazione, segni di punteggiatura e altri caratteri speciali e si procede alla conversione del testo in minuscolo per standardizzare le parole.

Un altro passaggio cruciale è l'eliminazione delle stopword. Utilizzando liste predefinite per la lingua italiana e fornite dalla libreria *nltk*, si rimuovono parole comuni ma di poco valore informativo, come articoli e preposizioni. Inoltre, vengono eliminati saluti e frasi di cortesia comuni come "buongiorno" e "grazie", per concentrarsi sui termini di maggiore rilevanza e migliorare così la qualità

complessiva del dataset. Questo aiuta a ridurre il rumore e a garantire che l'analisi si focalizzi su contenuti significativi e utili.

Un ulteriore step è la lemmatizzazione, effettuata utilizzando nuovamente la libreria spaCy. Questo passaggio riduce le parole alla loro forma base, consentendo di normalizzare le varie forme morfologiche e di trattare varianti grammaticali come coniugazioni verbali e plurali dei sostantivi come se fossero la stessa parola.

I ticket vengono quindi filtrati mantenendo solo i token alfabetici e quelli non presenti nelle liste di stopword. Inoltre, vengono esclusi i nomi propri, che sono identificati attraverso il part-of-speech tagging (POS tagging). Questo processo assegna etichette grammaticali alle parole, permettendo di distinguere tra diverse parti del discorso come nomi, verbi e aggettivi.

Questi approcci mantengono nel testo finale soltanto token ad alto contenuto informativo e, dunque, potenzialmente rilevanti.

La Tabella 2 mostra un esempio dell'effetto della fase di pre-processing applicata a un campione di ticket selezionati.

TICKETID	**ReportedDate**	**LDTEXT**	**testo_pulito**
56611	30/04/2021	cassa 3 bloccata durante trxsig gianluca cv	cassa bloccare
71813	19/04/2022	cassa 1 si blocca spessorem_daniele kp	cassa bloccare

82025	03/12/2022	cassa 1 bloccata ref_emanuela rp	cassa bloccare
83631	12/01/2023	come ieri la cassa 8 si e bloccata	cassa bloccare
88252	31/03/2023	1140 cassa 3 bloccata grazie	cassa bloccare

Tabella 2: Pre-processing su ticket di esempio

Il codice implementato ha rimosso frasi di cortesia come “grazie” e “cordiali saluti”, caratteri numerici, nomi propri e stopword. Inoltre, attraverso la lemmatizzazione, parole come “bloccata” e “blocca” sono state ridotte alla loro forma base “bloccare”. Grazie a questi passaggi, i ticket di esempio sono stati tutti convertiti nella medesima frase “cassa bloccare”. Questo approccio ha ridotto la variabilità iniziale dei termini utilizzati nei ticket, permettendo, al contempo, di mantenere le informazioni rilevanti senza il rumore creato da elementi non significativi.

Il risultato della fase di pre-processing è un testo depurato da elementi non informativi, costituito da token significativi e lemmatizzati, che vengono poi uniti in una singola stringa.

Per garantire ulteriormente la qualità dei dati, si applica una fase finale di filtraggio: i documenti con meno di due parole nel testo lemmatizzato sono

identificati come non sufficientemente informativi e rimossi dal dataset. Dopo il pre-processing, infatti, la presenza di un solo token in un ticket indica una forte riduzione dell'informazione utile, poiché una parola isolata, priva di contesto, non apporta valore all'analisi testuale. Il risultato finale è un dataset che rappresenta un corpus testuale di maggiore qualità, pronto per le analisi successive.

Il processo, una volta applicato al dataset, ha portato alla rimozione di 115 ticket, portando l'analisi a concentrarsi sui restanti 9.408 ticket. Inoltre, il pre-processing ha determinato una riduzione della lunghezza media dei ticket, che ora è pari a 11 token per documento. Tuttavia, la lunghezza dei ticket continua a variare in modo significativo, con un minimo di 2 e un massimo di 935 parole per ticket. Nonostante il processo di pre-processing abbia contribuito a ridurre la variabilità nella lunghezza dei documenti, la dispersione rimane piuttosto ampia, con una deviazione standard pari a 21,76. Questo riflette la diversità intrinseca del dataset originale.

3.3 Sentence Embedding

Dopo aver migliorato la qualità dei dati tramite il pre-processing, l'analisi si concentra sul contenuto testuale dei ticket pre-elaborati. L'obiettivo è ora rispondere al seguente requisito di progetto: utilizzare tecniche di clustering per

identificare gruppi di ticket in modo che i ticket all'interno dello stesso gruppo trattino argomenti simili, mentre quelli appartenenti a gruppi distinti facciano riferimento a temi diversi.

Prima di procedere al clustering vero e proprio, si procede alla vettorizzazione del testo mediante Sentence Embedding, un approccio usato nell'ambito dell'elaborazione del linguaggio naturale (NLP), che rappresenta ciascuna frase come un vettore di numeri reali (embedding) in uno spazio multidimensionale. La conversione delle frasi in vettori numerici consente di catturare il significato semantico e il contesto delle frasi in un formato facilmente interpretabile e utilizzabile da modelli di Machine Learning.

Nel corso degli anni, l'elaborazione del linguaggio naturale e le tecniche di embedding hanno visto notevoli progressi, con significativi miglioramenti nella rappresentazione e nell'analisi dei testi.

Word2Vec è stata una delle tecniche più importanti e influenti nel campo NLP nel decennio scorso. Introdotto nel 2013 da un team di ricercatori guidato da Tomas Mikolov presso Google, ha rivoluzionato il modo in cui le parole vengono rappresentate e comprese dai modelli di analisi. Tale tecnica si basa su reti neurali e crea rappresentazioni vettoriali dense delle parole, note come "word embeddings". Questi vettori catturano relazioni semantiche e sintattiche tra le

parole, fornendo una base molto più ricca per il trattamento del linguaggio rispetto ai metodi tradizionali come Bag-of-Words e TF-IDF, che non erano in grado di cogliere il significato profondo e il contesto in cui le parole venivano utilizzate. L'idea centrale è che parole con significati simili compaiano in contesti simili, e questa associazione viene riflessa nella vicinanza dei loro vettori nello spazio vettoriale. Grazie a questa rappresentazione, parole con significato affine risultano vicine tra loro nello spazio multidimensionale, evidenziando la loro somiglianza semantica.

L'anno successivo il concetto di Word2Vec è stato esteso con lo sviluppo di Doc2Vec, noto anche come Paragraph Vector (Mikolov et al., 2014). Questa tecnica di deep learning si propone di rappresentare documenti o paragrafi come vettori di dimensione fissa, ampliando così l'idea degli embedding dalle singole parole a unità di testo più lunghe. Con Doc2Vec è possibile mappare interi documenti in uno spazio vettoriale continuo, dove testi simili risultano vicini tra loro. Questo approccio contente di mantenere una visione coerente e integrata del contenuto testuale, facilitando compiti complessi come la classificazione di documenti e l'analisi della similarità.

Il Doc2Vec può essere implementato tramite due approcci: Distributed Memory (PV-DM) e Distributed Bag of Words (PV-DBOW).

La variante PV-DBOW ha l'obiettivo di predire le parole estratte da un documento utilizzando la rappresentazione vettoriale del documento stesso, senza considerare l'ordine delle parole circostanti. Durante il processo di addestramento, viene ottimizzato il vettore che rappresenta il documento per migliorare la capacità di prevedere correttamente le parole che lo compongono. Questo approccio produce un vettore di embedding che cattura il significato generale del documento, rappresentazione d'insieme che riflette il contenuto e il contesto complessivo del testo.

La variante PV-DM si differenzia per il suo approccio: mira a prevedere una parola del contesto – ovvero le parole vicine a una data parola – utilizzando sia il vettore del documento e sia i vettori delle parole circostanti. Questo si basa sull'idea di apprendere una rappresentazione vettoriale per ogni frase, paragrafo o documento tenendo conto del contesto in cui appaiono le parole. Durante l'addestramento, i vettori delle parole e dei documenti vengono aggregati, tramite media o concatenazione, in un unico vettore che rappresenta sia il documento che il contesto locale. L'ottimizzazione si focalizza sulla capacità di questo vettore di prevedere correttamente ciascuna parola contenuta nel documento.

Mentre questa architettura effettua previsioni basate sia sul significato generale del documento sia sulla struttura locale delle parole, l'approccio DBOW è più semplice e diretto. In questo modello, per la previsione delle parole si utilizza solo

il vettore del documento, senza tenere conto del contesto delle parole circostanti. Questo approccio semplifica l'addestramento e riduce la complessità computazionale, poiché non richiede l'aggiornamento dei vettori delle parole in base al loro contesto. Le due architetture a confronto sono rappresentate nella Figura 2.

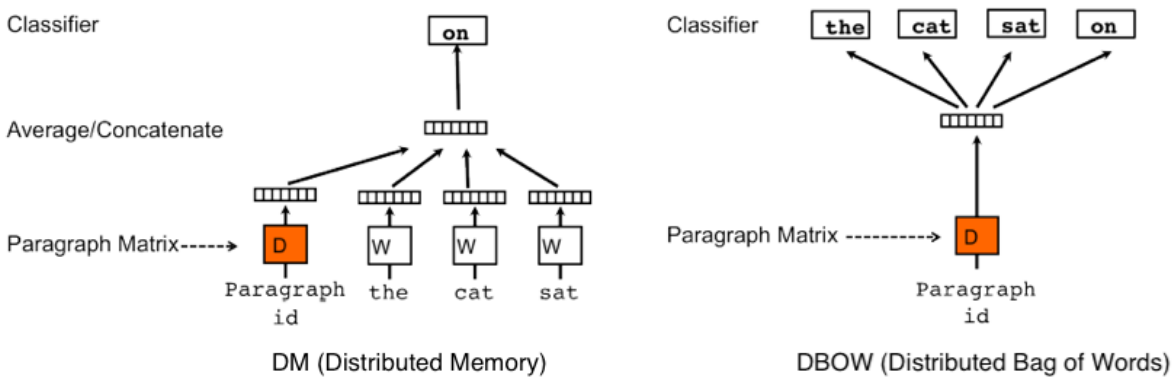

Figura 2: Framework per l'apprendimento dei vettori di paragrafo

Nel 2018, Google ha fatto un ulteriore progresso nel campo degli embedding con l'introduzione del Universal Sentence Encoder (USE) (Cer et al., 2018).

