Università degli Studi di Brescia
Dipartimento di Ingegneria dell'Informazione
Dottorato di Ricerca in Ingegneria Dell'Informazione - XXXIII Ciclo


# Attention Mechanism e Interpretabilità del Deep Learning per il Natural Language Processing in Ambito Biomedico

*Relatore*
Dr. Alberto Lavelli
f.to digitalmente ex art.24
*Correlatore*
Prof. Ivan Serina
f.to digitalmente ex art.24
*Tutor*
Prof. Alfonso E. Gerevini

*Studente*
Luca Putelli
matricola 721221
f.to digitalmente ex art.24

Settore Scientifico Disciplinare ING-INF/05

# Indice

# Elenco delle figure

# Elenco delle tabelle

The trouble with modern education is
you never know how ignorant people are.
With anyone over fifty you can be fairly confident
what's been taught and what's been left out.
But these young people have such an intelligent,
knowledgeable surface, and then the crust
suddenly breaks and you look down into
depths of confusion you didn't know existed.
(Evelyn Waugh, Brideshead Revisited)

# Capitolo 1

# Introduzione

Questa tesi illustra lo studio e il lavoro svolto presso il Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Brescia durante la mia frequentazione del corso di Dottorato di Ricerca in Ingegneria dell'Informazione, più precisamente nel curriculum di Ingegneria Informatica e Automatica, XXXIII ciclo. Ho svolto le mie attività sotto la supervisione del Prof. Alfonso Emilio Gerevini e del Prof. Ivan Serina dell'Università degli Studi di Brescia, e del Dr. Alberto Lavelli, ricercatore della Fondazione Bruno Kessler di Trento.

## 1.1 Il problema affrontato

La capacità di creare parole, di associarvi un significato, di combinarle per formare una frase affinché sia possibile la costruzione e la trasmissione di pensieri complessi attraverso il linguaggio naturale è una delle caratteristiche che definiscono l'essere umano. Contenuti sotto forma di testo permeano la nostra società sotto forma di libri, giornali, brevi appunti scritti su un biglietto o nei monitor di televisori, computer, smartphone e altri dispositivi elettronici.

Proprio la capacità di scrivere tramite i software di word processing, di immagazzinare contenuti testuali su supporti elettronici di memoria di massa e di diffonderli in via telematica ha aperto nuove possibilità di interpretazione

e utilizzo del testo. Esattamente come i numeri e le immagini, infatti, un testo rappresentato in formato elettronico può essere analizzato in modo automatico tramite un algoritmo.

Dalla metà degli anni '90, con lo sviluppo di Internet, la quantità di testi in formato elettronico è aumentata a dismisura. Se prendiamo in considerazione database di articoli scientifici come PubMed[1] o arXiv[2], enormi quantità di informazioni sono disponibili per essere lette e interpretate e quindi ricavarne conoscenza utile dal punto di vista scientifico o medico. Dall'altro lato, gli utenti di Internet e dei social network generano continuamente testo sotto forma di commenti, recensioni ai prodotti che utilizzano, ai luoghi in cui hanno trascorso le vacanze ecc. Nel campo del marketing, capire le opinioni delle persone su di un prodotto, una notizia o un personaggio pubblico può essere fondamentale per approntare strategie pubblicitarie con notevoli effetti economici. Tuttavia, in entrambi i casi l'analisi del testo richiede l'intervento umano, potenzialmente anche di persone con conoscenze tecniche elevate, rendendo quindi il processo molto dispendioso sia in termini economici che di tempo.

Per questi motivi, l'elaborazione automatica del linguaggio naturale, meglio conosciuta con il termine inglese di **Natural Language Processing** (NLP), è un settore fondamentale nell'ambito della ricerca a cavallo tra informatica, linguistica e intelligenza artificiale. Il suo compito principale è far sì che un software possa capire il significato di un testo, estrarne conoscenza utile, tradurlo in un'altra lingua e molte altre possibili applicazioni. Se la ricerca in questo ambito è iniziata negli anni '50, negli ultimi due decenni la maggiore disponibilità di dati e di risorse computazionali hanno permesso l'applicazione di tecniche di **Machine Learning** nel campo del Natural Language Processing, ottenendo ottimi risultati.

Il Machine Learning è un settore fondamentale dell'intelligenza artificiale che sviluppa tecniche e algoritmi capaci di *imparare* dai dati in modo automatico [58]. Se ad esempio si vuole creare un software che riconosca se una recensione di un film è positiva o negativa, si può utilizzare un algoritmo di

[1] https://pubmed.ncbi.nlm.nih.gov
[2] https://arxiv.org

Machine Learning a cui vengono dati in input una serie di esempi (il *training set*) di recensioni opportunamente etichettate da un essere umano come *positivi* o *negativi* in base all'opinione che esprimono. L'algoritmo quindi imparerà quali sono gli aspetti dei dati di input che denotano un'opinione positiva o negativa, ad esempio focalizzandosi su aggettivi quali *bello*, *divertente*, *noioso*, ecc. Le prestazioni del sistema vengono poi valutate su di un *test set* di altri esempi non utilizzati per l'addestramento.

La possibilità di estrarre informazione dal testo in modo automatico, specialmente se non richiede la creazione di tecniche specifiche in base al contesto applicativo ma piuttosto di sfruttare le tecniche già ampiamente studiate di Machine Learning, offre potenzialità molto importanti anche per velocizzare il lavoro dell'essere umano, specialmente in un settore in cui l'analisi del testo è resa difficile dal linguaggio altamente tecnico e in cui la conoscenza estratta può essere di grande aiuto per prendere decisioni: l'ambiente clinico e biomedico.

In un ospedale infatti moltissime informazioni riguardanti lo stato di salute dei pazienti possono essere reperite sotto forma di testo: referti, schede di dimissione, note sulle cartelle cliniche, ecc. La capacità di poter estrarre conoscenza in forma aggregata da questi documenti tuttavia richiede conoscenza medica e notevole dispendio di tempo e energie da parte di esperti del dominio. Allo stesso tempo, studi scientifici su malattie, terapie o farmaci vengono pubblicati quotidianamente e resi disponibili pubblicamente. Tuttavia, la quantità di testi disponibili è talmente elevata che non può essere consultata in toto, per cui, ad esempio, un meccanismo automatico di filtraggio che verifichi se un articolo può essere rilevante o meno per il trattamento di una malattia può essere molto utile.

Se però da un lato c'è un notevole interesse relativo all'applicazione di tecniche di Machine Learning e Natural Language Processing nell'ambito clinico e biomedico, dall'altro questo genere di documenti presenta un gergo tecnico molto più specifico e problematico rispetto al linguaggio corrente che si può trovare nel commento a un film o in un articolo di giornale. Questo pone due problemi [45, 72]:

- Gli algoritmi di Machine Learning devono essere abbastanza complessi

da poter analizzare il testo maggiormente nel dettaglio. Se ad esempio per distinguere un articolo riguardante la farmacologia da uno che riguarda l'informatica può essere sufficiente individuare alcuni termini (quali ad esempio *dose* o *paziente* invece di *hardware* o *computazione*), tutt'altra questione riguarda capire che tipo di interazione tra farmaci è descritta nell'articolo.

- La raccolta e la creazione degli esempi con cui addestrare gli algoritmi di Machine Learning richiede maggior tempo e risorse, visto che un testo va analizzato più approfonditamente e da personale specializzato. Se infatti è immediatamente riconoscibile l'opinione espressa in un commento a un prodotto, capire se un referto medico descrive una particolare malattia è un processo decisamente più complesso e delicato.

I migliori risultati per quanto l'analisi del testo sono stati ottenuti utilizzando tecniche di **Deep Learning**, una branca recente e notevolmente complessa del Machine Learning [6]. Queste tecniche operano sul testo (ad esempio, per capire se un referto descrive una neoplasia o meno) facendo uso di reti neurali profonde, ovvero meccanismi per cui, dato un documento di input, una serie di *neuroni* connessi tra loro lo rielaborano progressivamente fino ad arrivare ad una predizione. Questa predizione viene poi confrontata con il *target*, ovvero il valore corretto che l'esperto ha assegnato al documento in questione. Nel caso in cui la predizione sia risultata sbagliata, la rete neurale modifica i propri parametri interni, cercando quindi di imparare in base all'esperienza attraverso un procedimento matematico chiamato *discesa del gradiente* [34].

Tuttavia, algoritmi maggiormente complessi richiedono anche un numero maggiore di dati. Nell'ambito biomedico, come detto precedentemente, questo rappresenta un problema date le difficoltà di raccolta e soprattutto di interpretazione anche da parte dell'essere umano. L'applicazione quindi delle tecniche di Deep Learning richiede spesso la creazione di modelli più specifici, in cui anche l'introduzione di conoscenza medica può dare un importante contributo nell'aumentare le prestazioni dell'algoritmo.

Inoltre, maggiormente complessa è la tecnica di Machine Learning o Deep Learning impiegata, meno questa è intellegibile all'essere umano. I principali algoritmi utilizzati, reti neurali comprese, infatti sono del tutto assimilabili a *black box* che, dato un input, producono un output senza che si possa facilmente comprendere il motivo. Se pensiamo a un'applicazione di queste tecniche nell'ambito delle decisioni cliniche questo è un problema rilevante. In un algoritmo che riconosca, in base al referto medico, se un paziente ha una lesione di tipo neoplastico o meno, e quindi in base a questo risultato prescrivere o meno un esame di controllo, è importante anche che l'algoritmo fornisca i criteri su cui ha basato le proprie conclusioni in modo tale che il medico possa facilmente capire potenziali errori e bias. In poche parole, l'**interpretabilità** del sistema di Deep Learning riveste un ruolo cruciale nelle applicazioni cliniche e biomediche, proprio per l'estrema delicatezza dei dati che sono coinvolti nell'analisi.

Se per dati numerici e immagini sono state progettate tecniche ormai standard [20, 51] per estrarre le caratteristiche (*feature*) più rilevanti che hanno portato un algoritmo di Machine Learning o Deep Learning a fornire una predizione piuttosto che un'altra, per l'analisi del testo queste si sono rivelate piuttosto problematiche [10, 12, 105]. Al contrario, una forte spinta verso l'interpretabilità degli algoritmi di Deep Learning applicati al testo è stata data dall'**Attention Mechanism** [4, 76], un particolare meccanismo che permette di assegnare un peso alle parole in base alla loro utilità per il compito svolto dall'algoritmo. Ad esempio, la descrizione di una lesione può essere *evidenziata* dall'attention mechanism per giustificare che un referto medico descrive una neoplasia.

Questo componente progettato specificatamente per il testo apre un'importante prospettiva all'interpretabilità di un sistema di Deep Learning per il Natural Language Processing. Tuttavia, la sua reale efficacia nel rendere comprensibile il comportamento interno dell'algoritmo è ancora oggetto di dibattito [38, 98]. Inoltre, mentre diversi articoli hanno mostrato il risultato prodotto dall'attention mechanism, evidenziando le parole ritenute più importanti, pochi hanno provato ad analizzare più in generale il suo comportamento [91]. Nel campo biomedico e clinico inoltre, nonostante l'inter-

pretabilità in questo settore sia fondamentale ma viste anche le difficoltà nel reperire dati, i pochi esperimenti condotti hanno dato risultati contrastanti [38].

Questa tesi quindi si propone di analizzare il comportamento dell'attention in due casi distinti di applicazione di tecniche di Deep Learning al Natural Language Processing: la classificazione di referti radiologici in italiano, estratti direttamente dalla pratica clinica quotidiana degli *Spedali Civili di Brescia*, e l'estrazione di interazione tra farmaci dalla letteratura scientifica in lingua inglese.

## 1.2 Struttura della tesi

In questa sezione, presentiamo una breve panoramica sulla struttura dei contenuti esposti in questa tesi.

Nel Capitolo 2 vengono introdotti i concetti base del Natural Language Processing, come la rappresentazione delle parole e i principali compiti affrontati. Successivamente, vengono illustrate le caratteristiche e le difficoltà del NLP in ambito clinico e biomedico. L'ultima parte presenta i concetti più importanti relativi all'interpretabilità nell'ambito dell'analisi del testo.

Lo stato dell'arte nell'ambito del Natural Language Processing e delle tecniche di Deep Learning applicate ad esso è presentato nel Capitolo 3. A partire dal concetto di rete neurale (Sezione 3.1), vengono introdotte le tecnologie di *word embedding*, ovvero di come rappresentare le parole e il loro significato attraverso il Deep Learning (Sezione 3.2). Le reti neurali ricorrenti, che rappresentano lo stato dell'arte per molte applicazioni di NLP, sono mostrate nella Sezione 3.3. L'attention mechanism, come spiegazione teorica del suo funzionamento e delle sue varianti, nelle più recenti architetture di Deep Learning (come Transformer o BERT) e nel suo utilizzo per l'interpretabilità, è presentato nella Sezione 3.4.