Questo modello rappresenta un'evoluzione rispetto ai precedenti approcci, grazie al suo pre-addestramento su un ampio e vario corpus di dati. Questo gli consente di fornire rappresentazioni semantiche di alta qualità pronte all'uso, eliminando la necessità di ulteriori fasi di addestramento specifico. Il modello principale dell'USE

si basa sull'architettura Transformer, che sfrutta meccanismi di attenzione per ottenere rappresentazioni contestualizzate delle parole all'interno di una frase o di un documento. Questo approccio permette di catturare le relazioni tra parole in una frase e tra frasi diverse, migliorando la comprensione del contesto e la qualità delle rappresentazioni semantiche generate.

L'USE è facilmente accessibile su TensorflowHub, una piattaforma che offre la possibilità di scaricare e utilizzare modelli pre-addestrati. Questa disponibilità immediata lo rende uno strumento pratico e facilmente integrabile in diverse applicazioni di NLP, motivo per cui è stato scelto come punto di partenza per la presente analisi.

Un aspetto critico dell'utilizzo dell'Universal Sentence Encoder (USE) nei compiti di clustering è la generazione di vettori di dimensione fissa, precisamente di 512 dimensioni. Sebbene queste rappresentazioni siano molto efficaci nel catturare il significato e il contesto del testo, l'alta dimensionalità può dare origine al *curse of dimensionality*. Questo fenomeno si riferisce a una serie di problemi che emergono quando i dati sono rappresentati in spazi a dimensioni molto elevate.

Nel contesto del clustering, il *curse of dimensionality* può causare un deterioramento delle prestazioni in diversi modi. In spazi ad alta dimensione, la densità dei dati può diminuire drasticamente, il che significa che i punti tendono

ad occupare solo una piccola regione dello spazio. Inoltre, la nozione di distanza o similitudine tra punti diventa meno significativa: due punti che sembrano vicini in uno spazio a bassa dimensionalità potrebbero risultare molto distanti in uno spazio ad alta dimensionalità, rendendo difficile identificare i cluster naturali e le loro strutture. Inoltre, l'elaborazione di dati in spazi ad alta dimensionalità comporta un aumento significativo dei requisiti computazionali. Alcuni algoritmi di apprendimento automatico possono diventare estremamente lenti o addirittura impraticabili quando si lavora con molte dimensioni, a causa dell'enorme quantità di operazioni necessarie per gestire e analizzare i dati. Questo incremento dei requisiti computazionali può ulteriormente ostacolare la capacità di eseguire clustering in modo efficiente.

Per affrontare questa sfida, si seguiranno due approcci distinti e si confronteranno di volta in volta i risultati ottenuti.

Il primo prevede l'applicazione di una tecnica di riduzione della dimensionalità sugli embedding generati dal modello USE, in particolare la Principal Component Analysis (PCA). Questa scelta si basa sulla capacità della PCA di semplificare la rappresentazione di dati complessi e ad alta dimensionalità, preservando al contempo la maggior parte delle informazioni rilevanti.

La PCA raggiunge questo obiettivo riducendo il numero di variabili attraverso la trasformazione dei dati originali in uno spazio a dimensioni inferiori. Questo processo si basa sulla creazione di nuove variabili, chiamate componenti principali, che sono combinazioni lineari delle feature originali. Le componenti principali vengono ordinate in base alla quantità di varianza che spiegano: l'obiettivo è di mantenere solo quelle componenti che catturano la maggior parte dell'informazione utile contenuta nei dati. In questo modo, la PCA non solo preserva gran parte della variabilità e delle informazioni cruciali del dataset originale, ma semplifica anche la struttura dei dati. Questo approccio riduce significativamente la complessità dello spazio di clustering, rendendolo meno complesso e quindi più gestibile

Il secondo approccio implica l'uso di una tecnica alternativa per la rappresentazione testuale, il Doc2Vec. A differenza dell'Universal Sentence Encoder, il Doc2Vec permette di definire a priori la dimensionalità dei vettori, offrendo così maggiore flessibilità e controllo sulla rappresentazione dei dati.
Questo approccio consente di scegliere dimensioni dei vettori più contenute, che possono mitigare gli effetti negativi associati alla *curse of dimensionality*.

La valutazione dei diversi approcci si basa sull'efficacia nel compito specifico di clustering. L'algoritmo scelto per questo contesto è il KMeans, una delle tecniche più ampiamente adottate nell'ambito dell'apprendimento non supervisionato. K-Means è approccio di clustering basato su prototipi, in cui i gruppi o cluster sono definiti da rappresentanti centrali noti come centroidi. Ha l'obiettivo di suddividere un insieme di dati in un numero predefinito di gruppi o cluster, in modo tale che gli elementi all'interno di ciascun gruppo siano il più simili possibile tra loro, mentre gli elementi appartenenti a cluster diversi siano il più dissimili possibile. Questo viene realizzato ottimizzando al contempo due criteri: la minimizzazione della varianza intra-cluster e la massimizzazione della varianza inter-cluster.

Per analizzare e confrontare i risultati ottenuti dalle diverse tecniche di Sentence Embedding, vengono utilizzate congiuntamente le seguenti metriche:

- **Within-cluster sum of squares (WCSS)**: questa metrica intra-cluster misura la dispersione dei punti dati all'interno di ciascun cluster rispetto al loro centroide. È calcolata come la media delle distanze euclidee al quadrato tra i punti dati e il centroide del cluster a cui appartengono. Valori più bassi di WCSS suggeriscono che i punti all'interno di un gruppo sono più vicini

tra loro e al centroide, denotando una struttura di cluster più compatta e ben definita.

- **Between-cluster sum of squares (BCSS)**: questa metrica inter-cluster misura la separabilità tra i cluster, garantendo che i gruppi siano il più distinti possibile. La BCSS è calcolata come la media delle distanze euclidee tra tutte le coppie di centroidi dei cluster. La massimizzazione del BCSS indica che i centroidi dei cluster sono più distanti tra loro, suggerendo una maggiore separabilità e distintività dei gruppi.

- **Davies-Bouldin Index**: Calcolato utilizzando la funzione davies_bouldin_score di scikit-learn, questo indice misura la qualità del clustering valutando la similarità media di ogni cluster con il cluster a lui più simile. La similarità è definita come il rapporto tra la dispersione interna dei cluster e la distanza tra i centroidi dei cluster. Un valore basso di DBI suggerisce che i cluster sono ben distinti e ben separati gli uni dagli altri, mentre un valore più alto indica che i cluster sono più simili tra loro.

3.4 Universal Sentence Encoder: confronto dei risultati di clustering

Come descritto in precedenza, il primo approccio consiste nella generazione di sentence embedding tramite Universal Sentence Encoder. L'implementazione inizia con il caricamento del modello pre-addestrato tramite TensorFlow Hub, utilizzando un URL specifico che punta al modello disponibile online. I ticket da analizzare vengono convertiti in una lista e passati al modello per ottenere gli embedding, ovvero rappresentazioni numeriche dei testi in uno spazio a 512 dimensioni. Per garantire che tutte le dimensioni dei vettori siano comparabili, i dati vengono normalizzati utilizzando il MinMaxScaler della libreria scikit-learn, che scala i valori dei vettori in un intervallo standardizzato.

Il processo che viene di seguito descritto è stato utilizzato sia per il clustering con l'Universal Sentence Encoder da solo, sia per l'USE combinato con la tecnica di riduzione della dimensionalità.

Questo approccio ha previsto un'esplorazione su un ampio intervallo di possibili numeri di cluster e l'implementazione di una ricerca a griglia, finalizzata a testare diverse configurazioni di PCA. Nello specifico, sono stati considerati un numero di componenti principali pari a 10, 15, 20, 25, 30, 50, 90, 130, 170 e 210. L'ampiezza dell'intervallo esplorato ha lo scopo di individuare la configurazione ottimale, che

bilanci efficacemente la quantità di informazioni catturate e la complessità dello spazio di clustering, senza aumentare eccessivamente i costi computazionali.

La Tabella 3 mostra i risultati ottenuti sia con l'Universal Sentence Encoder che con l'USE seguito dalla Principal Component Analysis, utilizzando vettori ridotti a 50 dimensioni. Il confronto è stato effettuato su un intervallo di cluster che va da 2 a 300, con l'obiettivo di analizzare l'andamento delle metriche di interesse all'aumentare del numero di cluster.

La scelta di presentare i dati relativi a vettori a 50 dimensioni è stata guidata dall'analisi della varianza cumulata: con tali dimensioni si riesce a catturare il 68,91% della variabilità totale, offrendo così un bilanciamento tra la capacità di rappresentare la maggior parte delle informazioni contenute nel dataset e la riduzione della complessità del problema.

	BCSS		**WCSS**		**db_index**	
n_cluster	**USE**	**USE + PCA**	**USE**	**USE + PCA**	**USE**	**USE + PCA**
2	2,34	2,34	67032,18	44351,25	3,28	2,66
12	2,99	3,09	9434,53	5659,00	3,58	2,73
22	3,41	3,20	4784,22	2751,58	3,17	2,53
32	3,32	3,26	3156,43	1773,63	3,28	2,42
42	3,44	3,30	2338,16	1283,88	3,24	2,43
52	3,54	3,38	1837,36	1002,93	3,16	2,46

62	3,63	3,50	1509,25	808,45	3,11	2,39
72	3,60	3,48	1282,20	678,56	3,17	2,38
82	3,63	3,51	1108,94	582,06	3,09	2,47
92	3,73	3,54	973,86	508,88	3,09	2,50
100	3,68	3,50	887,03	459,94	3,11	2,49
150	3,90	3,54	562,26	285,38	3,01	2,48
200	3,97	3,59	408,38	203,69	2,94	2,46
250	3,98	3,68	316,48	156,27	2,90	2,44
300	4,02	3,72	258,16	126,11	2,86	2,40

Tabella 3: Confronto tra USE e USE seguito da PCA

Quando il numero di cluster è sufficientemente elevato, ci si aspetta che la distanza media tra i centroidi (BCSS) tenda a diminuire. Questo accade perché ciascun cluster tende a focalizzarsi su una porzione più specifica e ristretta dello spazio dei dati, il che porta i centroidi ad avvicinarsi tra loro per meglio riflettere la distribuzione dei dati all'interno di ciascuna regione. Tuttavia, i dati ottenuti mostrano un comportamento differente: sia con l'USE che con l'USE seguito dalla PCA, si osserva un incremento continuo della BCSS. Questo indica che l'aumento del numero di cluster non migliora la separazione tra i cluster come previsto. Anche riducendo la dimensionalità da 512 a 50, il problema persiste. Questo suggerisce che l'utilizzo della PCA non abbia migliorato le performance di clustering in termini di separazione tra cluster.

L’andamento della WCSS, invece, è coerente con le aspettative teoriche per entrambe le tecniche. Quando si aumenta la frammentazione dei dati, infatti ci si aspetta una riduzione della variabilità all’interno di ciascun cluster, riflettendo una maggior compattezza dei gruppi. Dai risultati, emerge che il clustering con USE seguita dalla PCA presenta valori di WCSS in generale più bassi, il che può essere l’effetto della riduzione della dimensionalità e/o di una migliore compattezza dei cluster.

Un effetto analogo si osserva nell’andamento del Davies-Bouldin Index (DB Index). I dati suggeriscono che la PCA abbia avuto un impatto positivo sulla qualità del clustering. Risulta evidente però che a fronte di un significativo aumento del numero di cluster la variazione dell'indice risulta limitata, segnalando una riduzione meno marcata della qualità rispetto alle aspettative.

In conclusione, l’uso esclusivo dell’Universal Sentence Encoder ha portato a performance insoddisfacenti, come dimostrato dall’andamento anomalo della distanza tra i centroidi e dai valori elevati di WCSS e DB Index. Sebbene l'applicazione della PCA abbia comportato una riduzione dei valori di WCSS e del Davies-Bouldin Index, tali miglioramenti non sono stati particolarmente significativi. Ciò suggerisce che, nonostante la riduzione della dimensionalità, le

prestazioni complessive del clustering preceduto dall'uso dell'Universal Sentence Encoder rimangono ancora insoddisfacenti.

3.5 Doc2Vec: implementazione del Sentence Embedding

Per ottenere una comprensione più approfondita e migliorare la qualità del clustering, si procede con la generazione e il test degli embedding tramite Doc2Vec.

Sebbene questa tecnica richieda un processo di addestramento specifico sul corpus di dati e non sia immediatamente utilizzabile come l'USE, offre vantaggi significativi in termini di personalizzazione. Consente, infatti, di regolare vari iperparametri, inclusa la dimensionalità degli embedding, permettendo così di adattare la rappresentazione dei dati alle esigenze specifiche del compito di clustering.

Il modello Doc2Vec è implementato tramite la libreria Gensim e richiede la definizione e la regolazione di vari iperparametri. Tra questi, i più rilevanti sono:

- **dm:** definisce la variante del Doc2Vec da utilizzare, PV-DM o PV-DBOW.
- **vector_size**: determina la dimensionalità dei vettori di embedding.

- **window**: indica la dimensione della finestra di contesto, ovvero il numero di parole circostanti che vengono utilizzare per prevedere la parola target.
- **min_count**: definisce la frequenza minima di occorrenza di una parola nel corpus di testo affinché venga inclusa nel vocabolario del modello. Parole con una frequenza inferiore alla soglia stabilita vendono ignorate.

Nel contesto dell'apprendimento non supervisionato, la valutazione della qualità dei modelli è resa complessa dall'assenza di etichette predefinite che permettano un confronto diretto con le previsioni. Di conseguenza, è necessario adottare metodi alternativi per misurare l'efficacia dei modelli, continuando a mantenere l'attenzione sul task di raggruppamento.

Una delle strategie utilizzate è la valutazione dell'autosimilarità, che si concentra sulla capacità del modello di riconoscere e classificare correttamente i documenti all'interno del proprio corpus di addestramento. In pratica, il modello viene prima addestrato e successivamente utilizzato per inferire nuovi vettori per ciascun documento del corpus di addestramento stesso. Questi vettori inferiti vengono poi confrontati con quelli di addestramento per determinarne la somiglianza. Il confronto viene effettuato tramite il metodo most_similar, che permette di ordinare i vettori in base alla similarità coseno rispetto a un vettore di riferimento.

La similarità coseno è una misura che valuta la somiglianza relativa tra i vettori, escludendo l'influenza della loro lunghezza. Essa calcola l'angolo tra due vettori nello spazio vettoriale e restituisce un valore compreso tra -1 e 1. Una similarità pari a 1 indica che una somiglianza perfetti tra i vettori, mentre una similarità pari a -1 indica che i vettori sono diametralmente opposti.

L'analisi dei ranghi di similarità, ovvero le posizioni di ciascun documento nella lista ordinata in base alla similarità coseno, offre un primo strumento di valutazione. L'aspettativa è che il vettore inferito di un documento risulti più simile a se stesso rispetto agli altri documenti del corpus di addestramento, o almeno che compaia tra i primi ranghi di similarità. Anche se questo approccio non fornisce una misura assoluta di accuratezza, rappresenta un indicatore della coerenza e plausibilità dei risultati prodotti dal modello.

Un secondo metodo di valutazione prevede un'analisi manuale dei contenuti testuali dei ticket: utilizzando la cosine similiarity, è possibile esaminare direttamente i documenti che il modello ha classificato come più o meno simili. Questo approccio consente di valutare, attraverso la lettura dei testi, se le rappresentazioni generate dai modelli di sentence embedding sono in linea con le aspettative umane. Analizzando i risultati ottenuti tramite il metodo most_similar,

è possibile visualizzare e confrontare i contenuti testuali dei documenti per verificare se le rappresentazioni catturano adeguatamente le relazioni semantiche tra i documenti, offrendo così una valutazione più soggettiva e interpretativa della qualità del modello.

Il codice implementato ha avuto l'obiettivo di individuare, mediante un approccio basato su grid search, la configurazione di parametri che garantisse le migliori prestazioni.

Ogni documento è stato inizialmente rappresentato come un TaggedDocument, struttura di input composta da una lista di parole presenti nel documento e un tag univoco che identifica il ticket all'interno del corpus. Per quanto riguarda l'implementazione di Doc2Vec, sono stati selezionati tre parametri chiave da testare:

- **vector_size**: le dimensioni dei vettori di embedding considerate sono 10, 15, 20, 25, 30, 50, 90, 130, 170 e 210 per garantire la comparabilità con i risultati ottenuti tramite l'Universal Sentence Encoder.

- **min_count**: analizzando la frequenza delle parole nel corpus, sono stati selezionati i valori 5 e 10 al fine di escludere quelle parole che compaiono troppo raramente per fornire informazioni significative.

- **window**: per bilanciare la qualità della rappresentazione del contesto, in linea con la lunghezza media e la variabilità dei documenti nel corpus, i valori selezionati sono 5 e 10.

Il codice itera quindi su ciascuna combinazione dei parametri selezionati, addestrando le due varianti del modello Doc2Vec: PV-DM e PV-DBOW. Durante questo processo, vengono costruiti i vocabolari e avviato l'addestramento per entrambe le varianti. Una volta completato l'addestramento, i modelli vengono valutati: per ogni documento, viene inferito un vettore e confrontato con i vettori dei documenti di addestramento utilizzando la similarità coseno.

L'obiettivo è verificare quante volte il documento di addestramento appare al rango 0 nella lista dei documenti più simili a quello inferito. Questo indicatore mostra la capacità del modello di riconoscere i documenti in base alla loro rappresentazione. Per ciascuna combinazione di parametri, il codice crea un dizionario per tracciare queste occorrenze.

Questo approccio si concentra principalmente sulla similarità di un documento con sé stesso, e non considera completamente la capacità del modello di distinguere tra documenti con contenuti semanticamente diversi. Per ottenere una valutazione più completa, viene effettuata un'analisi manuale di alcuni risultati. A partire da ticket selezionati casualmente, si esaminano i contenuti dei documenti classificati come i più simili e quelli meno simili. Questo processo permette di valutare visivamente quanto il modello riesca a distinguere tra documenti con contenuti simili e quelli con contenuti dissimili.

Dai risultati ottenuti, emerge che la variante PV-DM presenta una percentuale superiore di documenti riconosciuti al rango 0 nella maggior parte delle configurazioni testate. L'analisi approfondita dei contenuti testuali dei documenti, sia quelli classificati come i più simili che quelli meno simili, mostra che il modello PV-DBOW può offrire prestazioni superiori o comunque paragonabili a quelle del modello PV-DM.

Di seguito è riportato un esempio utilizzando vettori di embedding di dimensione 50, una soglia di frequenza minima delle parole posta a 5 e una finestra di contesto di ampiezza 10.

Il modello PV-DBOW riconosce sé stesso come il documento più simile circa il 2% delle volte, mentre il modello PV-DM raggiunge il 10%. Si tratta comunque di basse percentuali che non aumentano significativamente al variare delle dimensioni del vettore di embedding. Tuttavia, le percentuali aumentano se si considera la frequenza con cui il documento appare tra i primi 10 risultati più simili.

Dal punto di vista semantico, viene utilizzato come test il ticket “cassetto cassa sostituire non aprire”.

In base al modello PV-DM, il documento più simile è “cassa bloccare non rispondere comandi” con una similarità di 0.95, mentre il documento meno simile è “riscontrare problema cassa dettaglio non avviare programma non emettere scontrino fiscale oasi lombardo carmelare” con una similarità di -0.79. Questo risultato mostra una difficoltà nel riconoscere come più simile un documento che si riferisce specificamente al malfunzionamento del cassetto della cassa, suggerendo che il modello potrebbe non sempre catturare il contesto semantico specifico.

Nel caso della variante PV-DBOW, il documento più simile è “cassetto cassa rompere chiave interno non aprire piu riferire molla cassetto dare problema richiedere sostituzione cassetto cassa ref imma pi” con una similarità di 0.97, mentre il documento meno simile è “cassa offlineref” con una similarità di -0.17.

Questi risultati suggeriscono che il modello PV-DBOW riesce a mantenere una buona coerenza semantica nei documenti simili e a distinguere quelli meno pertinenti.

Alla luce di queste osservazioni, il modello PV-DBOW si dimostra particolarmente efficace nel preservare la coerenza semantica e nel separare documenti simili da quelli dissimili, rendendolo una scelta vantaggiosa ai fini del clustering. Inoltre, considerando le prestazioni equiparabili e la maggior semplicità del modello PV-DBOW, si è deciso di proseguire l'analisi con quest'ultimo. La configurazione selezionata prevede una soglia minima di frequenza delle parole fissata a 5, per bilanciare adeguatamente rilevanza semantica ed efficienza operativa.

Il modello Doc2Vec individuato è stato successivamente testato per il clustering su un ampio intervallo di valori di cluster, in modo analogo a quanto fatto con la PCA. Le metriche di valutazione del clustering sono riportate nella Tabella 4 in modo da fornire un confronto tra le due tecniche.

Nello specifico, si presentano i risultati del clustering utilizzando vettori di embedding generati con Doc2Vec a 90 dimensioni e vettori ottenuti da USE dopo

l'applicazione della PCA con 90 componenti principali. Questo consente un confronto diretto al variare del numero di cluster.

L'obiettivo è valutare se Doc2Vec, con la sua maggiore flessibilità e capacità di personalizzazione, può offrire risultati superiori in termini di compattezza e separazione dei cluster.