I contributi originali di questa tesi sono presentati nei Capitoli 4 e 5, rispettivamente dedicati al problema di classificazione di referti radiologici e all'estrazione di interazioni tra farmaci.

Nel Capitolo 4, innanzitutto vengono mostrate le finalità del progetto e le caratteristiche dei dati. Dato che questo lavoro si colloca come prosecuzione e miglioramento di un sistema già avviato, nella Sezione 4.2 viene illustrato un primo tentativo di risoluzione del problema con tecniche di Machine Learning e attraverso l'utilizzo di annotazioni manuali, ovvero di parti di testo ritenute importanti dai radiologi ai fini della classificazione del referto. Dopo aver analizzato pregi e difetti di questo approccio, nella Sezione 4.3 viene mostrata la nuova proposta basata su tecniche di Deep Learning e introduzione di conoscenza pregressa nella progettazione degli algoritmi. Le prestazioni dei due sistemi vengono analizzate, confrontate e discusse nella Sezione 4.4. La Sezione 4.5 del capitolo è interamente dedicata all'interpretabilità del sistema attraverso l'utilizzo dell'attention mechanism, con un'analisi dettagliata del suo comportamento e di come questo varia in base alle caratteristiche del referto. Viene inoltre presentato un confronto tra il risultato dell'attention mechanism e le annotazioni manuali dei radiologi, mostrando affinità e divergenze tra i due approcci. Infine, vengono presentati dei risultati preliminari derivati dall'impiego di un modello basato sull'architettura BERT per la lingua italiana, che rappresenta lo stato dell'arte per molti task linguistici. Questi risultati vengono poi confrontati con l'architettura basata su reti ricorrenti e attention mechanism.

Nel Capitolo 5 viene presentata una soluzione basata su Deep Learning e attention mechanism per l'estrazione di interazioni tra farmaci all'interno di frasi estratte dalla letteratura scientifica. Dopo una presentazione dei dati e delle tecniche di pre-processing (Sezioni 5.1 e 5.2), vengono presentate diverse possibili architetture, con varianti in base al tipo di attention mechanism considerato e in base a come l'input viene analizzato (Sezione 5.3). Il confronto tra queste soluzioni è presentato nella Sezione 5.4, con particolare attenzione all'effetto dei diversi attention mechanism sulle prestazioni. Viene poi mostrata un'analisi degli errori e un confronto tra le prestazioni del sistema e lo stato dell'arte. Nella Sezione 5.5, il comportamento dell'attention mechanism è valutato anche in funzione dell'interpretabilità del sistema, in un contesto radicalmente diverso da quello dei referti radiologici (affrontato nel Capitolo 4) per diversità di compito, lingua e caratteristiche

dei dati. Infine, nell'ultima parte del capitolo vengono mostrati gli sviluppi più recenti della ricerca per questo task, con l'impiego di vari modelli basati su BERT, analizzando i risultati presenti in letteratura, i loro pregi e difetti e mostrando alcuni esperimenti per il miglioramento dei risultati.

# Capitolo 2

# NLP in ambito biomedico

In questo capitolo vengono illustrati lo scopo e le caratteristiche dei concetti di base per l'analisi automatica del linguaggio naturale (Natural Language Processing, o per brevità NLP), prestando particolare attenzione alla loro applicazione in ambito biomedico.

Lo studio di tecnologie per il trattamento del linguaggio naturale ha seguito parallelamente la storia dell'informatica fin dagli anni '50. Campi quali la traduzione automatica (Machine Translation), l'estrazione di informazioni dal testo (Information Extraction) o la progettazione di sistemi per la comunicazione tra agenti informatici ed esseri umani (Dialogue systems, Question Answering) continuano ad essere tra i più attivi àmbiti di ricerca nel campo dell'Intelligenza Artificiale [47].

Le applicazioni del Natural Language Processing sono molteplici: sistemi come i chatbot o Siri sono capaci di comprendere un testo scritto o parlato e fornire un servizio agli utenti; sistemi di autocompletamento e autocorrezione sono ormai implementati nelle interfacce utente dei motori di ricerca; tecniche di Natural Language Processing possono essere utilizzate per analizzare le opinioni dei consumatori a fini pubblicitari o di marketing, oppure per capire se un documento è rilevante o meno ai fini di una ricerca.

## 2.1 Concetti di base del NLP e applicazioni in campo biomedico

Nello specifico, in questa sezione vengono illustrati i principali compiti e concetti di base relativi alle applicazioni di Natural Language Processing, a prescindere dall'ambito effettivo di utilizzo.

### 2.1.1 Rappresentazione delle parole e dei documenti: il modello bag of words

Mentre un'immagine è immediatamente rappresentabile come una matrice di pixel, ognuno dei quali di un certo colore che a sua volta può essere codificato attraverso 3 numeri da 0 a 255, la rappresentazione delle parole all'interno di un documento non è così immediata.

Da un punto di vista strettamente informatico, infatti, una parola non è altro che una stringa di caratteri, e quindi anche questa può essere rappresentata da una sequenza di numeri in codice ASCII. Quello che però viene tralasciato da questa rappresentazione è ciò che rende fondamentale la parola nel suo utilizzo quotidiano: il suo significato. Il suo ruolo all'interno della frase, le sue caratteristiche grammaticali, come si collega ad altri termini in una determinata lingua, sono qualità impossibili da rappresentare considerando solo la sequenza di caratteri. Se prendiamo ad esempio le parole *rumore* e *fracasso* capiamo immediatamente che sono due sinonimi e che quindi trovare l'una o l'altra in documento è equivalente; tuttavia, analizzando carattere per carattere, non è ravvisabile nessuna somiglianza tra le due.

Nonostante siano informazioni fondamentali, la rappresentazione più semplice di una parola non tiene conto delle sue caratteristiche nel linguaggio e del suo significato. Dato un set di $N$ documenti contenenti $K$ parole diverse, la rappresentazione **one-hot encoding** si costruisce:

1. assegnando un id compreso tra 0 e $K-1$ ad ognuna delle $K$ parole;
2. costruendo un vettore di $K$ posizioni in cui nel valore della posizione corrispondente all'id è 1 ed è 0 in tutte le altre posizioni.

Utilizzando lo one-hot encoding è possibile costruire anche una rappresentazione dei documenti: la **bag of words** [33]. Per i nostri $N$ documenti e $K$ parole, la bag of words non è altro che una matrice $M$ di dimensione $N \times K$ (*document-term matrix*) in cui:

1. ad ogni riga $i \in [1, N]$ corrisponde un documento e ad ogni colonna $j \in [1, K]$ una parola;

2. l'elemento $M_{i,j}$ della matrice è 1 se la parola $j$ è presente nel documento $i$, 0 altrimenti.

Come già detto precedentemente, questo modello, introdotto a metà degli anni '50, non tiene conto di semantica o caratteristiche grammaticali delle parole come la differenza di genere o di numero (ad esempio, le parole *medico* e *medici* sono considerate completamente diverse), nè del fatto che alcuni termini, come ad esempio le congiunzioni o le preposizioni (le cosiddette *stop words*), sono presenti in qualsiasi tipo di documento e possono risultare poco utili ai fini dell'analisi di un documento. Sono stati quindi studiati dei correttivi come l'indice TF-IDF (Term Frequency - Inverse Document Frequency) [78] che permette di attribuire un'importanza maggiore alle parole che più caratterizzano un documento e penalizzare quelle più generiche. Più recentemente, sono stati anche progettati dei metodi più sofisticati di rappresentazione delle parole come le tecniche di word embedding, che verranno illustrate nella Sezione 3.2. Ciò nonostante, per alcuni task molto semplici e con pochi documenti a disposizione, il modello bag of words può essere una rappresentazione capace di far ottenere prestazioni soddisfacenti.

### 2.1.2 Tokenizzazione, POS-tagging, NER

Al fine di ricavare informazioni da un documento, per prima cosa un sistema di NLP deve essere in grado di svolgere *sentence splitting* e *tokenizzazione*, ovvero dividere un documento nelle singole frasi che lo compongono e ciascuna frase nei singoli *token* che la costituiscono. Sebbene molte parole possano essere individuate semplicemente andando a cercare i caratteri che indicano uno spazio, in alcuni casi la tokenizzazione deve gestire segni quali

trattini, barre o punteggiatura che rendono questo compito meno semplice di quanto possa sembrare a prima vista.

Una volta separate le singole parole, si può svolgere il *Part-of-Speech Tagging*, ovvero l'assegnazione ai token della frase delle parti del discorso dal punto di vista grammaticale (nomi, aggettivi, verbi, preposizioni ecc.).

Parallelamente a questo, il task di *Named Entity Recognition* (NER) ha come scopo di individuare e catalogare le parti all'interno del testo che si riferiscono a un'entità specifica (*named entity*), quale una persona, una città, una sostanza chimica o altro.

É importante sottolineare che una named entity può essere composta anche da più parole. Se prendiamo ad esempio la capitale del Sudafrica, *Città del Capo*, le tre parole *Città*, *del* e *Capo* dovranno essere riconosciute dalla NER come appartenenti alla stessa entità. In alcuni casi, un sistema deve tener conto di molteplici difficoltà anche legate alla tokenizzazione: per riconoscere correttamente il nome della competizione sportiva *Alpine Skiing-Women's World Cup Downhill*, il sistema dovrà avere regole per separare i singoli nove token (le sette parole, l'apostrofo e il trattino) ma anche sufficientemente accurato da riunirli in un'unica entità [54].

Nel Capitolo 4 di questa tesi, la Named Entity Recognition viene considerata nel contesto clinico, in particolare nell'analisi di referti radiologici di TAC al torace, cercando di trovare quali termini descrivono una lesione oppure il sito in cui si trova. Ad esempio, nella frase "*Al controllo odierno immodificato il nodulo parzialmente calcifico con diametro massimo di 5mm nel segmento anteriore del lobo superiore destro*" si dovrebbe riconoscere l'espressione *nodulo parzialmente calcifico* come esempio di lesione e *segmento anteriore del lobo superiore destro* come esempio di sito. Riuscire ad individuare queste espressioni in un intero referto può essere molto importante per capire la malattia di cui il paziente soffre, le sue condizioni di salute, eventuali miglioramenti, ecc.

### 2.1.3 Relation e Information Extraction

Il problema di *relation extraction* è quello di riconoscere se, date due entità, nel testo venga espressa una relazione tra di esse e, nel caso ci fosse, capire quale sia all'interno di un insieme di possibilità. Ad esempio, nella frase *Parigi è la capitale della Francia*, tra l'entità *Parigi* e l'entità *Francia* intercorre la relazione *essere capitale*.

Da un punto di vista strettamente applicativo, il task di Relation Extraction può essere considerato come un caso particolare di classificazione in cui le possibili relazioni sono le classi. Tuttavia, ci sono alcune differenze:

- un documento può contenere un numero arbitrario di entità, ognuna delle quali può essere in relazione con un numero arbitrario di altre entità;
- mentre nella classificazione è necessario assegnare una o più etichette, tra due entità è possibile non sia presente alcuna relazione.

Per adattare quindi le tecniche di classificazione e di apprendimento supervisionato al caso della Relation Extraction è necessario prendere in considerazione due entità alla volta e aggiungere una classe *negativa* rappresentante l'assenza di relazione tra le entità considerate [28].

Ad esempio, la frase *La Spagna confina a Nord con la Francia e a Ovest con il Portogallo* contiene tre entità (*Spagna*, *Francia* e *Portogallo*) e poniamo di voler capire quali entità hanno una relazione *confina*. Per fare ciò, è necessario considerare ogni possibile coppia tra le entità (*Spagna-Francia*, *Spagna-Portogallo*, *Francia-Portogallo*) e classificare ognuna di esse scegliendo tra la classe *confina* e quella *negativa*. Nel testo, possiamo facilmente vedere che non viene citato nessun confine tra *Francia* e *Portogallo* per cui quell'istanza andrà classificata come negativa.

Mentre nell'esempio appena illustrato abbiamo una relazione tra due entità dello stesso tipo (e sarà così anche nel Capitolo 5 dove verrà descritta una Relation Extraction per scoprire interazioni tra farmaci), è possibile avere relazioni tra entità di tipo diverso. Ad esempio, in un importante dataset

biomedico, Adverse Drug Events (ADE) [32], abbiamo due entità diverse in relazione tra loro: il farmaco e l'effetto collaterale.

Nel task di Relation Extraction il focus è solamente sulla relazione, per cui i token che compongono le diverse entità vengono precedentemente annotati e poi forniti all'utente come parte dell'input del problema. Partendo però da un semplice documento di testo, come ad esempio l'abstract di un articolo scientifico che descriva un effetto collaterale dovuto ad un farmaco, possiamo considerare un processo a due fasi: la Named Entity Recognition, deputata quindi a trovare le entità che descrivono farmaci ed effetti collaterali, e la Relation Extraction, che li collega nel modo corretto. Questo processo si chiama Information Extraction ed una sua implementazione è, ad esempio, descritta in [63], per l'estrazione di interazioni tra proteine. Solitamente, l'Information Extraction si accompagna all'Entity Linking, ovvero l'operazione di collegare il testo che identifica una particolare entità con la sua descrizione in una base di conoscenza. Ad esempio, la parola *emoglobina* si può collegare all'entità Q43041 di WikiData, che ne fornisce (in un formato che è possibile consultare automaticamente con linguaggi di interrogazione) la formula chimica, le caratteristiche principali e numerosi collegamenti con altre basi di conoscenza.