	BCSS		WCSS		DB_index	
n_cluster	**USE + PCA**	**Doc2Vec**	**USE + PCA**	**Doc2Vec**	**USE + PCA**	**Doc2Vec**
2	2,34	0,55	53454,57	1138,71	2,93	1,55
12	3,09	1,80	7167,60	104,95	3,06	1,42
22	3,24	1,39	3561,61	46,98	2,86	1,55
32	3,33	1,22	2322,39	28,92	2,80	1,70
42	3,42	1,10	1699,32	20,45	2,72	1,78
52	3,48	1,05	1330,35	15,55	2,72	1,76
62	3,58	1,00	1085,65	12,50	2,72	1,80
72	3,52	0,98	915,02	10,36	2,75	1,82
82	3,60	1,00	787,81	8,81	2,69	1,83
92	3,62	0,97	690,21	7,63	2,78	1,86
100	3,63	0,98	627,90	6,88	2,77	1,86
150	3,65	0,92	393,95	4,16	2,70	1,90
200	3,69	0,89	282,27	2,92	2,68	1,91
250	3,79	0,89	217,04	2,21	2,59	1,88
300	3,88	0,85	176,74	1,77	2,59	1,89

Tabella 4: Confronto tra USE seguito da PCA e Doc2Vec

Nel caso dell'USE combinato con PCA, i valori di BCSS mostrano un incremento costante con l'aumentare del numero di cluster, con un notevole aumento fino a 12 cluster e una tendenza a crescere ulteriormente con l'aggiunta di nuovi gruppi. Diversamente, i valori di BCSS per Doc2Vec seguono un andamento più tipico: inizialmente, la distanza tra i centroidi aumenta rapidamente per poi stabilizzarsi o diminuire man mano che il numero di cluster cresce.

In entrambi i casi, si osserva un significativo decremento della WCSS nel passaggio da 2 a 12 cluster, con una continua diminuzione fino a 300 cluster. Tuttavia, i valori di WCSS ottenuti con Doc2Vec sono generalmente più bassi rispetto a quelli dell'USE con PCA. Questo suggerisce che Doc2Vec potrebbe essere più efficace nel generare rappresentazioni dei dati più dense e cluster più compatti.

Questa compattezza si riflette anche nei valori del Davies-Bouldin index. Anche se non vi sono variazioni particolarmente marcate, i valori del DB index per Doc2Vec risultano generalmente più bassi, indicando una migliore separazione tra i cluster e una maggiore omogeneità interna rispetto all'USE combinato con la PCA.

I risultati indicano che Doc2Vec potrebbe offrire prestazioni superiori rispetto all'USE combinato con PCA: Doc2Vec mostra valori significativamente più bassi sia per la WCSS che per il DBI, suggerendo cluster più ben definiti e separati. Nonostante l’Universal Sentence Encoder sia più semplice da implementare,

Doc2Vec si distingue per la sua maggiore flessibilità e, potenzialmente, migliori prestazioni di clustering, motivo per cui è stato scelto per continuare l'analisi.

3.6 Doc2vec: clustering

Dopo aver identificato Doc2Vec come la tecnica di Sentence Embedding più efficace, è stata intrapresa un'ulteriore esplorazione di un intervallo più specifico di configurazioni di clustering. I risultati, come riportato nella Tabella 4, avevano mostrato un miglioramento significativo di tutte le metriche considerate nel passaggio da 2 a 12 cluster, indipendentemente dalle dimensioni dei vettori di embedding utilizzati. Questo miglioramento suggerisce che, all'interno di questo intervallo, le configurazioni di clustering sono state maggiormente efficaci nel riflettere le strutture sottostanti dei dati.

Alla luce di queste osservazioni, l'analisi si è concentrata maggiormente su questo intervallo di valori, approfondendo come la scelta del numero di cluster possa influenzare le prestazioni del modello.

In particolare, i numeri di cluster che hanno mostrato valori più rilevanti nelle metriche di interesse sono stati 6 e 8. I risultati ottenuti con queste configurazioni vengono confrontati nella Tabella 5.

	BCSS		WCSS		DB Index	
Vector_size	**6 cluster**	**8 cluster**	**6 cluster**	**8 cluster**	**6 cluster**	**8 cluster**
vector_size_10	0,29	0,46	56,22	38,44	1,62	1,46
vector_size_15	0,32	0,61	78,34	52,01	1,68	1,42
vector_size_20	0,37	0,66	106,84	72,79	1,77	1,53
vector_size_25	0,37	0,69	118,51	79,90	1,72	1,46
vector_size_30	0,89	0,77	114,71	79,30	1,57	1,75
vector_size_50	1,13	0,98	171,69	115,06	1,56	1,48
vector_size_90	1,54	2,29	277,07	186,26	1,47	1,35
vector_size_130	1,85	2,82	316,88	212,33	1,50	1,32
vector_size_170	2,16	3,29	385,39	253,54	1,45	1,39
vector_size_210	4,66	3,72	392,27	264,08	1,21	1,33

Tabella 5: Confronto tra 6 e 8 cluster

Nel valutare la scelta, è importante considerare il bilanciamento tra le metriche di qualità del clustering e la complessità introdotta dall'aumento del numero di cluster.

Per quanto riguarda il clustering a 8 cluster, si osserva un trend crescente del BCSS all'aumentare della dimensione del vettore di embedding. Questo incremento suggerisce una migliore separazione tra i cluster rispetto a quelli ottenuti con la

configurazione a 6 cluster, segnalando che i gruppi sono più distinti tra loro. Contemporaneamente, la WCSS evidenzia una significativa riduzione con 8 cluster rispetto alla configurazione a 6. Questo implica che i punti all'interno di ciascun gruppo risultano più vicini tra loro, favorendo una maggiore compattezza dei cluster.

Un'ulteriore conferma di questo miglioramento nella qualità del clustering viene dai valori del DB Index, generalmente più bassi con 8 cluster. Questo riflette una migliore definizione e separazione tra i gruppi. Tuttavia, ci sono alcune eccezioni, come per le dimensioni vettoriali 30 e 210, dove i valori di DB Index per 8 cluster risultano superiori.

Nel complesso, la configurazione a 8 cluster risulta vantaggiosa sotto diversi aspetti: l'incremento del BCSS segnala una miglior separazione, la riduzione del WCSS riflette una maggiore compattezza interna e il DB Index, generalmente inferiore, indica una migliore definizione.

Si osserva, inoltre, che l'aumento del numero di cluster porta a una distribuzione dei dati tra i cluster meno omogenea. Questo fenomeno è evidenziato nella Tabella 6, che riporta le dimensioni dei cluster generati nelle diverse configurazioni.

	Points per cluster	
Vector_size	**6 cluster**	**8 cluster**
vector_size_10	[1755, 1749, 2899, 632, 908, 1465]	[1555, 17, 2796, 611, 730, 947, 941, 1811]
vector_size_15	[3701, 898, 1951, 814, 795, 1249]	[776, 1682, 784, 1082, 890, 13, 780, 3401]
vector_size_20	[1206, 3483, 682, 1368, 1222, 1447]	[13, 1313, 690, 1092, 1280, 667, 3119, 1234]
vector_size_25	[1197, 609, 1288, 2088, 697, 3529]	[613, 3477, 13, 1219, 1862, 820, 678, 726]
vector_size_30	[3709, 822, 697, 13, 1703, 2464]	[580, 1846, 723, 639, 818, 13, 1795, 2994]
vector_size_50	[626, 1322, 1406, 4069, 13, 1972]	[1635, 3640, 13, 675, 590, 661, 1412, 782]
vector_size_90	[4084, 825, 720, 13, 1987, 1779]	[1251, 789, 799, 2758, 3125, 13, 669, 4]
vector_size_130	[1687, 597, 13, 2520, 3872, 719]	[2412, 4, 524, 13, 3616, 648, 1604, 587]
vector_size_170	[13, 1668, 653, 3991, 599, 2484]	[1643, 4, 618, 13, 3308, 2478, 755, 589]
vector_size_210	[2446, 496, 4, 13, 2403, 4046]	[1070, 4, 613, 408, 1546, 3705, 2049, 13]

Tabella 6: Distribuzione dei dati tra i cluster

È evidente che l'aumento del numero di cluster, indipendentemente dalle dimensioni dei vettori di embedding, porta alla formazione di cluster molto piccoli e poco uniformi. Questi piccoli gruppi potrebbero non rappresentare cluster significativi, ma piuttosto potrebbero essere costituiti da ticket che rappresentano rumore o anomalie nel dataset. In particolare, i cluster di dimensioni molto ridotte,

come quelli con solo 13 o 4 elementi, tendono a riflettere dati che si discostano significativamente dal resto della distribuzione. Pertanto, questi cluster ridotti potrebbero essere considerati outlier naturali piuttosto che gruppi coerenti con caratteristiche distintive proprie.

Per approfondire la comprensione della struttura dei cluster e verificare la coerenza tematica al loro interno, è stato necessario esaminare l'omogeneità dei temi affrontati in ciascun gruppo. Questo processo ha coinvolto un'analisi dettagliata dei ticket più vicini ai centroidi di ogni cluster, ovvero di quei ticket che possono essere considerati rappresentativi dei temi dominanti all'interno di ciascun gruppo.

Durante questa fase di analisi, è emerso che l'utilizzo di vettori a 15 dimensioni si distingue per la sua capacità di garantire una migliore interpretabilità dei risultati rispetto ad altre configurazioni. I vettori di questa dimensione si sono rivelati sufficienti e ottimali, mostrando una maggiore coerenza tematica all'interno dei gruppi e facilitando la rilevazione di argomenti distinti e ben definiti. Permettono, infatti, di individuare con chiarezza i temi trattati, sia nel caso di una suddivisione in 6 cluster che in 8 cluster. Questa caratteristica è particolarmente utile per comprendere la distribuzione dei ticket e i problemi di supporto a cui fanno

riferimento. Per tale interpretabilità e chiarezza si ritiene la scelta più adatta per le considerazioni successive.

Dal confronto tra i temi rilevati mediante l'analisi a 8 cluster e quelli ottenuti con la suddivisione in 6 cluster emergono due differenze sostanziali.

La prima differenza consiste nella formazione di un cluster di dimensioni molto ridotte, mentre la seconda riguarda l'individuazione di un tema specifico che non era stato rilevato con un numero inferiore di gruppi: i problemi legati al funzionamento della stampante della cassa.

Approfondendo la prima differenza osservata, si notano cluster che contengono un numero molto limitato di dati, nel caso specifico 13. Un'analisi dettagliata di questi ticket isolati rivela che trattano temi eterogenei già identificati attraverso gli altri cluster. Inoltre, questi ticket condividono una caratteristica comune: presentano prevalentemente informazioni aziendali standard, inclusi codici interni per la gestione del ticket, allegati e una nota finale riguardante la riservatezza delle informazioni. Nella Tabella 7 si riporta il contenuto testuale di un ticket a titolo di esempio.