Una volta svolto anche l'Entity Linking, da un semplice testo scritto è possibile arricchire questi database con informazioni nuove, ad esempio un'interazione tra farmaci appena scoperta, e renderle facilmente accessibili alla comunità scientifica.

### 2.1.4 Classificazione

Uno dei compiti principali del NLP è la classificazione di documenti. Nella sua accezione classica, si tratta di riconoscere ed etichettare correttamente un documento in base all'argomento che tratta. Ad esempio, il benchmark "20 Newsgroup" prevede di classificare messaggi di una newsletter in 20 classi tra cui religione, annunci pubblicitari e hockey [58].

Tuttavia, ereditando il termine dal Machine Learning, il termine di classificazione può essere esteso a qualsiasi problema che richieda l'interpretazione

del documento e l'attribuzione di etichette in base al suo contenuto.

Ci sono moltissimi esempi di classificazione di documenti in ambito clinico: da quelli più semplici come il riconoscimento se il paziente descritto è fumatore o meno oppure se è presente o meno una frattura [95], a quelli più complicati come la classificazione gerarchica di referti radiologici descritta nel Capitolo 4. In quest'ultimo caso, da un referto è possibile riconoscere il tipo di esame che è stato svolto, se il paziente è in peggioramento o meno, la tipologia di lesione descritta nel referto e dove si trova.

In ambito biomedico, un'applicazione molto importante è quella che riguarda la rilevanza di una pubblicazione scientifica per un particolare problema. I numeri sempre più grandi di studi svolti e di articoli disponibili online, pongono seri limiti alla capacità, da parte dei ricercatori, di trovare quelli maggiormente rilevanti per la ricerca che stanno svolgendo. Il sistema proposto in [39], dato un particolare argomento, seleziona appunto automaticamente gli studi scientifici più rilevanti.

## 2.2 Particolarità del NLP in ambito clinico e biomedico

Dall'introduzione delle cartelle cliniche elettroniche, gli ospedali raccolgono una grande quantità di testo che descrive le condizioni dei pazienti, i loro sintomi, i trattamenti a cui sono sottoposti ecc. Questa mole di informazioni, che può essere sfruttata al fine di predire l'evoluzione delle condizioni del paziente, per migliorare l'organizzazione dell'ospedale o per fini di ricerca, tuttavia richiede una notevole quantità di lavoro per essere analizzata [24, 72].

Oltre all'intrinseca difficoltà di estrarre conoscenza da dati completamente non strutturati come il testo libero, il testo clinico presenta ulteriori difficoltà [45] quali:

- gergo tecnico estremamente specifico, con utilizzo di convenzioni e abbreviazioni definite all'interno del singolo ospedale o del singolo reparto;

- contesto poco definito e poco uniforme, in cui segni come trattini, barre ecc. possono sottintendere cose diverse in base al contesto. Ad esempio, nella frase *Lieve incremento (3-5 mm) del nodulo* il trattino significa che il nodulo si è ingrandito da 3 a 5mm; invece nella frase *Minute formazioni litiasiche (1-2mm)*, sta a indicare che le formazioni litiasiche hanno una dimensione compresa tra 1 e 2mm;

- grammatica atipica e sintassi destrutturata, tra cui mancanza di articoli e preposizioni o di verbi. Ad esempio, la frase *Non masse mediastiniche.* non presenta né soggetto né predicato verbale;

- errori di battitura, dovuti al fatto che il testo è scritto, o spesso dettato a un software, dai medici spesso cercando di minimizzare la quantità di tempo necessaria alla stesura del testo clinico.

Oltre a questo, compiti come la Relation Extraction o la classificazione richiedono algoritmi di apprendimento supervisionato che necessitano che il testo non sia solo disponibile ma anche annotato manualmente in modo corretto. Se il comprendere l'argomento di un articolo di giornale può essere fatto in modo rapido da pressoché chiunque, capire se un referto descrive una determinata malattia è possibile solo per un medico e con un impiego di tempo più consistente. Per questo motivo, la creazione di dataset specifici in campo clinico è molto più dispendioso e le loro dimensioni, che per un algoritmo di Machine Learning sono fondamentali, sono molto più ridotte rispetto a quelli generici.

Anche la qualità delle annotazioni e delle etichette di classificazione, in ambito clinico, può presentare dei problemi. Nella pratica, infatti, nel valutare la presenza o lo stato di una malattia, di una risposta a un trattamento, è possibile un certo grado di libertà e quindi divergenze tra un medico e l'altro, in quanto possono avere esperienza e formazioni diverse [24, 79]. Nell'apprendimento supervisionato, questo riveste chiaramente un problema in quanto l'algoritmo è chiamato ad apprendere su esempi che possono essere incerti, contraddittori e in cui è presente un tasso di errore intrinseco all'applicazione.

Oltre al testo clinico, nell'ambito biomedico viene prodotta un'enorme quantità di testo anche sotto forma di pubblicazioni scientifiche. Essendo

queste pubbliche, senza le problematiche di privacy e di sensibilità dei dati presenti nel testo clinico, numerosi dataset [96, 88] e applicazioni [50] sono state realizzate negli anni per estrarre informazioni da questo genere di pubblicazioni per i principali compiti di NLP. Chiaramente, in una pubblicazione scientifica non sono presenti i problemi relativi alla forma, alle convenzioni e alla mancanza di struttura del testo clinico ma permangono i problemi relativi alla difficoltà di annotazione, di incertezza e della necessità di strumenti specifici per trattare questo genere di dati. Per esempio, possiamo facilmente intuire le difficoltà nel capire che l'espressione *1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine*, composta da 17 token, rappresenta un singolo composto chimico generato dall'assunzione di un farmaco.

Per questo motivo, sono stati creati e addestrati strumenti o modelli specifici, quali ad esempio *scispaCy* [61] (versione biomedica del popolare tool per il NLP spaCy[1]), per trattare il testo in ambito biomedico. Allo stato attuale, tuttavia l'utilizzo di strumenti pre-addestrati come *scispaCy* è confinato a testi clinici e biomedici in lingua inglese. Mentre strumenti per documenti generici sono presenti in diverse lingue, per il linguaggio tecnico, che come abbiamo visto richiede una notevole mole di lavoro per essere annotato, le risorse disponibili sono quasi solamente le pubblicazioni scientifiche, comunemente disponibili solo in inglese. Trattare quindi il testo clinico in un'altra lingua richiede di scegliere tra:

- creare e addestrare un sistema ex novo specifico per la lingua e per l'applicazione in questione, impiegando una notevole quantità di tempo e risorse;
- utilizzare strumenti, tool e modelli già disponibili per documenti di tipo generico, accettando una percentuale di incertezza e di errore che tuttavia può anche essere significativa;
- definire regole e procedure per adattare quanto già disponibile all'applicazione specifica.

[1] https://spacy.io

## 2.3 Il problema dell'interpretabilità

Sia nell'ambito dell'Intelligenza Artificiale e del Machine Learning che nello specifico del NLP, molteplici sforzi si sono profusi affinché le decisioni o le predizioni fornite da un algoritmo possano essere comprese dagli esseri umani che utilizzano il sistema (*Explainable AI).*

Rimanendo nello specifico delle applicazioni cliniche, è evidente come il compito di formulare una diagnosi, suggerire una terapia o individuare un rischio analizzando i dati di un paziente, quali ad esempio un referto medico o i risultati degli esami clinici, non possa essere lasciato completamente a un algoritmo. Accanto ad esso, è necessario che agiscano medici che controllino e validino le decisioni prese con la loro conoscenza del dominio. Sapere quindi quali siano le caratteristiche più importanti su cui il modello si focalizza, quali dati ha tenuto in considerazione oppure perché è stata presa una particolare decisione, è molto importante affinché il sistema possa essere utilizzato proficuamente in ambiente clinico [77].

Nel trattamento del linguaggio, la questione si complica ulteriormente. Ad esempio, in un task di classificazione per argomento, per giustificare il fatto che un documento è classificato come di tipo *referto medico* può essere sufficiente evidenziare la presenza delle parole *lesione*, *nodulo* e *flogistica*; al contrario, capire ad esempio se la lesione di cui parla il referto è di tipo neoplastico o meno richiede un'analisi più approfondita che tenga conto non solo della presenza delle parole, ma anche il loro significato e come si relazionano tra loro. Per questo non solo è necessario un modello più complesso, ma anche la stessa spiegazione può risultare meno intelligibile.

### 2.3.1 Definizioni e concetti preliminari

Dati i recenti successi in termini di prestazioni delle tecniche di deep learning per il NLP, pubblicazioni molto recenti [38, 98, 84] hanno iniziato a definire potenzialità e limiti di questi modelli in termini di interpretabilità. Tuttavia, questi lavori perseguono obiettivi e usano tecniche diverse, senza una visione d'insieme. Partendo dal lavoro in [37], in questa sezione verranno

introdotti i concetti chiave nella definizione del problema di interpretabilità in NLP.

Innanzitutto, data una risposta fornita da un modello di Natural Language Processing (quale ad esempio se in un referto è descritta la presenza di una patologia o meno), **la sua interpretazione è il procedimento che permette di fornire all'utente le parti dell'input (ovvero i token) che motivano la risposta**. Se prendiamo ad esempio un task di Relation Extraction per trovare se due farmaci assunti contemporaneamente (in questo caso *aripiprazole* e *antihypertensive agents*) possono portare a effetti indesiderati, nella frase "*Given the primary CNS effects, caution should be used when aripiprazole is taken in combination with antihypertensive agents.*" le espressioni *caution should be used* e *is taken in combination* sono fondamentali per definire che tra i due farmaci possa esserci un'interazione di tipo negativo.

Possiamo definire due tipi di interpretazione:

- L'interpretazione **plausibile**, quando i token sottolineati dal sistema possono convincere un essere umano della bontà della decisione presa. Per avere un'interpretazione plausibile è quindi necessario che la predizione del sistema sia quella corretta.

- L'interpretazione **fedele**, cioè quando i token evidenziati dal sistema riflettono effettivamente quelli che il modello ritiene più importanti. Per avere un'interpretazione fedele quindi non è necessario che la predizione sia corretta: deve solo rispecchiare il ragionamento interno del modello.

Andando più nel dettaglio, in letteratura spesso vengono utilizzate due assunzioni per definire quando un'interpretazione non è fedele:

- Dati due modelli che prendono la stessa decisione su di un input, se la loro spiegazione è diversa allora l'interpretazione non è fedele [98].

- Dati un modello e due input simili, l'interpretazione non è fedele se fornisce due spiegazioni diverse [38, 98].

Queste assunzioni, che vengono spesso utilizzate per creare dei controesempi con cui dichiarare non fedele un sistema di interpretazione, possono

essere troppo restrittive in quanto non tengono conto innanzitutto del fatto che un sistema di interpretazione di un modello comunque ha la necessità di semplificare (per poter risultare comprensibile all'essere umano) il reale comportamento del modello stesso. In secondo luogo, le due assunzioni definiscono la questione in modo binario: fedele o non fedele. Un sistema di interpretazione, tuttavia, può essere utile nella pratica per un compito specifico, o per alcune categorie dell'input. Ad esempio, in ambito clinico, un sistema interpretazione può sottolineare le parti più importanti di un referto solo quando viene effettivamente rilevata una malattia; invece, se nel referto non viene rilevato nulla, non è interessante sapere quali parti del testo hanno portato il modello a quella decisione [37].

Infine, i metodi di interpretabilità possono essere suddivisi in due macrocategorie, che prescindono anche dall'applicazione nel NLP:

- metodi post-hoc, che lavorano su un modello già allenato ma non interpretabile e utilizzano tecniche matematico-statistiche per fornire una spiegazione. Un esempio di questo tipo può essere l'algoritmo SHAP [51, 105] che, data una predizione per un'istanza di input rappresentata da un insieme di feature $F$ (nel caso della rappresentazione bag-of-words, questo è equivalente al vocabolario), assegna un valore numerico $s$ a ciascuna feature $f \in F$ che quantifichi quanto il modello si sia basato sul valore di $f$ per la predizione.

- metodi intrinsecamente interpretabili, la cui formulazione è già di per sé comprensibile all'essere umano. Un esempio classico è certamente quello degli alberi di decisione che verificano una serie di condizioni sull'input seguendo delle regole gerarchiche.