TICKETID	LDTEXT
77432	buongiorno il pos della cassa 4 non funziona esce fuori questa scritta e il pagamento non va saluti loreto 441 arianna carelli capo reparto casse ff oasi ipermercati filiale 441 via pizzardeto snc 60025 loreto an 071978371 0717501879 admin d41 gabriellispa it e6204ef7 fed64b1d etico d l gs 231 01 presente sul sito gruppo gabrielli il gruppo con la gente al centro il gruppo gabrielli e una testimonianza concreta di come sia possibile crescere con il territorio oggi e presente in 5 regioni con 3 diversi format distributivi rispondenti ad esigenze diversificate ai sensi del reg ue n 2016 679 e normativa vigente si precisa che il testo e gli eventuali documenti allegati trasmessi possono contenere documenti confidenziali e o materiale riservato al destinatario qui indicato la riservatezza della presente e mail e e proibita image0011662278930635 screenshot 2 saved as attachment image0021662278930642 screenshot 3 saved as attachment pos1662278930642

Tabella 7: Ticket di esempio

Il contenuto omogeneo dei dati suggerisce che il cluster di piccole dimensioni non rappresenta una problematica unica o significativa ma è il risultato di ticket dal contenuto standardizzato.

Non contribuendo in modo significativo alla segmentazione dei dati e all'individuazione di informazioni rilevanti, si ritiene opportuno escluderli dall'analisi.

La seconda differenza riguarda la capacità della configurazione a 8 cluster di individuare un tema che non emerge chiaramente con 6 cluster: i problemi legati al funzionamento della stampante della cassa.

Con 8 cluster, i dati vengono suddivisi in modo più granulare, il che consente di mettere in luce problematiche specifiche che, in una configurazione con 6 cluster,

potrebbero essere confuse con temi diversi a causa della maggiore aggregazione. Questa suddivisione più dettagliata permette di distinguere in modo più preciso tra problemi di natura differente, fornendo una visione più completa e migliorando così la capacità di intervento e risoluzione.

Capitolo 4

4.1 Analisi dei temi rilevati

Alla luce delle considerazioni del capitolo precedente, si è deciso di adottare il modello di Sentence Embedding Doc2Vec. L'analisi delle diverse configurazioni di clustering e la valutazione delle loro performance ha mostrato che suddividere i dati in 8 cluster genera alcuni gruppi di dimensioni molto ridotte, che possono essere considerati outlier a causa di standardizzazioni informative. Tuttavia, questa configurazione ha anche il merito di far emergere temi specifici, come i problemi legati alle stampanti delle casse, che altrimenti potrebbero non essere rilevati.

Pertanto, la scelta della configurazione con 8 cluster offre un vantaggio informativo, consentendo all'azienda di identificare con maggiore dettaglio le criticità operative.

Per valutare l'omogeneità tematica dei cluster, sono stati analizzati manualmente i ticket più vicini al centroide di ciascun gruppo. L'obiettivo era comprendere se i punti assegnati a ciascun cluster condividessero un argomento comune o presentassero caratteristiche simili dal punto di vista tematico. Questo approccio

ha permesso di valutare quanto efficacemente il modello di clustering fosse in grado di catturare e raggruppare i ticket.

Di seguito sono elencate le aree problematiche riscontrate, insieme al numero di ticket associati a ciascuna area e ai testi originali ('LDTEXT') dei ticket più vicini al centroide di ciascun cluster:

- Problemi di apertura del cassetto della cassa: 776 ticket

TICKETID	LDTEXT	Centroid_distance
103905	buongiorno cassetto cassa n 2 non si apre in automatico saluti d37	0,0594
58325	cassetto cassa 3 non si apriva piu la cassiera ha rotto la chiave all interno della serratura ora ovviamente e inutilizzabile saluti	0,0697
85565	la cassa 8 non funziona l apertura cassetto	0,0734
107859	buonasera si richiede intervento per la cassa 1 non si apre il cassetto della cassa grazie francesca	0,0740
70183	buongiorno sono rotte le molle del cassetto della cassa 7 non si apre antonella	0,0741

I ticket segnalano problemi con i cassetti delle casse, tra cui difficoltà di apertura automatica, malfunzionamenti dovuti a chiavi rotte, molle danneggiate, e necessità di assistenza manuale. Il cluster risulta piuttosto omogeneo semanticamente, con i ticket più distanti dal centroide che continuano a riferirsi a malfunzionamenti del cassetto.

- Problemi con le pistole scanner: 1682 ticket

TICKETID	LDTEXT	Centroid_distance
68957	buongiorno pistola scanner cassa 7 tasto rotto saluti fil 538 ancona	0,0596
104410	la pistola hand scanner della cassa self n 109 non funziona	0,0649
105741	buongiorno la pistola scanner della cassa 2 non funziona fil 538 ancona	0,0682
100740	si richiede intervento per il cavo scanner della cassa 5 danneggiato con piega e scotc grazie fild38 castelfidardo donati	0,0702
91768	nella cassa 4 non funziona la pistola saluti fil 538 ancona	0,0706

I ticket trattano problemi legati alle pistole scanner delle casse nei vari punti vendita. I problemi segnalati includono scanner non funzionanti, tasti rotti e cavi danneggiati. In generale, si richiedono interventi tecnici per ripristinare il corretto funzionamento degli scanner nelle casse. Anche in questo cluster, i ticket più distanti dai centroidi sono coerenti riferendo problemi di funzionamento delle pistole scanner.

- Problemi con i pagamenti elettronici e in contanti: 784 ticket

TICKETID	LDTEXT	Centroid_distance
56554	cassa bloccata dopo pagamento elettronico	0,0676
110444	cassa 6 pagamento in corso 18 53 pag contati cliente in cassa rem maurizio 0699320043	0,0685
101501	cassa 2 bloccata su dhcp non abilitato dopo errore pagamento contanti 4 96	0,0709

103328	cassa 2 non e stato emesso scontrino con spesa 11 11 pagato con bancomat cliente in cassa katiuscia 06 9141559 cp	0,0730
91786	cassa 1 bloccata su pos abends durante pagamento contanti sa	0,0741

I ticket segnalano che le casse si bloccano spesso durante i pagamenti, sia elettronici che in contante. Inoltre, ci sono problemi con l'emissione degli scontrini e difficoltà di comunicazione tra il sistema di pagamento e i dispositivi associati. Molti tra i ticket più distanti dai centroidi riguardo malfunzionamenti dei POS e blocchi durante le operazioni di pagamento, mentre alcuni trattano problemi legati alla gestione dei premi del catalogo e alle operazioni con buoni pasto.

- Blocchi su schermate specifiche: 1082 ticket

TICKETID	LDTEXT	Centroid_distance
85596	cassa 1 bloccata sulle vendite utente riavvia cassa ma il problema persiste nessun mess d errore ref liliana 06 85305446	0,0875
84994	cassa 2 in schermata desktop ref_daniela 0864201081 rp	0,0878
87534	cassa 1 bloccatabloccata in schermata desktop rem_pasquale rp	0,0892
84491	cassa 2 bloccato in schermata desktop rem_fabrizio rp	0,0907
91357	cassa 1 bloccata in schermata desktop rem_pietro rp	0,0908

I ticket segnalano che diverse casse si bloccano frequentemente durante il loro utilizzo. I problemi includono schermi neri, schermate di avvio bloccate, e blocchi generali sulla schermata desktop. Alcuni ticket indicano anche che le casse non emettono messaggi di errore specifici e che il problema persiste anche dopo i tentativi di riavvio. Analizzando i ticket meno rappresentativi del cluster emergono coerenti problemi di blocco delle casse, ma anche nuovi aspetti come malfunzionamenti del touch screen, difficoltà con i codici operatore e problemi specifici legati alla bilancia della cassa.

- Problemi con la stampante scontrini della cassa: 890 ticket

TICKETID	LDTEXT	Centroid_distance
55892	verifica stampante cassa 5 attachment 20210419081028608161881335731 7 pdf	0,0739
69021	la scontrino stampatop dalla stampante della cassa n 6 non si vede	0,0748
103760	buongiorno la stampante della cassa n 1 stampa male sovrappone la scritta saluti valentina	0,0757
103763	buongiorno la stampante della cassa n 1 stampa sovrapponendo le scritte saluti valentina	0,0770
104723	buongiorno la cassa 7 stampa male allego scansione in esempio saluti antonella attachment 20231223095159314170332237076 8 pdf	0,0773

I ticket segnalano vari problemi con le stampanti delle casse: alcune stampanti producono scontrini con caratteri sovrapposti, poco leggibili o sfocati, mentre altre hanno problemi meccanici come il taglio non corretto degli scontrini o rumori anomali durante la stampa. Il cluster risulta piuttosto coerente ed omogeno riferendosi il più delle volte agli stessi temi.

- Problemi relativi a chiusure pendenti: 780 ticket

TICKETID	LDTEXT	Centroid_distance
109545	cassa 2 una chiusura pendente romina 0736 845143	0,0521
61865	cassa 1 chiusure pendenti 9tonia	0,0556
63871	chiusure pendenti 0736349974 domenico	0,0567
110472	4 chiusure pendenti su cassa 2 085 4683000 barabara cs	0,0586
63066	2 chiusure pendenti per tutte le casse barbara 074344801pp	0,0590

I ticket segnalano principalmente problemi legati alle chiusure pendenti delle casse, indicando difficoltà nel completare le procedure di chiusura su diverse postazioni. I ticket più distanti dai centroidi trattano non solo di chiusure pendenti, ma anche di malfunzionamenti degli scanner, problemi con i cassetti e difficoltà nel passaggio di codici a barre, mostrando una minore omogeneità tematica.

- Blocchi delle casse: 3401 ticket

TICKETID	LDTEXT	Centroid_distance
90872	cassa 3 in telematic trasmission ref di pietro	0,0332
63433	solo 2 casseref federica	0,0408
60741	cassa1 bloccata solo due casse caduta linea in chiamta 0734 658068 pp	0,0409
58130	cassa 2 bloccata su pos abendspdv ha solo una cassa operativa sig lorenzo cb	0,0412
93667	cassa 2 bloccata in caricam durante trans ref valeria	0,0416

I ticket segnalano principalmente problemi di blocco delle casse in diverse situazioni, come durante la trasmissione telematica o l'accensione. Questo cluster è il più numeroso e mostra una notevole varietà. I ticket più vicini ai centroidi riguardano principalmente il blocco generico delle casse, mentre quelli più distanti trattano una gamma più ampia di problemi, tra cui malfunzionamenti delle stampanti, difficoltà con gli scanner di codici a barre, problemi con le gift card. Questa varietà nella tipologia dei problemi riscontrati indica una maggiore disomogeneità rispetto agli altri gruppi individuati.

4.2 Analisi della distanza di modifica

Una volta suddivisi i ticket in cluster, si procede con un'analisi approfondita. L'obiettivo in questa fase è esaminare la relazione tra la distanza sintattica tra i ticket appartenenti ad un determinato cluster e la loro vicinanza al rispettivo centroide. Questo permette di valutare se il modello di clustering utilizzato stia effettivamente catturando le somiglianze semantiche o se si stia basando esclusivamente su corrispondenze sintattiche tra i ticket.

In particolare, vogliamo verificare se esiste una correlazione tra la vicinanza di un ticket al centroide del proprio cluster e la sua differenza sintattica rispetto al ticket più vicino a quel centroide. L'ipotesi alla base è che, in un modello di sentence embedding efficace, la distanza tra i ticket non dovrebbe essere guidata solo dalle differenze sintattiche, ma dovrebbe riflettere una somiglianza semantica più profonda.

Se osservassimo una correlazione chiara tra la distanza sintattica e la distanza dal centroide - se i ticket più vicini al centroide avessero una bassa differenza sintattica e i più lontani una differenza sintattica elevata - ciò indicherebbe che il clustering si basa principalmente su somiglianze sintattiche. In tal caso, il modello non sarebbe riuscito a catturare la dimensione semantica, dimostrando un limite nell'efficacia del sentence embedding scelto.

Al contrario, se non emergesse alcuna correlazione chiara, potremmo affermare che il sentence embedding ha funzionato come previsto, catturando le somiglianze semantiche tra i ticket in modo indipendente dalle loro somiglianze a livello di struttura sintattica. In questo contesto, dimostreremmo che il modello Doc2Vec, è non solo efficace ma anche superiore rispetto a un approccio basato su word embedding, come Word2Vec.

La ragione di questa affermazione risiede nella capacità del sentence embedding, in particolare di modelli come Doc2Vec, di catturare il contesto globale delle frasi. A differenza dei modelli di word embedding, che rappresentano ciascuna parola in isolamento e che sono quindi limitati a catturare relazioni tra singole parole, Doc2Vec tiene conto dell'intero contesto della frase. Questo significa che, anche se due frasi contengono parole diverse, un modello di sentence embedding può riconoscerle come semanticamente simili.

In questa analisi, la differenza sintattica viene misurata utilizzando la distanza di Levenshtein, implementata tramite pylev. Questa distanza quantifica il numero minimo di operazioni — come inserimenti, cancellazioni o sostituzioni di parole — necessarie per trasformare una stringa in un'altra. Nel caso specifico, si confrontano i ticket lemmatizzati, il che dovrebbe ridurre naturalmente le distanze tra di essi, minimizzando le variazioni legate alla morfologia delle parole.

Di seguito viene presentato un esempio relativo al cluster 0, associato a problemi durante le operazioni di pagamento. Per offrire una visualizzazione chiara, sono stati selezionati e ordinati i primi 20 ticket in base alla distanza vettoriale dal ticket più vicino al centroide. Inoltre, per ogni ticket è riportata anche la distanza di Levenshtein sempre riferita al ticket più rappresentativo del cluster, così come mostrato nella Figura 3.

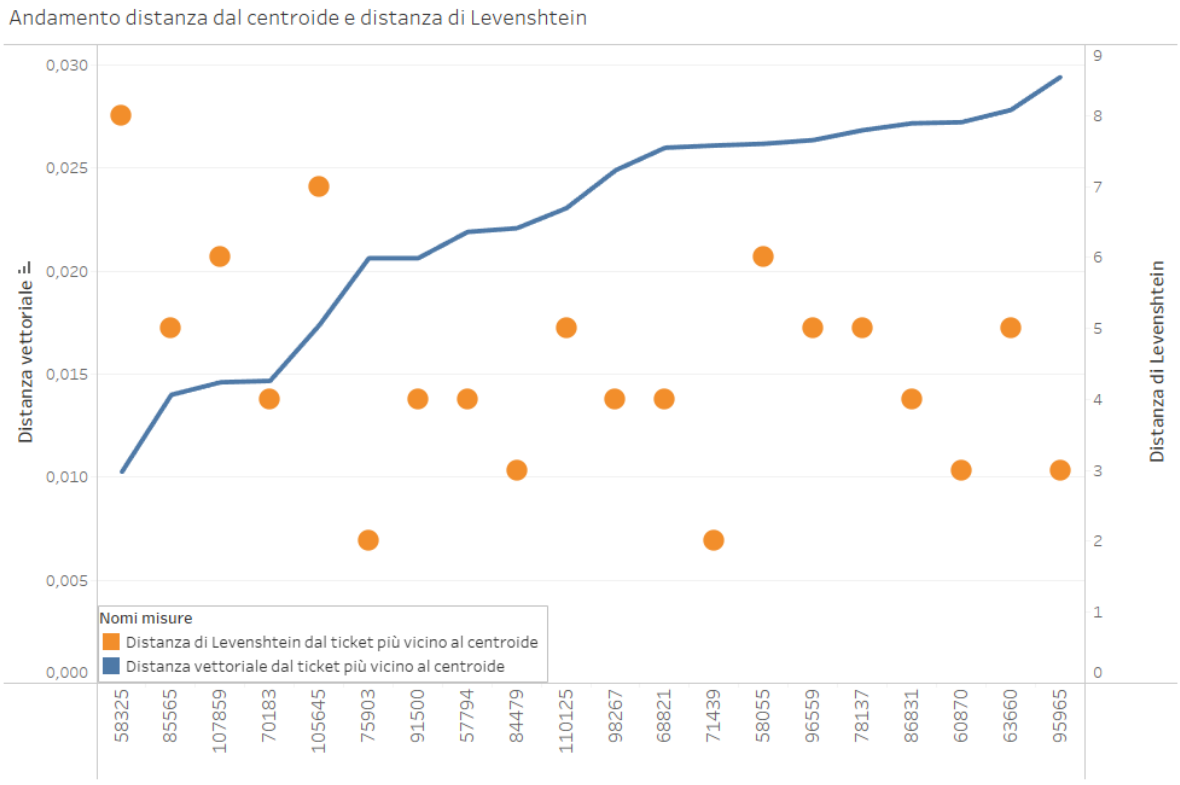

Figura 3: Distanza vettoriale e distanza di Levenshtein nel cluster 0

Dall'analisi emerge un andamento variabile: i ticket più vicini al centroide non presentano distanze vettoriali più basse rispetto a quelli più distanti. In generale, la variabilità delle distanze di Levenshtein suggerisce che il clustering non ha raggruppato i ticket solo in base alla presenza di determinate parole, ma piuttosto ha catturato la somiglianza semantica.

Ad esempio, il secondo ticket più vicino al centroide ha una distanza di modifica di 8 rispetto al primo, ma la distanza vettoriale tra i due è di solo 0,01. Di seguito vengono confrontati i testi lemmatizzati dei due ticket considerati:

- ID 103905: “cassetto cassa n non aprire automatico”
- ID 58325: “cassetto cassa non aprire cassiera rompere chiave interno serratura ovviamente inutilizzabile”

Nonostante la distanza di Levenshtein sia relativamente elevata, i ticket sono semanticamente molto vicini, poiché descrivono lo stesso problema: un malfunzionamento nell'apertura del cassetto della cassa. Questo supporta l’idea che il clustering si sia basato su somiglianze semantiche piuttosto che sintattiche.

Un altro aspetto rilevante riguarda l’impatto della lunghezza dei ticket sulla distanza di Levenshtein. Ticket più lunghi tendono ad avere una distanza di Levenshtein più alta rispetto a ticket più brevi, semplicemente a causa della maggior presenza di parole. Tuttavia, il clustering, che si è basato sui sentence

embeddings, è riuscito a raggruppare correttamente anche ticket di lunghezza molto diversa, purché riferiti ad argomenti semanticamente simili.

Ad esempio, nel cluster associato a problemi durante i pagamenti, osserviamo ticket di lunghezza variabile, tra cui gli esempi riportati nella Tabella 8:

TICKETID	LDTEXT	Testo pulito	Centroid_distance
56554	cassa bloccata dopo pagamento elettronico	cassa bloccare pagamento elettronico	0,0676
103337	buongiorno abbiamo la cassa 7 bloccata dopo la transazione con pagamento elettronico n 64 delle 11 42 e uscito solo meta dello scontrino e si e bloccata nella ristampa che abbiamo mandato abbiamo riavviato ma non va oltre la schermata allegata invio anche foto del display saluti valeria fil 564 img 20231202 wa00021701517987042 screenshot 2 saved as attachment img 20231202 wa00041701517987058	cassa bloccare transazione pagamento elettronico n uscire meta scontrino bloccare ristampa mandare riavviare non schermato allegare invio foto display valeria fil img wa screenshot saved attachment img wa	0,413

Tabella 8: Ticket di esempio con rispettiva distanza dal centroide del cluster

Nonostante la distanza di Levenshtein tra questi ticket lemmatizzati sia elevata, nello specifico è pari a 32, entrambi descrivono lo stesso problema: una cassa bloccata dopo una transazione elettronica. Questa similitudine semantica ha portato i ticket a essere raggruppati correttamente nello stesso cluster.

I dati raccolti dimostrano, quindi, che non vi è una relazione evidente tra la distanza sintattica e la vicinanza al centroide all'interno del gruppo. Questo indica che il clustering è riuscito a cogliere le problematiche indipendentemente dalla struttura sintattica delle frasi.

Questo risultato conferma l'efficacia del sentence embedding e, in particolare del modello Doc2Vec, rispetto ai modelli di word embedding come Word2Vec. Doc2Vec, infatti, è in grado di catturare il contesto globale delle frasi in modo più accurato, rendendolo un valido strumento per rappresentare e raggruppare correttamente il significato semantico dei documenti.

4.3 Analisi della modalità di risoluzione

Tra le informazioni disponibili sui ticket di supporto, la categoria "Third Level" identifica la modalità di risoluzione adottata per ciascun ticket. Concentrarsi su questa categoria offre una prospettiva importante su come le problematiche rilevate vengono affrontate e risolte, permettendo di ottimizzare le strategie di supporto.

La modalità di risoluzione, infatti, può fornire un quadro sulla distribuzione dei ticket tra i cluster. Da questa prospettiva, analizzare questa informazione può essere un modo per valutare l'efficacia del clustering stesso: ciò permette di

comprendere se il clustering ha effettivamente separato in modo accurato i ticket in base a caratteristiche rilevanti, come il tipo di problema e la soluzione adottata. Ad esempio, la predominanza all'interno di un cluster di una specifica modalità di risoluzione potrebbe suggerire che il cluster stesso è altamente rappresentativo di quella tipologia di problema o soluzione. D'altro canto, un'eterogeneità in termini di risoluzione potrebbe riflettere la diversità dei problemi trattati e/o delle soluzioni applicate nel cluster.