# Capitolo 3

# Deep learning per NLP

Negli ultimi anni, le tecniche di deep learning hanno raggiunto le migliori prestazioni nell'analisi dei dati, fissando nuovi standard anche per il trattamento del linguaggio naturale. Da un lato, l'utilizzo di reti neurali ha permesso una migliore rappresentazione delle parole; dall'altro le reti neurali ricorrenti e in particolare le Long Short Term Memory networks (LSTM) sono capaci di analizzare l'intera sequenza di parole in un testo, con la potenzialità quindi di cogliere relazioni tra parole, tenere in memoria il contesto della frase, ecc. Le rete ricorrenti quindi possono essere utilizzate per i task di NLP come la classificazione o la NER. Oltre a questo, l'attention mechanism permette di *prestare attenzione* solo alle parti più importanti del documento in input. In questo capitolo, quindi, viene presentato lo stato dell'arte delle tecniche di deep learning per il NLP.

## 3.1 Reti neurali feedforward

In questa sezione, vengono presentate brevemente le reti neurali feedforward, che sono tra le tecniche più utilizzate ed efficaci nell'ambito del machine learning [34].

Una rete neurale feedforward è composta da diversi *livelli* di *neuroni* che formano un grafo orientato e aciclico. Nell'esempio in Figura 3.1, è possibile vederne una con quattro livelli: partendo da sinistra uno con quattro neuroni,

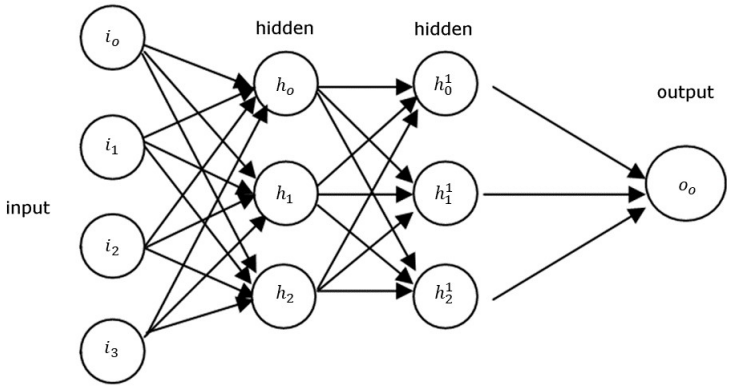


Figura 3.1: Esempio di rete neurale feedforward con due livelli nascosti

due con tre e l'ultimo con un solo neurone. Come si può intuire dalle frecce, ogni neurone è collegato con tutti quelli del livello successivo. Per questo motivo, questi livelli e queste reti neurali vengono anche detti *fully-connected*.

Il primo livello è detto livello di input, in quanto riceve le informazioni in ingresso. Se in un problema di machine learning le istanze da predire sono rappresentate tramite un vettore reale di dimensione $f$, il livello di input è composto da $f$ neuroni, uno per ogni feature in ingresso.

Il livello di input è collegato completamente con il primo livello nascosto (*hidden layer*), che può essere di dimensione variabile. Ciascuno dei neuroni del livello nascosto computa una somma pesata degli input a cui è collegato. Ad esempio, il neurone $h_0$ è collegato con $i_0, i_1, i_2$ e $i_3$, per cui:

$$w_{0,0} * i_0 + w_{1,0} * i_1 + w_{2,0} * i_2 + w_{3,0} * i_3 \tag{3.1}$$

in cui $w_{i,j} \in \mathbb{R}$ è il peso numerico assegnato al collegamento tra il neurone $i \in [0, 3]$ del livello di input e il neurone $j \in [0, 2]$ del livello nascosto. Più in generale, l'operazione fatta dal livello nascosto è:

$$H = \sigma(WI + B) \tag{3.2}$$

in cui, dato un livello nascosto di dimensione $h$, $I \in \mathbb{R}^f$ è il vettore di ingresso, $W \in \mathbb{R}^{f \times h}$ è la matrice dei pesi e $B \in \mathbb{R}^h$ è un vettore di pesi aggiuntivi chiamato *bias vector*. La funzione $\sigma$ è la cosiddetta funzione di

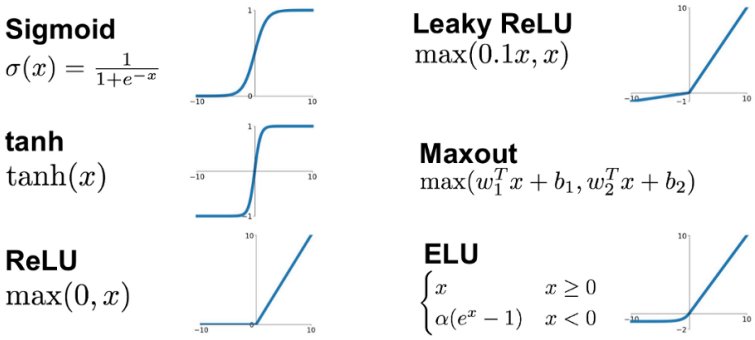


Figura 3.2: Alcuni esempi di funzioni di attivazione

attivazione, a cui appartengono ad esempio la sigmoide o tanh e altre visibili in Figura 3.2. La computazione è esattamente la stessa anche per il secondo livello nascosto o per qualunque degli eventuali successivi, semplicemente considerando come vettore $I$ non più i valori del livello di input ma quelli prodotti dal precedente livello nascosto.

L'ultimo livello è quello di output, ed è quello che fornisce in output il risultato del modello predittivo. La dimensione dipende dal task: in una classificazione binaria, è sufficiente un neurone con attivazione sigmoide, in modo tale che fornisca 0 o 1 in uscita. Nel caso di scelta tra più classi invece, si utilizza la funzione softmax che fornisce in output la distribuzione delle probabilità per le varie classi.

In una rete neurale feedforward ogni livello ha quindi la propria matrice di pesi $W$ e il vettore di bias $B$, che sono dei parametri del modello da imparare mediante l'addestramento della rete. Essendo una rete neurale un modello di apprendimento supervisionato, alla rete neurale è necessario fornire dei dati di *training*, ovvero coppie tra un'istanza di input $x$ e il valore *target* di output $y$ che servano per trovare i migliori parametri. Inizialmente, questi parametri, che per comodità chiameremo $P$, sono inizializzati in modo casuale. La rete quindi esegue i propri calcoli e fornisce un valore $\hat{y}$ in uscita per ogni istanza, che viene confrontato con $y$. Date $n$ istanze di training, viene quindi calcolata una funzione di errore (*loss function*), come ad esempio la cross-entropy:

$$\mathbb{L}(P) = \sum_{i=1}^{n} [-y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \tag{3.3}$$

L'obiettivo dell'algoritmo è minimizzare la loss function $\mathbb{L}$, ovvero trovare la configurazione di $P$ che produca meno errori possibile. La minimizzazione della funzione e la conseguente modifica dei parametri vengono svolte attraverso il procedimento matematico di discesa del gradiente con l'algoritmo di *backpropagation* [82]. L'aggiornamento dei pesi si può svolgere ad ogni istanza (*stochastic gradient descent*), considerando tutto l'insieme dei dati di training (*batch gradient descent*) oppure solamente una parte di dimensione selezionabile (*mini-batch*). In ogni caso, l'addestramento prevede di riutilizzare più e più volte gli stessi dati, al fine di migliorare progressivamente la qualità della predizione. L'utente ha quindi la possibilità di selezionare anche il numero di *epoche* dell'addestramento, ovvero quante volte la rete dovrà analizzare tutti i dati di training.

## 3.2 Word embedding

Come detto nella sezione 2.1.1, il modello bag of words non riesce a rappresentare concetti importanti del linguaggio quali le differenze di genere e numero, i sinonimi, raggruppare parole dello stesso argomento e simili.

Oltre a questo, nella *document-term matrix* a ogni parola corrisponde una colonna, per cui ogni minima variazione dello stesso termine (pensiamo ad esempio a tutte le coniugazioni di un verbo) è rappresentata da una colonna a sé stante. Inoltre, considerando un problema di classificazione di testi per argomento, possiamo anche capire come un grande numero di parole siano relative al gergo di un argomento specifico e non siano presenti negli altri. Per questo motivo, un documento viene rappresentato (attraverso una riga nella matrice) da molteplici 0, corrispondenti alle parole appartenenti ad argomenti estranei al suo contenuto, e da una minoranza di 1. La struttura dati prodotta dalla bag-of-words tende quindi sia ad aver problemi di dimensione che di sparsità, con notevole spreco di memoria.


Figura 3.3: Proiezione bidimensionale di una rappresentazione word embedding. Ogni parola è rappresentata come un punto nello spazio.

L'idea del word embedding è quindi di fornire una rappresentazione più compatta delle parole tenendo conto del loro significato e del loro utilizzo. Ogni parola infatti è rappresentata come un vettore di $n$ numeri reali (*word vector*), cioè da un punto in uno spazio vettoriale di dimensione $n$. Questa rappresentazione viene costruita in modo tale che:

- parole appartenenti allo stesso argomento siano nella stessa regione dello spazio, con particolare vicinanza tra i sinonimi oppure per parole che variano solo per genere, numero o coniugazione;
- parole con significato e contesto di utilizzo molto diversi siano rappresentate da punti molto distanti.

Nella Figura 3.3 viene mostrato un esempio a due dimensioni di una rappresentazione word embedding. In basso a destra, possiamo notare tre parole che rappresentano tipi di frutta *Kiwi*, *Apple* e *Orange*, molto distanti da *Cat* e *Dog*, a loro volta invece vicini tra loro.

Tuttavia, mentre la bag of words è immediatamente ricavabile per qualsiasi corpus di parole e documenti, calcolare una rappresentazione word embedding è un procedimento notevolmente complesso che richiede l'utilizzo di reti neurali e una mole notevole di documenti da analizzare, di tempo e di risorse computazionali per ottenerne una soddisfacente. Negli ultimi anni, diversi

algoritmi come GloVe [66], FastText [41] e soprattutto Word2Vec [57] sono stati progettati allo scopo di *comprendere il funzionamento del linguaggio* e ottenere una rappresentazione word embedding.

Mentre l'addestramento di un modello di word embedding è una procedura che richiede tempo e risorse computazionali, è altrettanto importante sottolineare che, una volta addestrato, il modello è riutilizzabile per diversi task. Un modello per la lingua italiana, ad esempio costruito sulla base dell'intero corpus di Wikipedia, potrà essere utilizzato sia per un task di classificazione di articoli di giornale che per una Named Entity Recognition degli attori in un corpus di recensioni cinematografiche, senza necessità di doverne addestrare uno specifico. Inoltre, molti modelli già allenati sono disponibili online.

Nonostante ciò, il testo clinico e biomedico ha un vocabolario altamente specifico che non può essere catturato da un corpus generico: in questo caso è necessario quindi addestrare un modello, o procurarsene uno già pronto [74], basato su testi clinici o pubblicazioni mediche. La reperibilità di modelli addestrati su un dominio specifico è piuttosto buona per la lingua inglese (come quello giuridico [11] o matematico [30]) ma molto più difficile per lingue meno utilizzate, come ad esempio quella dell'italiana. Questo può rappresentare un serio limite nell'applicazione delle tecniche di word embedding negli ambiti potenzialmente più complicati. Qualora si intendesse procedere all'addestramento di un modello in proprio, la necessità di reperire una mole considerevole di documenti pone inoltre ulteriori problemi.

Un altro difetto intrinseco delle tecniche di word embedding è il caso degli omonimi, ovvero parole con la stessa grafia ma con significati diversi (ad esempio *riso* può indicare l'atto di ridere così come un cereale). In questo caso, il vettore risultante sarà una sorta di combinazione tra i possibili significati, in base a quante volte uno o l'altro compaiono nei documenti utilizzati per l'addestramento. I modelli di word embedding sono inoltre molto sensibili a piccoli errori, per cui una parola contenente ad esempio un errore di battitura non può essere rappresentata tramite un vettore di numeri reali.

### 3.2.1 Word2Vec

L'idea fondamentale di questo algoritmo creato nel 2013 è un'assunzione tipica del linguaggio: il significato di una parola può essere intuito dal contesto in cui è inserita. Se noi prendiamo ad esempio la frase incompleta *Sto mischiando il _ di carte*, possiamo facilmente capire che al posto di _ possiamo inserire la parola *mazzo* per formare una frase di senso compiuto.

Il modello iniziale di Word2Vec, detto anche Continuous Bag of Words (CBOW), esegue la stessa operazione, cioè predire una parola dato il suo contesto, utilizzando una rete neurale. Più formalmente, data la n-esima parola di una frase $w_n$ e considerando una *context window* di 2 parole, utilizza le parole precedenti $w_{n-1}$, $w_{n-2}$ e quelle successive $w_{n+1}$ e $w_{n+2}$ per predire $w_n$. L'input è un vettore di lunghezza pari a quella del vocabolario, in cui è presente il valore 1 nelle posizioni corrispondenti agli indici delle parole in ingresso (il contesto della frase) e 0 nelle altre. In output ci sarà un vettore di tutti 0 tranne che nella posizione dell'indice di $w_n$, in cui è presente il valore 1.