Tale approfondimento, inoltre, diventa uno strumento utile per identificare pattern ricorrenti nelle soluzioni adottate, offrendo un'opportunità per affinare ulteriormente le strategie di supporto. L'idea principale è quella di utilizzare la modalità di risoluzione per capire meglio i contenuti di ciascun cluster, associando ciascun gruppo a una specifica tipologia di problema e alla relativa soluzione.

Per effettuare una valutazione completa, è importante osservare come si distribuiscono le modalità di risoluzione dei ticket nel dataset complessiva. La distribuzione della variabile è mostrata in Figura 4.

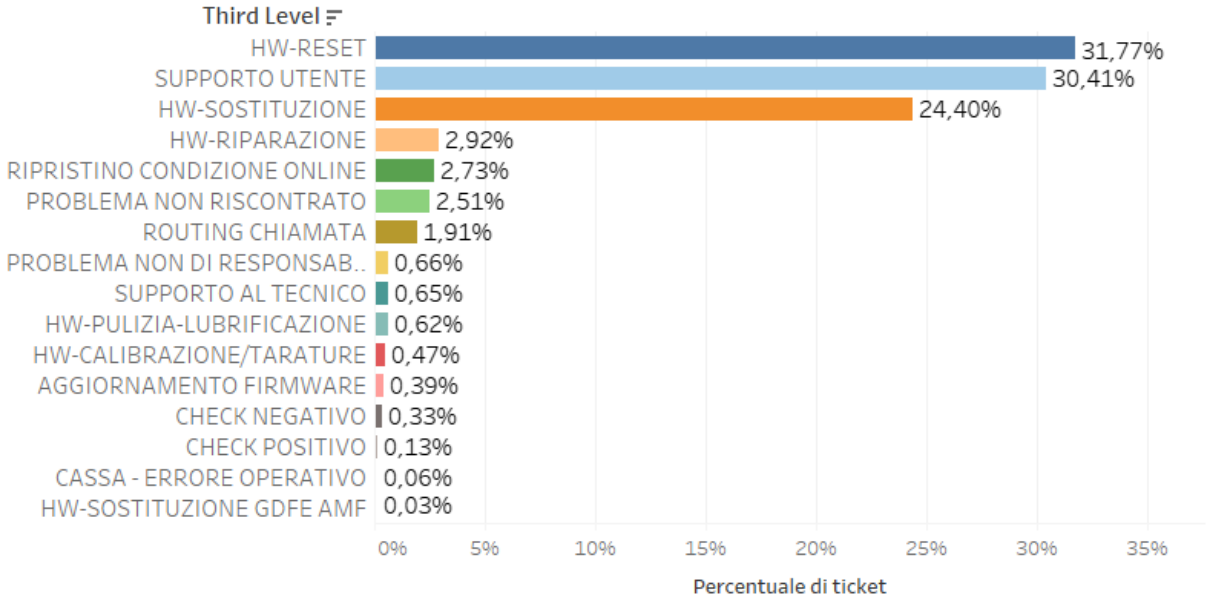

Figura 4: Distribuzione delle modalità di risoluzione

Dal grafico della distribuzione di frequenza emerge chiaramente che su un totale di 16 modalità di risoluzione, ben l'86,6% dei ticket è stato gestito attraverso solo tre categorie principali: reset, supporto utente e sostituzione. Questo indica una forte predominanza di queste modalità nella risoluzione dei ticket. Le restanti 13 modalità coprono solo il 13,4% del totale. Questa distribuzione fortemente sbilanciata suggerisce che la maggior parte dei ticket viene risolta attraverso poche soluzioni principali, mentre le altre categorie di risoluzione sono molto meno frequenti.

Tale sbilanciamento nella distribuzione delle modalità di risoluzione non solo caratterizza il dataset complessivo, ma si riflette inevitabilmente anche nella distribuzione delle soluzioni all'interno dei cluster generati. Ogni cluster potrebbe quindi mostrare una distribuzione delle modalità di risoluzione che replica o accentua questa disuguaglianza.

Per analizzare e valutare la concentrazione delle modalità di risoluzione all'interno di ciascun cluster, utilizziamo tre metriche chiave relative a ciascuna modalità di risoluzione:

- **Precision:** Questa metrica indica quali modalità sono le più frequenti all'interno di ciascun cluster. Un valore elevato di precision suggerisce che il cluster è particolarmente rappresentativo di una determinata modalità di risoluzione e che il tipo di problema associato è frequentemente trattato con quella soluzione. Nello specifico, la precision è calcolata come segue:

$$Precision_{ij} = \frac{N_{ij}}{N_j}$$

dove:

N_{ij} indica il numero di ticket risolti con la modalità i nel cluster j

N_j indica il numero totale di ticket del cluster j

- **Recall:** Questa metrica è utile a comprendere come le diverse modalità di risoluzione si distribuiscono tra i cluster identificati. Un valore elevato di recall per una soluzione in un cluster indica che una grande porzione dei ticket risolti con quella modalità è concentrata in quel cluster specifico. Al contrario, valori bassi di recall indicano che la modalità di risoluzione in questione è distribuita in maniera più equa tra diversi cluster, o che il cluster considerato non contiene una rappresentazione significativa di ticket risolti con quella modalità. Nello specifico, la misura è calcolata come segue:

$$Recall_{ij} = \frac{N_{ij}}{N}$$

dove:

N_{ij} indica il numero di ticket risolti con la modalità i nel cluster j

N indica il numero totale di ticket nel dataset

- **F1 score:** Questa misura combina le metriche precedenti per fornire una valutazione complessiva delle performance. È progettata per riflettere il valore minimo tra precision e recall, penalizzando scarsi risultati in una delle due metriche. Di conseguenza, l'F1 Score offre un'importante indicazione di quanto bene un cluster rappresenti una modalità di

risoluzione, integrando entrambi gli aspetti critici della performance. Un valore di F1 score pari o vicino a 1 indica una performance ottimale, segnalando risultati eccellenti sia in termini di precision che di recall. La misura è calcolata come la media armonica tra precision e recall, nello specifico:

$$F1\_score_{ij} = \frac{2 * Precision_{ij} * Recall_{ij}}{Precision_{ij} + Recall_{ij}}$$

Per ogni cluster generato, è stato inoltre calcolato il weighted F1-score, un indicatore aggregato che valuta la qualità della rappresentazione delle modalità di risoluzione dei ticket all'interno di ciascun cluster. La misura è calcolata come segue:

$$Weighted\ F1\_score_j = \sum_{i=1}^{I} w_i * F1\_score_i$$

Dove:

I è il numero di modalità di risoluzione presenti nel cluster j

$F1\_score_i$ è l'F1 score della modalità i nel cluster j

w_i è il peso associata a ciascuna modalità i, nello specifico è dato dalla frequenza relativa della modalità considerata rispetto all'intero dataset.

I punteggi del weighted F1-score, riportati nella Tabella 9, variano principalmente tra 0,12 e 0,18. Il punteggio più alto, pari a 0,29, è riferito al cluster 7, che è il gruppo che contiene il maggior numero di punti.

Cluster	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Weighted F1-score	0,107	0,187	0,122	0,150	0,125	0,003	0,125	0,291

Tabella 9: Weighted F1 score dei cluster

Nonostante ciò, la maggior parte dei cluster presenta dei valori piuttosto bassi, suggerendo una scarsa qualità nella rappresentazione delle modalità di risoluzione all'interno dei cluster. Questo indica che le modalità di risoluzione potrebbero essere distribuite in modo uniforme tra i cluster, piuttosto che concentrate in gruppi specifici di ticket. Di conseguenza, non si nota una chiara aggregazione delle soluzioni per problemi simili, il che sottolinea come la distribuzione sbilanciata delle modalità di risoluzione nel dataset influenzi significativamente i risultati del clustering.

Per ottenere una comprensione più approfondita, è utile esaminare la distribuzione delle modalità di risoluzione più frequenti nel dataset. Questa analisi consente di capire come tali soluzioni siano ripartite tra i cluster e di identificare eventualmente quelli che mostrano una predominanza significativa.

Dall'analisi emerge che HW-RESET è la modalità di risoluzione più frequente nei cluster 7 e 3, con dei valori di precision rispettivamente pari al 41,6% e al 45,47%. I valori di riferimento sono riportati nella Tabella 10. Questi cluster sono stati associati a problemi di blocco delle casse, sia in termini generici che in relazione a specifiche schermate. La frequenza con cui viene utilizzato il reset suggerisce una correlazione tra questa soluzione e i problemi riscontrati, rendendo tali gruppi rappresentativi di questa modalità di risoluzione.

HW-RESET			
cluster	**precision**	**recall**	**f1_score**
7	0,4161	0,4734	0,4429
3	0,4547	0,1646	0,2417
2	0,4031	0,1057	0,1675
1	0,1891	0,1064	0,1362
6	0,2782	0,0726	0,1151
4	0,1607	0,0478	0,0737
0	0,1095	0,0284	0,0452
5	0,2308	0,0010	0,0020

Tabella 10: Distribuzione della modalità di risoluzione HW-RESET

In particolare, il cluster 7 mostra l'F1 score più elevato, indicando una buona combinazione di precision e recall. Questo significa che non solo un gran numero di ticket in questo cluster è stato risolto tramite reset, ma anche che quasi il 50% dei ticket risolti con reset si concentra in questo gruppo.

Esaminando invece la modalità di risoluzione HW-SUPPORTO UTENTE, osserviamo che i punteggi di F1 score più elevati si riscontrano nei cluster 7 e 6, così come mostrato nella Tabella 11.

SUPPORTO UTENTE			
cluster	**precision**	**recall**	**f1_score**
7	0,3176	0,3775	0,3449
6	0,5987	0,1632	0,2565
3	0,3420	0,1293	0,1877
2	0,3750	0,1028	0,1613
1	0,1908	0,1122	0,1413
4	0,2438	0,0758	0,1157
0	0,1430	0,0388	0,0610
5	0,0769	0,0003	0,0007

Tabella 11: Distribuzione della modalità di risoluzione SUPPORTO UTENTE

Nel cluster 7, il punteggio pari a 0,34 deriva da un buon equilibrio tra precision e recall. Nel cluster 6, invece, il valore pari a 0,26 è fortemente influenzato da una bassa recall. Questo è coerente con le osservazioni precedenti riguardo alla

frequenza di utilizzo di questa modalità di risoluzione. Nonostante questo, è evidente che la modalità in analisi è predominante nel cluster 6: quasi il 60% dei ticket associati a chiusure pendenti sono stati risolti ricorrendo al supporto utente.

Valutando, infine, la distribuzione della modalità HW-SOSTITUZIONE, riportata nella Tabella 12, si osserva che i cluster 0, 1 e 4 presentano i valori più elevati di F1 score, indicando una maggior concentrazione in tali gruppi.