La rete neurale è composta, oltre che dal necessario livello di input, da un livello nascosto di dimensione $h$ e da uno di output, entrambi fully-connected. Mentre il primo ha attivazione lineare, il secondo ha come attivazione la funzione *softmax*. Data $w^i$, cioè la parola a cui è stato assegnato l'indice $i$ nel livello di input, gli $h$ pesi che collegano il neurone $i$ a ciascun neurone del livello nascosto formano la rappresentazione word embedding della parola $w^i$ [81] che è quindi un vettore di dimensione $h$.

In Figura 3.4 è riportato un esempio volutamente semplice con un corpus di sole otto parole, ognuna delle quali associata a un indice tra 0 e 7 (*apple* ha indice 0, *drink* 1 e così via). In questo esempio, dato il contesto formato dalle parole *apple* e *eat*, si vuole predire la parola *orange*. Per cui, a sinistra i neuroni di input sono colorati di rosso nelle posizioni corrispondenti agli indici di *apple* e *eat*. Allo stesso modo, a destra l'output desiderato è quello con un 1 solamente nella posizione di *orange*. Il livello nascosto, al centro, è formato da soli 5 neuroni ed è completamente connesso sia ai neuroni di input che a quelli di output. Prendendo ad esempio la parola *eat*, i 5 collegamenti del neurone

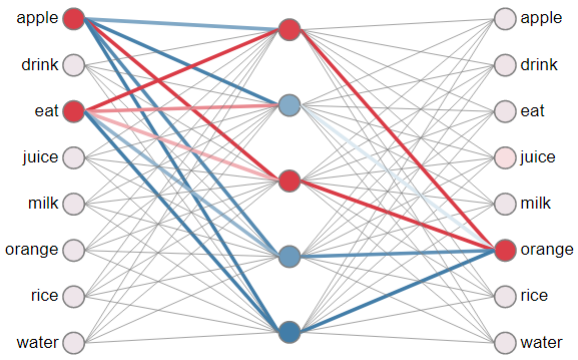


Figura 3.4: Visualizzazione semplificata della rete neurale di Word2Vec su un corpus di 8 parole. Date le parole *apple* e *eat* si vuole predire la parola *orange*, con un livello nascosto di dimensione 5. Il colore rosso indica pesi positivi, il blu negativi e più la tonalità è forte più il valore è alto. I pesi che collegano ciascuna parola (e quindi il corrispondente neurone di input) al livello nascosto sono il word embedding della parola stessa.

di input corrispondente a quella parola al livello nascosto rappresentano il word embedding di *eat*, che è quindi un vettore di 5 elementi. In questa rappresentazione grafica, più il colore del collegamento è rosso, più il peso associato a quel collegamento è vicino a 1, mentre più è blu più è vicino a $-1$. Per cui, il word embedding di *eat* mostrato nella figura può essere ad esempio il vettore $[0.7, 0.25, 0.2, -0.3, -0.5]$. La stessa rappresentazione può essere utilizzata anche per i collegamenti tra il livello nascosto e l'output, ma non corrispondono a nessun word embedding.

Dato quindi un corpus di documenti, è possibile generare un numero anche molto elevato di esempi esaminando tutte le parole di una frase e prendendo il relativo contesto. Ognuno di questi esempi può essere usato per addestrare la rete neurale che, attraverso l'algoritmo di backpropagation, modificherà i pesi dal livello di output al livello nascosto e tra quest'ultimo e quello di input, cioè il word embedding. Dopo quindi aver analizzato molti esempi e per molte epoche, la rete è in grado di predire una parola dato il suo contesto. Per fare ciò, la rete deve quindi aver costruito una rappresentazione che contenga informazioni relative al significato delle parole e al funzionamento del linguaggio, con un guadagno notevole rispetto alla Bag of Words. Questa

rappresentazione può essere poi estratta dalla rete neurale e riutilizzata per diversi task di NLP.

Una variante di Word2Vec e, nel caso di corpus di documenti molto grandi, più performante è il modello Skip-Gram. Il suo funzionamento è simile a quello della CBOW, solo con la differenza che in input viene data una singola parola e in output bisogna predire quelle che formano il contesto. Ad esempio, dalla frase "*The cat sat on the mat*" possiamo estrarre diverse coppie: usando *cat* come input possiamo predire *sat*, usando *on* possiamo predire *mat*, ecc.

GloVe [66] invece sfrutta lo stesso concetto in un modo leggermente diverso. Questo algoritmo infatti addestra un modello per predire la probabilità di co-occorrenza di due parole nello stesso contesto. Se prendiamo le parole *solido*, *gassoso* e *ghiaccio*, la probabilità di trovare la prima nello stesso contesto della terza $P(solido|ghiaccio)$ sarà molto superiore a $P(gassoso|ghiaccio)$. La possibilità di quantificare questa probabilità, secondo gli autori, può essere un vantaggio rispetto a Word2Vec.

Una volta ottenuti i word vectors di dimensione $h$ (tipicamente 100 o 200), a prescindere dalla tecnica utilizzata, la rappresentazione di un documento di $k$ token (parole e punteggiatura) non è altro che una matrice di dimensione $k \times n$ in cui ognuno di essi, nell'ordine in cui compaiono nel testo, è sostituito dal word vector corrispondente.

## 3.3 Reti neurali ricorrenti

In linea teorica, un documento rappresentato utilizzando i word embeddings può essere analizzato da qualsiasi algoritmo di Machine Learning. Ad esempio, potremmo passare la sequenza dei word vectors a una rete neurale fully-connected con input di dimensione data dal prodotto di $k$ per $n$. Questo però pone due problemi:

- ogni singola posizione del word vector è messa in ingresso separatamente, lasciando alla rete il compito di capire che i primi 200 input si riferiscono alla stessa parola;

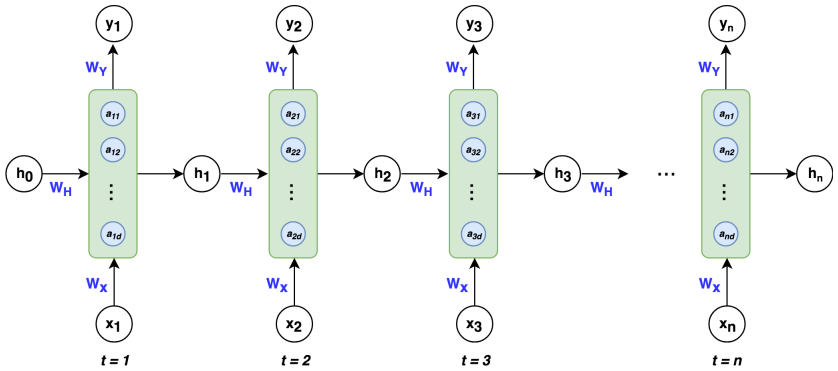


Figura 3.5: Schema di una rete neurale ricorrente funzionante su una sequenza di input $x_t$ con $t \in [1, n]$. $W_x$, $W_h$ e $W_y$ sono matrici di pesi calcolati dalla rete.

- ogni singola dimensione è collegata con tutti i neuroni del livello successivo, aumentando quindi di molto il numero di pesi da allenare;
- non viene presa in considerazione la sequenza con cui le parole si presentano nel testo ed infatti la rete le analizza contemporaneamente.

Per questo motivo, il trattamento del linguaggio (e in generale, delle sequenze temporali) è svolto tramite le Recurrent Neural Networks (RNN, in italiano reti neurali ricorrenti) che analizzano una parola alla volta, computando l'input i-esimo facendo riferimento anche a tutto ciò che è stato analizzato precedentemente.

### 3.3.1 Funzionamento di base

Partendo da un esempio relativo all'analisi del linguaggio, prendiamo un documento composto da $K$ parole. Per la parola t-esima, chiamiamo il suo word vector $x_t \in \mathbb{R}^d$ con $t \in [1, n]$. La RNN [22], come visibile nella Figura 3.5, è composta da $n$ *celle*, ognuna delle quali può essere vista come un livello fully-connected di $n$ neuroni collegato a due input: $x_t$ e $h_{i-1}$ dove quest'ultimo è il risultato della computazione precedente. Definiamo quindi le matrici di pesi $W_x \in \mathbb{R}^{d \times n}$ e $W_h$, $W_y \in \mathbb{R}^{d \times d}$ e i vettori di bias $b_x$, $b_h$, $b_y \in \mathbb{R}^d$ tali che:

$$h_t = \sigma_h(W_x x_t + W_h h_{t-1} + b_h) \tag{3.4}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{3.5}$$

utilizzando come attivazione $\sigma_h$ le funzioni tanh o $ReLU$.

L'Equazione 3.4 mostra come la rete processi gli elementi in sequenza tenendo conto di $x_t$ e del risultato della computazione precedente $h_{t-1}$, formando una nuova memoria. Invece, l'Equazione 3.5 calcola l'uscita della cella RNN: ad esempio, per un task quale la Named Entity Recognition $y_t$ può indicare se la parola ti-esima indica una nazione, una persona o una città. Se invece è necessario analizzare l'intero testo, come in un task di classificazione, è sufficiente considerare l'ultima uscita $y_K$, che viene calcolata in base a tutta la sequenza di parole. La scelta quindi di $\sigma_y$ può dipendere quindi dal task considerato: se si tratta di una classificazione tra più di due classi, ad esempio, è opportuno scegliere la funzione softmax.

È fondamentale sottolineare il fatto che le matrici di pesi e i vettori di bias sono condivisi per tutti gli input della sequenza. Questo sfrutta il fatto che, in linea di massima, una parola posta in un punto invece che in un altro dello stesso documento ha lo stesso significato e dev'essere trattata allo stesso modo. Questo accorgimento fa sì che il numero di pesi sia molto minore rispetto a quello che si avrebbe utilizzando una fully-connected, con notevole risparmio in termini di tempo e risorse computazionali.

Le RNN possono inoltre essere bidirezionali, cioè analizzano la sequenza contemporaneamente nell'ordine sia in cui si presenta che al contrario. Dati $h_t$ e $h_t^r$, con $h_t^r$ che rappresenta il risultato della computazione della sequenza in ordine inverso, l'output di una cella di una RNN bidirezionale $h_t^b$ è dato alla concatenazione di $h_t$ e $h_t^r$.

### 3.3.2 Long Short Term Memory

I difetti delle RNN tradizionali relativi alla difficoltà di addestramento [64] e il fatto di non poter facilmente "dimenticare" alcune parti della sequenza poco utili ai fini della predizione [26] hanno causato lo sviluppo e l'utilizzo di modelli più sofisticati di reti neurali ricorrenti come le Long Short Term Memory (LSTM) [36].

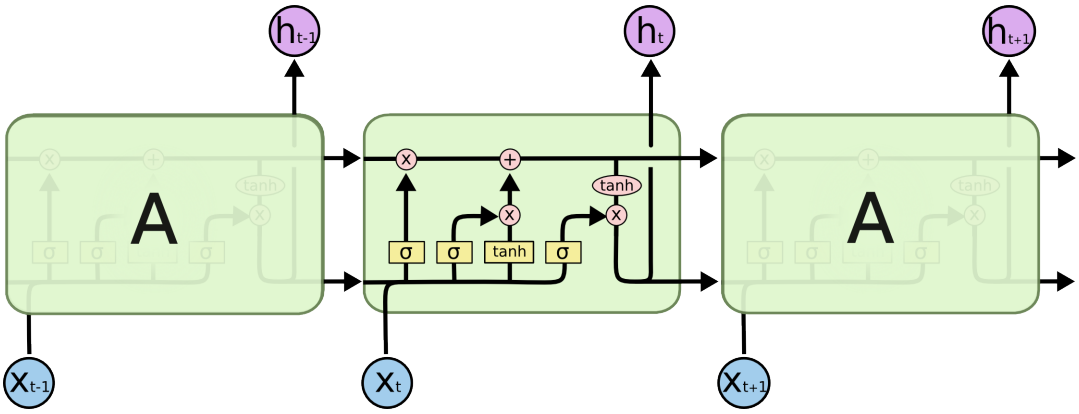


Figura 3.6: Schema di una cella LSTM. Ogni blocco giallo rappresenta un livello fully-connected con attivazione sigmoide ($\sigma$) o tanh

Come si può vedere in Figura 3.6, la struttura di una cella LSTM è ben più complessa di quella di una normale RNN. Infatti questa una cella LSTM contiene tre *gate*, cioè tre meccanismi che decidono se e come l'elemento $x_t$ della sequenza può essere utile ai fini della predizione:

- il *forget gate* $f_t$, che decide cosa dimenticare di quanto analizzato precedentemente ($c_{t-1}$). Se prendiamo la frase *A metà del XVIII secolo, Luigi XVI sono stati re di Francia* con il task di predire se soggetto e verbo sono concordi (in questo caso, ovviamente no), la prima parte può essere tralasciata in quanto non rilevante.
- l'*input gate* $i_t$ che decide quanto il nuovo input $x_t$ può essere rilevante;
- l'*output gate* $o_t$, che decide se la nuova memoria a lungo termine ($c_t$) può essere rilevante a breve termine ($h_t$) per l'uscita.