HW-SOSTITUZIONE			
cluster	**precision**	**recall**	**f1_score**
1	0,4602	0,3371	0,3891
0	0,5490	0,1855	0,2773
4	0,4584	0,1777	0,2561
7	0,1382	0,2047	0,1650
2	0,1212	0,0414	0,0617
3	0,0767	0,0361	0,0491
6	0,0410	0,0139	0,0208
5	0,6154	0,0035	0,0069

Tabella 12: Distribuzione della modalità di risoluzione HW-SOSTITUZIONE

L'analisi del contenuto testuale aveva precedentemente associato questi cluster a problemi specifici quali l'apertura del cassetto delle casse, malfunzionamenti delle pistole scanner e della stampante degli scontrini. Data la natura di questi problemi, legati a componenti esterni, risulta plausibile che la sostituzione sia stata utilizzata come soluzione in modo prevalente. Si nota, inoltre, un bilanciamento tra

precision e recall per il cluster 1, mentre il cluster 0 è caratterizzato da una precision più alta. In particolare il 55% dei ticket di quest'ultimo gruppo sono stati risolti tramite sostituzione.

A questo punto, la prospettiva di analisi cambia, focalizzandosi sulla distribuzione delle modalità di risoluzione all'interno di un cluster specifico.

Analizzando i dati relativi al cluster 1, possiamo fare alcune considerazioni riguardo le modalità di risoluzione che sono più frequentemente associate, sulla base dei risultati mostrati in Figura 5.

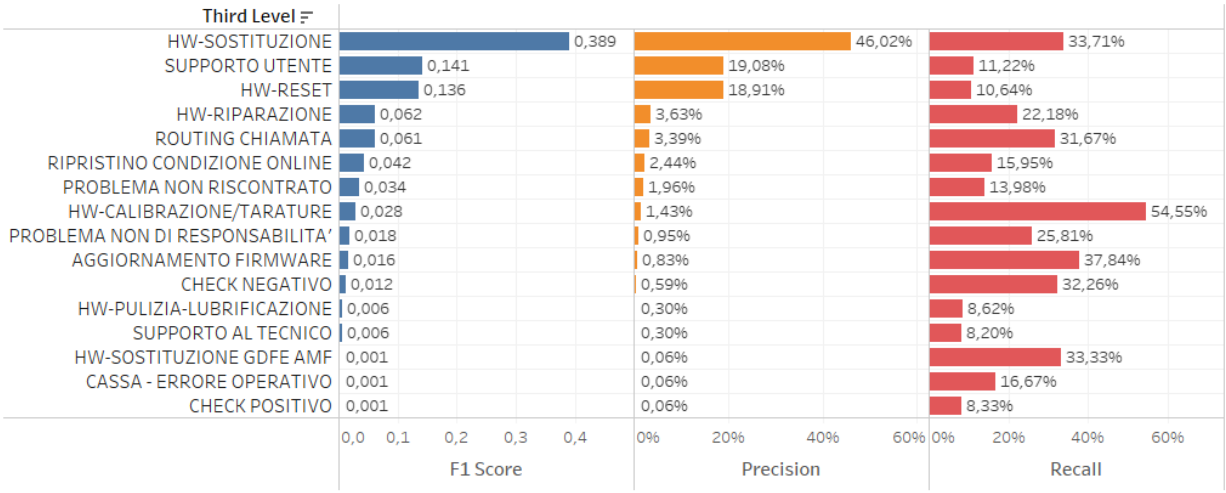

Figura 5: Distribuzione del "Third Level" nel cluster 1

Considerando che i ticket più vicini riguardano malfunzionamenti delle pistole scanner, è possibile identificare quali modalità di risoluzione risultano più coerenti con questo tipo di problematica.

La modalità HW-SOSTITUZIONE mostra i valori più alti sia in termini di precision che in generale in termini di F1 score. Ciò significa che una parte consistente dei ticket presenti nel cluster 1 viene risolta tramite la sostituzione dell'hardware, il che sembra coerente con la descrizione dei ticket. La precision relativamente alta (0,46) indica che quasi la metà dei ticket contenuti nel cluster sono stati risolti tramite sostituzione, mentre la recall (0,337) suggerisce che solo una parte di tutti i ticket risolti con questa modalità è concentrata nel cluster. Ciò è coerente con la distribuzione sbilanciata del dataset presentata in precedenza.

Le altre due modalità di risoluzione più frequenti, ovvero HW-SUPPORTO UTENTE e HW-RESET, mostrano valori simili per tutte le metriche e, insieme, costituiscono circa il 38% delle soluzioni dei ticket contenuti nel cluster. Inoltre, le recall indicano che solo una piccola parte dei ticket risolti con queste modalità è contenuta nel cluster 1. Questi potrebbero rappresentare approcci secondari alla risoluzione di problemi legati al malfunzionamento dello scanner, come tentativi di ripristino o assistenza utente.

I risultati relativi alle modalità HW-CALIBRAZIONE/TARATURE e AGGIORNAMENTO FIRMWARE offrono spunti interessanti: nonostante valori di precision molto bassi, le recall risultano relativamente alte. In particolare, notiamo che più del 54% dei ticket risolti tramite calibrazione o taratura è stato incluso nel cluster 1. Questo risultato è coerente considerando la tematica centrale del cluster. In generale, pur essendo pochi i ticket risolti con queste modalità all'interno del cluster, una buona parte del totale complessivo è concentrata in questo gruppo.

Una situazione simile si verifica, seppur in maniera più contenuta, nel cluster 4. Si nota che la modalità di risoluzione HW-PULIZIA/LUBRIFICAZIONE presenta la recall più alta tra tutte le soluzioni (34,48%), indicando che una parte significativa dei ticket risolti con questa modalità si trova nel cluster. Ciò è coerente con problemi di manutenzione delle stampanti scontrini, suggerendo che la pulizia e la lubrificazione sono soluzioni frequentemente adottate per tali malfunzionamenti. Tuttavia, anche qui il basso valore di Precision (2,2%) e quindi dell'F1 Score (0,042) segnalano che questa soluzione rappresenta marginalmente il cluster, indicando che, pur essendo rilevante, non è la soluzione più frequente.

Da queste analisi si nota che lo sbilanciamento nella distribuzione complessiva delle modalità di risoluzione nel dataset si riflette anche all'interno dei cluster

attraverso valori di precision bassi riferiti alla maggior parte delle modalità di risoluzione. L'integrazione con altre metriche, come la recall, ha comunque permesso di ottenere informazioni utili e di approfondire la comprensione dell'associazione tra i cluster e le problematiche affrontate.

Capitolo 5

Conclusioni e sviluppi futuri

L’obiettivo del presente lavoro era analizzare testualmente i ticket di supporto di Magazzini Gabrielli SpA per raggrupparli in base ai temi trattati e identificare le principali aree problematiche, con l’intendo di etichettare ciascun ticket in base al problema specifico descritto. Considerando l’elevata quantità di dati generati e le sfide associate all'analisi manuale, è stato cruciale sviluppare e applicare tecniche di analisi automatica per gestire e processare efficacemente i dati.

Per raggiungere questi obiettivi, sono state esplorate e confrontate diverse tecniche di Sentence Embedding per rappresentare semanticamente i ticket in modo efficace. Queste tecniche sono state valutate in base alle loro performance di clustering.

Una volta selezionata la tecnica di embedding ottimale, sono stati applicati nuovamente algoritmi di clustering per raggruppare i ticket in base alla loro similarità semantica. Successivamente, è stata condotta un'analisi dettagliata di ciascun gruppo per verificare se i ticket fossero stati correttamente raggruppati in base a tematiche simili. Questa fase ha portato all'identificazione di temi ricorrenti e problematiche predominanti all'interno di ogni cluster. L’analisi ha permesso

infatti di etichettare i ticket in linea con le problematiche specifiche che descrivono, offrendo così una visione chiara e strutturata delle aree di intervento prioritario.

L'approccio che ha portato ai risultati migliori ha previsto l'applicazione dell'algoritmo k-means con k = 8, utilizzando vettori di embedding di dimensione 15. Questa configurazione ha permesso di ottenere buoni risultati nelle metriche considerate: Between-cluster sum of squares (BCSS) pari a 0,61, Within-cluster sum of squares (WCSS) pari a 52,01 e un Davies-Boudin Index di 1,42. Analizzando la coerenza tematica all'interno dei cluster, è stato possibile associare un problema specifico a ciascun gruppo, facilitando così l'individuazione delle problematiche ricorrenti all'interno della categoria analizzata.

Alla luce dei risultati ottenuti, si può affermare che le tecniche di sentence embedding e di clustering siano state in grado di identificare efficacemente le problematiche principali, offrendo un quadro delle aree di intervento prioritario. Nonostante ciò, le metriche ottenute indicano che ci sono margini di miglioramento. È possibile affinare ulteriormente le tecniche per migliorare la coerenza all'interno dei cluster, identificare aree critiche che potrebbero non essere state rilevate in questo lavoro e perfezionare così l'analisi dei ticket.

Un importante sviluppo futuro sarebbe l’estensione dell'analisi ad altre categorie di ticket presenti nel dataset. Finora, il lavoro si è concentrato su un sottoinsieme specifico, ma includere ulteriori categorie potrebbe offrire una visione più completa delle problematiche aziendali. Inoltre, sperimentare con algoritmi di clustering e tecniche di sentence embedding alternativi potrebbe affinare la precisione dell'analisi. Allo stesso modo, l'adozione di Large Language Models (LLMs) potrebbe ulteriormente aumentare l'efficienza e l'efficacia nella gestione dei ticket.

Un ulteriore miglioramento potrebbe derivare dall'aumento del numero di annotatori coinvolti nella classificazione dei ticket. Incrementando il numero di annotatori e implementando un sistema di majority voting, si potrebbero ottenere classificazioni più affidabili e precise. Questo approccio rafforzerebbe l'efficacia della classificazione automatica dei ticket.

Questi sviluppi potrebbero rendere più efficace l'analisi dei ticket e migliorare le soluzioni proposte per Magazzini Gabrielli, portando a una gestione più strategica delle problematiche operative e un miglioramento della soddisfazione del cliente.

Bibliografia

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. In Proceedings of the International Conference on Learning Representations (ICLR).

https://arxiv.org/abs/1301.3781

Le, Q., & Mikolov, T. (2014). *Distributed representations of sentences and documents.*

https://arxiv.org/abs/1405.4053

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. st., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2018). *Universal Sentence Encoder.*

https://arxiv.org/abs/1803.11175

Rehurek, R. *Gensim models - Doc2Vec*.

https://radimrehurek.com/gensim/models/doc2vec.html