Ognuno di questi gate è un livello fully-connected con propria matrice di pesi, vettore di bias e funzione di attivazione. La computazione di una cella LSTM di dimensione $n$ è quindi data da:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.6}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i \tag{3.7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.8}$$

$$\hat{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3.9}$$

$$c_t = i_t * \hat{c}_t + f_t * c_{t-1} \tag{3.10}$$

$$h_t = \tanh(c_t) * o_t \tag{3.11}$$

in cui $[h_{t-1}, x_t]$ indica la concatenazione dei due vettori, il simbolo $*$ indica la moltiplicazione elemento per elemento, $W_f$, $W_i$, $W_o$, $W_c \in \mathbb{R}^{(n+d)\times n}$ sono le matrici di pesi e $b_f$, $b_i$, $b_o$, $b_c \in \mathbb{R}^n$ i vettori di bias e $\sigma$ è la funzione sigmoide. Il vettore $\hat{c}_t$ (Equazione 3.9) è l'elemento candidato a rappresentare la sequenza dopo aver analizzato $x_t$. A differenza delle normali RNN, questo però viene moltiplicato per i gate in modo da stabilire quanto ci sia di rilevante in esso (Equazione 3.10) e producendo il vettore $c_t$ che è la nuova *memoria a lungo termine* della LSTM. L'uscita $h_t$ (*memoria a breve termine)* è ulteriormente sottoposta all'output gate $o_t$ (Equazione 3.11).

Accanto alla LSTM, la Gated Recurrent Unit (GRU) [13] è un altro tipo di rete ricorrente che utilizza un singolo *update gate* $z_t$ che decide quanto mantenere dalla computazione di $x_t$. Una cella GRU calcola un candidato a sostituire $c_t$ allo stesso modo di $\hat{c}_t$ e successivamente decide cosa mantenere:

$$c_t = z_t * \hat{c_{t-1}} + (1 - z_t) * c_t \tag{3.12}$$

Diversamente dalle LSTM, nelle GRU non c'è differenza tra memoria a lungo termine o a breve, per cui $c_t$ è l'unico elemento passato alla cella successiva e in uscita.

In entrambi i casi, comunque, l'addestramento di una rete ricorrente richiede una notevole quantità di dati etichettati, non sempre disponibili specie in lingue diverse dall'inglese o in settori specifici. Inoltre, al fine di ottenere buone prestazioni è necessaria una ricerca degli iperparametri migliori (ad esempio, variando il numero di neuroni o le epoche di addestramento), addestrando diversi modelli basati sulla stessa architettura. Piccole modifiche nell'input, come l'aggiunta di nuova informazione, può inoltre richiedere una nuova ricerca, rendendo il processo di scelta del modello migliore molto dispendioso in termini di tempo.

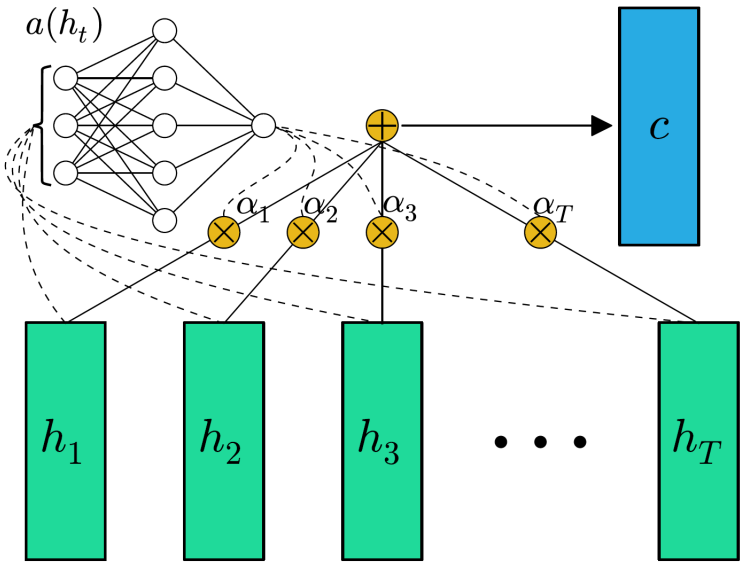


Figura 3.7: Schematizzazione dell'Attention Mechanism. I vettori $h1$, $h2$... $h_T$ sono le uscite della rete ricorrente. Figura tratta da [76].

## 3.4 Gli attention mechanism

Nonostante i miglioramenti dovuti all'utilizzo di celle LSTM, la capacità di una rete ricorrente di tenere una memoria a lungo termine, e quindi di poter collegare elementi della sequenza molto distanti tra loro, è comunque limitata [76]. Per questo motivo, in un task come la classificazione di un documento di $K$ parole $w_t$ con $t \in [1, K]$, l'ultima uscita della rete $h_K$ può non essere influenzata dalle prime parole e troppo dipendente dalle ultime.

Per risolvere questo problema, sono stati sviluppati i cosiddetti Attention Mechanism. Il primo e più conosciuto Attention Mechanism [4, 76] prende in considerazione ogni uscita $h_t$ di una LSTM, corrispondente alla computazione della sequenza fino a $w_t$, e calcola dei pesi $\alpha_t$ che rappresentano il contributo di $w_t$ alla predizione:

$$u_t = a(h_t) \tag{3.13}$$

$$\alpha_t = softmax(u_t) = \frac{exp(u_t)}{\sum_{k=1}^{K} exp(u_k)} \tag{3.14}$$

I pesi sono calcolati applicando la funzione softmax a $u_t$ (Equazione 3.14), ovvero a una quantità che rappresenta la relazione tra l'output finale e l'elaborazione della LSTM. Questa relazione è rappresentata dalla funzione $a$ ed è inizialmente sconosciuta, per cui viene creato un livello di rete neurale (tipicamente con attivazione tanh) per impararla tramite il meccanismo di backpropagation. Successivamente, l'Attention Mechanism calcola una rappresentazione finale del documento (*context vector)* come la media pesata dell'uscita delle celle LSTM:

$$c = \sum_{t=1}^{K} \alpha_t h_t \tag{3.15}$$

In Figura 3.7 è mostrata una schematizzazione dell'attention mechanism. Come si può vedere, a ogni elemento della sequenza in uscita dalla rete ricorrente $h_t$ è associato un peso $\alpha_t$. Il calcolo di $\alpha_t$ descritto nell'Equazione 3.14 è reso in forma grafica dalla rete neurale in alto a destra. Il nodo somma rappresenta la computazione del context vector come media pesata.

Unendo la rete ricorrente all'attention mechanism, si passa da un insieme di vettori rappresentanti l'input al vettore finale $c$ che ne rappresenta il significato. Per questo motivo, una rete neurale formata da LSTM e Attention può essere vista come un *encoder*, che codifica in termini numerici un documento o una frase. A questo punto, questa rappresentazione può essere collegata a un livello fully-connected con attivazione softmax per un task di classificazione o di Relation Extraction oppure rielaborata da un *decoder* che, ad esempio, la può tradurre in un'altra lingua.

La variante **Context Attention** introdotta in [101] inserisce un ulteriore vettore di pesi $v \in \mathbb{R}^n$ da moltiplicare con $u_t$ prima dell'esecuzione della softmax.

Il **Self-Interaction Attention Mechanism** [108] opera in maniera ancora più complessa, usando (analogamente al Context Attention) un vettore di pesi $v_i$ che è tuttavia diverso per ogni parola $w_i$. In questo modo, è pos-

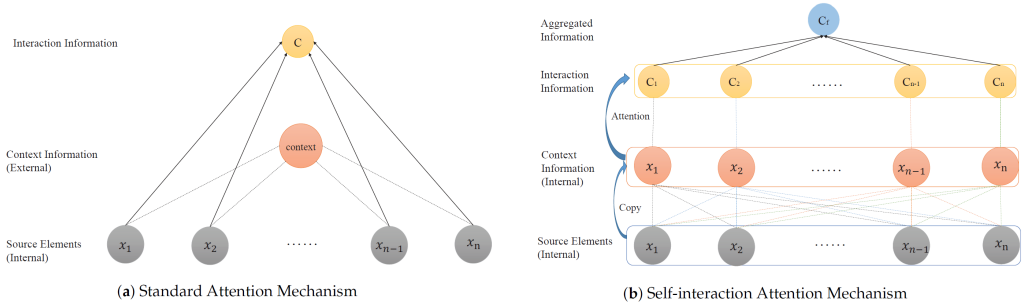


Figura 3.8: Differenza tra Attention standard e Self-Interaction Attention. Figura tratta da [108].

sibile calcolare l'interazione tra parola e il resto della frase, cioè ogni parola $w_k$ con $k \in [1, N]$:

$$\alpha_{i,k} = softmax(u_k^T v_i) \tag{3.16}$$

Nella Figura 3.8 è possibile vedere la differenza, in maniera molto schematica, tra l'Attention standard descritto precedentemente (a sinistra) e il Self-Interaction Attention (a destra). Come si può vedere, mentre a sinistra viene prodotto un singolo context vector, a destra ogni parola $x_i$ viene associata con tutte le altre e viene prodotto un peso per ciascuna coppia. Viene quindi calcolato un context vector per ogni parola, rappresentando come questa interagisce con tutte le altre all'interno della frase:

$$c_i = \sum_{k=1}^{K} \alpha_{i,k} h_i \tag{3.17}$$

Eventualmente si può effettuare un'operazione di pooling per ricavare poi un context vector finale, indicato nella figura come $c_f$.

### 3.4.1 Self-Attention

Anche il meccanismo Self-Attention [92] calcola l'interazione tra tutte le possibili parole, tuttavia ha scopi diversi e complessità ancora maggiore. L'idea infatti è che sia sufficiente creare un modello composto da soli attention

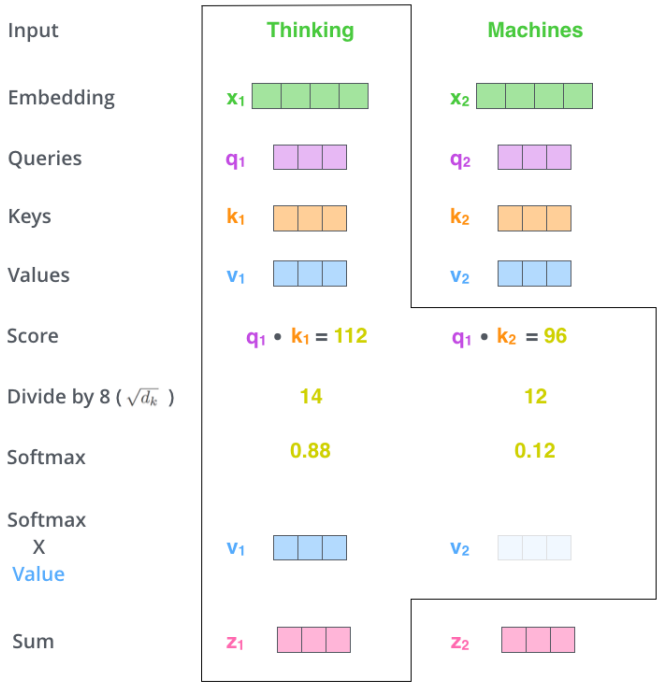


Figura 3.9: Procedimento del meccanismo di self-attention. In questo esempio, a scopo puramente illustrativo e per semplicità del calcolo, la dimensione iniziale del vettore $d_k$ è fissata a 64. In applicazioni reali, la dimensione è spesso fissata a 512 o 768.

e livelli fully-connected per poter comprendere il significato di una frase o di un documento, senza aver bisogno di una rete ricorrente.

Per cui, il Self-Attention opera direttamente sul word embedding $x_t \in \mathbb{R}^d$ applicando tre matrici di pesi $W_q,\ W_k,\ W_v \in \mathbb{R}^{d \times b}$ per ottenere tre rappresentazioni diverse della stessa parola (*query*, *key* e *value*):

$$q_t = W_q x_t,\ k_t = W_k x_t,\ v_t = W_v x_t \tag{3.18}$$

La dimensione $d$ del word embedding iniziale e quella delle nuove rappresentazioni ($b$) sono a discrezione dell'utente. Molto spesso, $b$ viene fissato uguale a $d$, come in [92] in cui viene scelto il valore 512. A questo punto, per ogni $i \in [1, N]$, possiamo calcolare l'influenza di ogni parola $x_i$ su $x_t$:

$$\alpha_{i_t} = softmax\left(\frac{q_t k_i^T}{\sqrt{d}}\right) \tag{3.19}$$

La nuova rappresentazione di $x_t$, che chiamiamo $z_t \in \mathbb{R}^b$, dopo l'applicazione del meccanismo di self-attention che tiene quindi conto di tutto il resto del documento pesandone le parti più o meno importanti ai fini di definire meglio il significato di $x_t$, è data dalla media pesata delle rappresentazioni *value*:

$$z_t = \sum_{i=1}^{N} \alpha_{i_t} v_t \tag{3.20}$$

Considerando solamente due parole, *Thinking* e *Machines*, la Figura 3.9 (tratta, come le seguenti in questa sezione e nella prossima, dall'articolo divulgativo in [1]), mostra passo dopo passo la computazione del self-attention per la prima delle due. Dopo aver calcolato le rappresentazioni query ($q_1$ e $q_2$), key ($k_1$ e $k_2$) e value ($v_1$ e $v_2$), vengono calcolati due score come prodotto scalare tra le rappresentazioni query e key. Questi vengono successivamente divisi per 8, ovvero la radice quadrata della lunghezza della rappresentazione originale, in questo caso, 64 (un numero totalmente arbitrario, scelto principalmente per semplicità di calcolo, i valori standard sono 512 o 768). Ai due score viene poi applicata la funzione softmax (vedasi Equazione 3.19) e il risultato viene moltiplicato per $v_1$ e $v_2$ che, sommati, fanno $z_1$.

### 3.4.2 Transformer e BERT

Il modello Transformer [92] e le architetture da esso derivate come BERT [103], che trattano il linguaggio senza l'ausilio di reti ricorrenti, hanno raggiunto risultati che sono il nuovo stato dell'arte per i principali task di Natural Language Processing. In questi modelli molto complessi, un solo self-attention non è sufficiente. Per questo motivo vengono messi in parallelo un numero arbitrario $m$ di questi meccanismi (di solito 8, 12, 16 o 24), creando il cosiddetto **Multi-Head Self-Attention**. Alla fine di questo procedimento quindi un documento è rappresentato da $m$ matrici $Z^m \in \mathbb{R}^{N \times b}$ in cui ogni riga è una rappresentazione $z_t^m$ data dal m-esimo self-attention.

Nella Figura 3.10 è possibile vedere a grandi linee il funzionamento: ognuna delle 8 head opera in parallelo su un documento composto da solamente

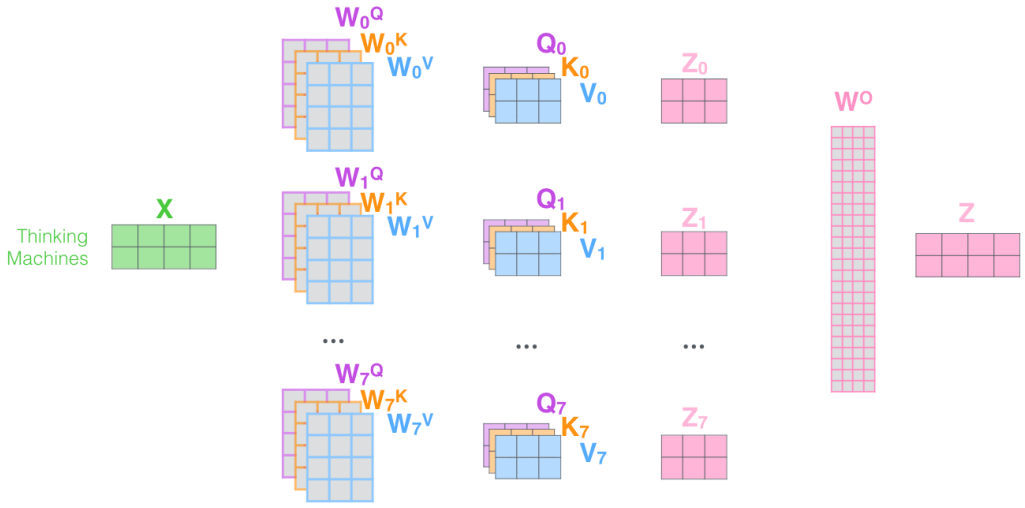


Figura 3.10: Schematizzazione del Multi-Head Self-Attention. Ogni head produce le proprie rappresentazione query, key e value e trasforma $x$ in una propria versione $z_i$. Queste poi vengono concatenate e moltiplicate per la matrice $W^0$ per trovare quindi la rappresentazione finale $z$.

due parole, producendo una loro rappresentazione resa graficamente dalle matrici $Z^0$, $Z^1$ ecc. che infatti possiedono due righe ciascuna. La rappresentazione finale $Z$ del multi-head attention è data dalla concatenazione delle singole matrici $Z^i$ e dalla moltiplicazione con una matrice di pesi $W^0$.

Il Multi-Head-Attention è solo la prima parte di un blocco del modello Transformer. Una rappresentazione grafica dell'intero blocco è visibile in Figura 3.11. Infatti, questo comprende anche la somma tra la vecchia rappresentazione $X$ e quella nuova $Z$ e la seguente normalizzazione. Successivamente, il risultato di questa operazione, che chiamiamo per comodità $\hat{Z}$, diviene l'input di un livello feedforward. Il risultato di questo livello viene risommato a $\hat{Z}$, finalmente concludendo l'esecuzione del blocco, che ha di fatto elaborato e rielaborato la rappresentazione delle parole tenendo conto più volte del contesto del documento.

Ognuno di questi blocchi compie un passo al fine di comprendere il significato del documento, per cui l'intero modello si compone di una serie di blocchi (da un minimo di 4 a un massimo di 32) per arrivare al risultato finale, cioè un vettore di numeri che rappresenti il documento. Questo encoder può

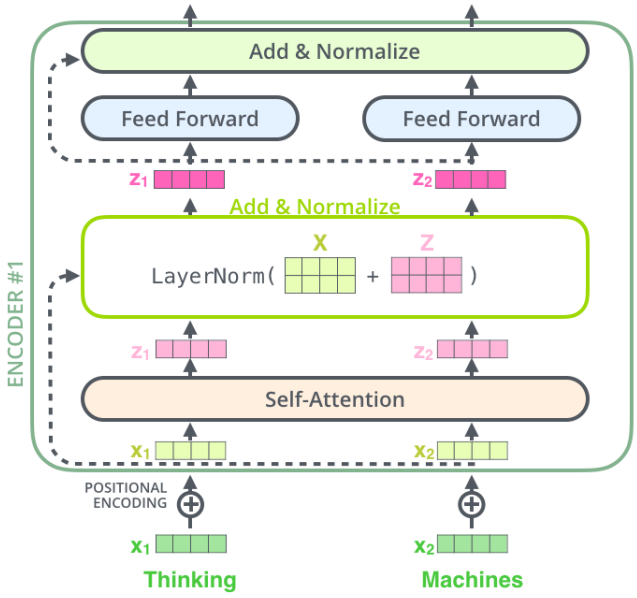


Figura 3.11: Singolo blocco di un modello Transformer, composto dal Multi-Head-Attention seguito da operazioni di somma e normalizzazione e da un livello feedforward.

essere collegato a livelli fully-connected per la classificazione oppure essere seguito da un decoder fatto alla stessa maniera che, dal vettore di numeri, ricostruisca il documento in un'altra lingua.

Come si può comprendere, l'addestramento di uno di questi modelli è estremamente complicato e richiede capacità, risorse computazionali e quantità di dati molto elevate e decisamente superiori rispetto a quelle per l'addestramento di un modello basato su reti ricorrenti. Per questo motivo, la pratica comune è reperire un modello pre-allenato e adattarlo, tramite il processo di *fine tuning*, ai propri scopi. Tuttavia, anche quest'ultima fase può risultare molto dispendiosa.

I modelli basati su Transformer hanno ottenuto risultati considerevoli in NER, Relation Extraction e classificazione. In particolare, uno dei più famosi e utilizzati è BERT [103] (Bidirectional Encoder Representations from Transformer) anche grazie al suo particolare training che comprende due task distinti:

- Il **Mask Language Model**. Data una frase, ne vengono opportunamente *mascherati* (ovvero sostituiti con un simbolo speciale che denota l'assenza di informazione) alcuni token. Il compito del modello è riuscire a predire correttamente questi token, confrontandosi con la versione della frase senza mascheramento, in un procedimento simile a quello di Word2Vec.

- Il **Next Sentence Prediction**. Per adattare ancora meglio BERT ai task a doppia sequenza come il Question Answering, viene creato un dataset formato da coppie di frasi con un'etichetta binaria che rappresenta se le due frasi appaiono consecutivamente in uno dei documenti del corpus.

Mentre la versione originale di BERT è allenata con documenti generici, come articoli di giornale o pagine di Wikipedia, ne esistono anche versioni più specifiche per il linguaggio scientifico o biomedico, come BioBERT [46].

### 3.4.3 Interpretabilità dell'Attention

Con lo sviluppo di modelli basati su reti ricorrenti e attention mechanism, la capacità di quest'ultimo di porre maggiore rilievo su alcune parti dell'input è stata vista come un grande passo avanti nell'interpretabilità del deep learning nel NLP. Questo anche a causa della difficoltà di applicare metodi classici come la gradient-based explanation [31], che studia la variazione del gradiente utilizzato durante la backpropagation in base all'input, o LIME (Local Interpretable Model-Agnostic Explanation) [80], che controlla come la perturbazione dell'input (come rimuovere una o più parole) cambi il risultato del modello.

La visualizzazione dell'attention mechanism è stata studiata e applicata, sia per le RNN [52] che per i modelli basati su Transformer [94], per comprendere il funzionamento interno del modello, i suoi errori e i suoi punti di forza [47]. Tuttavia, è ancora oggetto di discussione se il risultato dell'attention mechanism sia veramente attendibile e possa essere davvero utile ai fini dell'interpretabilità. L'articolo "Attention is not explanation" di

Jain e Wallace [38] utilizzando diversi dataset e task di NLP (classificazione binaria, question answering, riconoscimento di contraddizioni), ha infatti mostrato come si possa costruire una distribuzione alternativa o addirittura una contro-distribuzione (cioè una che differisce fortemente da quella iniziale) dei pesi $\alpha_t$ dell'attention mechanism in modo tale che il sistema, pure "concentrando l'attenzione" su altre sezioni del documento fornisse la stessa predizione. Chiaramente, se il sistema fornisce la stessa predizione pur basandosi su sezioni del documento di significato molto diverso, la distribuzione dell'attention non può essere utilizzata per un'interpretazione plausibile.

Tuttavia, l'articolo stesso fa notare che i dataset biomedici coinvolti nell'esperimento *Twitter Adverse Drug Reaction*, *MIMIC ICD9 (Diabetes)* e *MIMIC ICD9 (Anemia)* hanno un comportamento diverso. Nei primi due, ad esempio, i pochi token che indicano la classe positiva sono riconosciuti dall'attention e questa spiegazione concorda con altre forme di interpretazione utilizzate. Inoltre, per tutti e tre i dataset, perturbare la distribuzione dell'attention porta a grandi modifiche nella predizione.

Altri difetti di questo approccio sono stati rilevati dall'articolo "Attention is not not explanation" di Wiegreffe e Pinter [98], sia per quanto riguarda come la contro-distribuzione è stata costruita, sia per quanto riguarda dataset e task scelti. In diversi di questi, infatti, l'introduzione dell'attention non porta ad alcun beneficio nel modello che otterrebbe gli stessi risultati con solamente un livello LSTM. Per questo motivo quindi non ci si può attendere che i pesi dell'attention mechanism siano allenati in modo tale da risultare utili al task e quindi all'interpretazione [91]. Gli autori di questo articolo quindi, sulla base dei loro esperimenti, riaffermano come l'attention mechanism possa indicare l'importanza dei token ai fini della predizione.

L'articolo in [91] divide il problema dell'interpretabilità dell'attention in due casi distinti:

- i task a singola sequenza, cioè quelli che formulano una predizione rispetto a una singola frase o a un documento come la classificazione o la relation extraction;

- i task a doppia sequenza che prevedono due input: nel caso del question

answering, ad esempio, oltre alla domanda, formulata in linguaggio naturale, il sistema riceve anche il documento da cui estrarre la risposta.

I loro esperimenti di alterazione della distribuzione dei pesi mostrano come l'attention sia più rappresentativo nel secondo caso mentre, per i task a sequenza singola, abbia una funzione più che altro di *gate*, cioè possa rappresentare semplicemente quali input possano essere considerati e quali no. Mentre questo può non essere sufficiente per l'interpretabilità tout court, a nostro giudizio può risultare comunque utile per alcuni task e alcune categorie di input [37].

Va inoltre ricordato come, sebbene venga spesso utilizzato per capire l'importanza dei singoli token, l'attention mechanism lavori non direttamente sui word vectors ma sugli output della LSTM [69] che, come già detto in questo capitolo, dipendono anche da token precedenti e, nel caso bidirezionale, perfino dai seguenti. Parte dell'informazione contenuta in un token la cui uscita della corrispondente cella LSTM è stata resa poco importante dalla perturbazione dei pesi può comunque essere sopravvissuta nelle celle adiacenti. Gli esperimenti di [69] e [98] mostrano come, in assenza di reti ricorrenti, le prestazioni si abbassano drasticamente alterando l'attention.

Per quanto riguarda invece i modelli basati quasi solo esclusivamente sull'attention come Transformer o BERT, l'analisi di come operano internamente è ancora agli albori con risultati interessanti e da approfondire [87].

# Capitolo 4

# Classificazione di referti radiologici

In questo capitolo verrà descritto come sono state applicate le tecniche di deep learning e NLP su un problema di analisi di testi clinici in lingua italiana: la classificazione gerarchica di referti radiologici.

Questo progetto è iniziato nel 2015 come collaborazione tra l'*Università degli Studi di Brescia*, la *Fondazione Bruno Kessler* e gli *Spedali Civili di Brescia* e parte dalla considerazione che ogni giorno, negli ospedali, vengono raccolti, sotto forma di testo clinico, grandi quantità di dati che possono essere impiegati per estrarre conoscenza utile a fini logistici, di organizzazione ospedaliera, di raccolta di casi clinici a fini di ricerca e didattica ecc. Questa mole di dati totalmente non strutturati, tuttavia, per essere sfruttata dagli esseri umani richiede notevoli quantità di tempo e di esperienza, per cui si sono studiate tecniche per rendere disponibile queste informazioni in forma aggregata attraverso l'analisi del linguaggio e l'apprendimento supervisionato.

## 4.1 Descrizione dei dati

Un referto è un documento testuale prodotto dall'analisi di un'immagine ottenuta mediante l'esecuzione di un esame radiologico come la Tomogra-

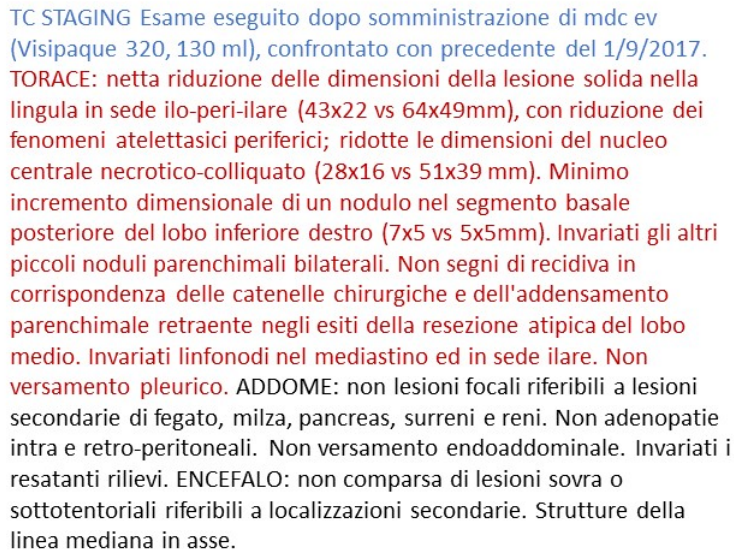
TC STAGING Esame eseguito dopo somministrazione di mdc ev (Visipaque 320, 130 ml), confrontato con precedente del 1/9/2017. TORACE: netta riduzione delle dimensioni della lesione solida nella lingula in sede ilo-peri-ilare (43x22 vs 64x49mm), con riduzione dei fenomeni atelettasici periferici; ridotte le dimensioni del nucleo centrale necrotico-colliquato (28x16 vs 51x39 mm). Minimo incremento dimensionale di un nodulo nel segmento basale posteriore del lobo inferiore destro (7x5 vs 5x5mm). Invariati gli altri piccoli noduli parenchimali bilaterali. Non segni di recidiva in corrispondenza delle catenelle chirurgiche e dell'addensamento parenchimale retraente negli esiti della resezione atipica del lobo medio. Invariati linfonodi nel mediastino ed in sede ilare. Non versamento pleurico. ADDOME: non lesioni focali riferibili a lesioni secondarie di fegato, milza, pancreas, surreni e reni. Non adenopatie intra e retro-peritoneali. Non versamento endoaddominale. Invariati i resatanti rilievi. ENCEFALO: non comparsa di lesioni sovra o sottotentoriali riferibili a localizzazioni secondarie. Strutture della linea mediana in asse.

Figura 4.1: Esempio di referto radiologico. In blu, l'intestazione del referto, in rosso la sezione relativa al torace, oggetto della nostra analisi. In nero, le parti non considerate dalla nostra applicazione.

fia Computerizzata (TC) o la Radiografia (RX). Sebbene fondamentalmente un referto si componga di testo libero, ci sono alcune linee guida per descrivere i rilievi riscontrati nell'immagine in modo tale che il referto possa essere utilizzato come strumento di comunicazione tra un medico e l'altro e, documentando effettivamente lo stato del paziente, per confronti a distanza di tempo. Una di queste linee guida è la suddivisione in sezioni standard, ognuna dedicata a un settore del corpo umano quali il torace, l'addome o l'encefalo.

In questa applicazione [24, 25, 72], i referti analizzati provengono direttamente dal Reparto di Radiologia Diagnostica 2 diretto dal prof. Roberto Maroldi e il focus dell'analisi è sui rilievi riscontrati nel torace e sulla neoplasia polmonare. In Figura 4.1 mostriamo un esempio di referto radiologico in cui è possibile notare alcune caratteristiche salienti del testo clinico e del referto radiologico (confronta anche la Sezione 2.2) quali:

- la suddivisione in sezioni: l'intestazione del referto che solitamente contiene dettagli su come l'esame è stato eseguito (TC, RX, con o senza

mezzo di contrasto e di che tipo, eventuali precedenti per confronto) e le sezioni relative al torace, addome e encefalo segnalate dalla presenza del nome in maiuscolo e dai due punti;

- la scarna struttura della frase: ad esempio in "*Invariati linfonodi nel mediastino ed in sede ilare*" non c'è predicato verbale e non c'è l'articolo prima di linfonodi;
- acronomi e convenzioni, come ad esempio "*mdc ev*" che sta per mezzo di contrasto per somministrazione endovenosa;
- difficoltà di tokenizzazione e incongruenze, ad esempio la parola "*ilo-peri-ilare*" (divisibile in tre token, visti i due trattini) in altri referti può essere anche scritta come "*ilo-perilare*" (divisibile in due token); altre incongruenze possono essere presenti nelle dimensioni di noduli lesioni, ad esempio in "*43x22 vs 64x49mm*" l'unità di misura è presente solo nella seconda dimensione;
- il linguaggio tecnico e fortemente specializzato, con termini come "*atelettastici*" e "*necrotico-colliquato*" che difficilmente possono essere riscontrati all'interno dei corpus generici con cui di solito si addestrano i modelli di Machine Learning.

### 4.1.1 Schema di classificazione

La nostra applicazione si propone di classificare i referti secondo diversi aspetti rilevanti, considerando diverse caratteristiche dell'esame fatto e dei rilievi riscontrati. Per questo motivo, non è stata definita una classificazione semplice, come ad esempio la scelta di un valore tra un insieme di opzioni, ma, con l'ausilio dei radiologi degli *Spedali Civili di Brescia*, una classificazione gerarchica fatta da più livelli in relazione tra loro.

Come si può vedere in Figura 4.2, lo schema di classificazione è composto da quattro livelli:

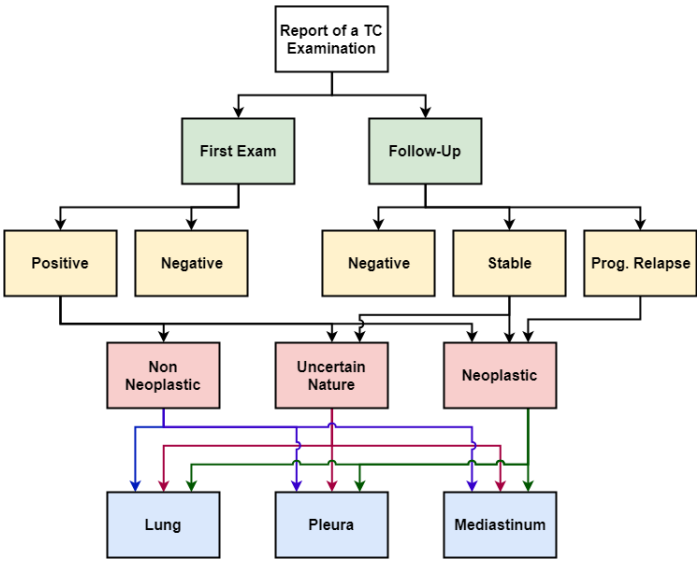


Figura 4.2: Schema di classificazione per referti radiologici, composto da quattro livelli. Le frecce indicano il valore che un livello può assumere, dato quello precedente.

1. **Tipo Esame**, che può assumere i valori *Primo Esame*, nel caso il paziente sia stato esaminato per la prima volta, o *Follow-Up*, nel caso in cui il paziente sia stato richiamato per un ulteriore controllo.

2. **Risultato**, che offre una valutazione dello stato finale del paziente e che può assumere i valori *Positivo* o *Negativo* nel caso il referto sia un *Primo Esame*, oppure *Negativo*, *Stabile* o *Progressione Recidiva* nel caso il referto sia un *Follow-Up*.

3. **Natura Lesione** che valuta se le lesioni descritte nel referto siano di tipo *Neoplastico*, *Natura Dubbia* o *Non Neoplastico*.

4. **Sito Lesione**, che indica dove si trovano le lesioni riscontrate e che può assumere i valori *Polmone*, *Pleura* e *Mediastino*. Diversamente dai precedenti, questo livello può assumere anche più valori contemporaneamente.

Nella medesima figura si possono vedere anche delle frecce tra un livello e l'altro. Ogni freccia indica il valore che un referto può assumere, nel livello

successivo, dato il valore di quello precedente. Queste relazioni gerarchiche possono essere formalizzate attraverso le seguenti regole (anch'esse formulate dai radiologi):

1. Nel caso di un *Primo Esame*, se viene riscontrato almeno un sospetto di lesione neoplastica (*Neoplastico* o *Natura Dubbia*), il risultato dev'essere obbligatoriamente *Positivo*.

2. Un *Primo Esame* può essere classificato *Positivo* anche se non c'è sospetto di lesione neoplastica, nel caso di presenza di polmonite, embolia polmonare o altre patologie simili. Chiamiamo questi casi i **Positivi Non Neoplastici**.

3. Dato che i *Follow-Up* sono prescritti esclusivamente per monitorare le condizioni di pazienti neoplastici, se nel referto non viene riscontrato almeno un sospetto di lesione neoplastica, il risultato deve assumere il valore *Negativo*. Questo vale anche per i casi di polmonite e embolia polmonare.

4. I referti classificati con risultato *Progressione Recidiva*, possono esclusivamente avere come Natura Lesione il valore *Neoplastico*.

## 4.2 Il sistema basato sulle annotazioni

Il primo sistema progettato (prima dell'inizio della mia attività di dottorato) per la classificazione gerarchica di referti radiologici prevedeva l'utilizzo di annotazioni manuali ed è stato approfonditamente descritto in [24, 25].

Questa tecnica richiede che i radiologi, o in generale gli esperti del dominio dell'applicazione, oltre a classificare il documento ne annotino anche le parti più importanti in modo tale da "giustificare" la classificazione scelta. Dato che si tratta di una classificazione a più livelli, questo processo va svolto per ogni livello: ad esempio, prendendo il referto in Figura 4.1, l'espressione *confrontato con precedente del 1/9/2017* viene annotata in quanto utile per indicare che il referto è un *Follow-Up*, mentre invece *lesione solida nella lingula in sede ilo-peri-ilare* viene annotata per la Natura Lesione.

Chiaramente, se già di per sè la classificazione è un impegno gravoso, l'annotazione richiede ancora più tempo e sforzo da parte degli esperti del dominio. Per questo motivo, il dataset iniziale del sistema basato sulle annotazioni (che d'ora in avanti verrà chiamato *dataset annotato*) è composto solamente di 346 referti selezionati con l'ausilio dei radiologi, di cui 278 utilizzati per l'addestramento degli algoritmi di machine learning e 68 per la fase di testing.

Dato un referto appartenente al test set (quindi non annotato), il sistema prevedeva i seguenti passi:

1. **La fase di pre-processing** con l'ausilio del tool di NLP per la lingua italiana TextPro [67] che comprende l'estrazione dei lemmi, l'analisi morfologica, il Part-of-Speech tagging, l'identificazione di espressioni temporali e numeriche e il riconoscimento di negazioni. Questo tool rappresenta quindi una parola tramite una serie di feature testuali (il lemma, il genere, il numero, la presenza di negazioni, il POS-tag ecc.).

2. **L'annotazione automatica**, con l'algoritmo Conditional Random Fields (CRF) [44] addestrato con le annotazioni manuali dei 296 referti di training. Questo algoritmo prende in input le feature prodotte da TextPro e classifica ogni token come appartenente o meno a un'espressione rilevante per la classificazione del referto. Per ogni livello di classificazione, viene quindi addestrato un modello CRF con il compito di evidenziare automaticamente le espressioni rilevanti per quel particolare livello. Il training set di questi modelli sono le annotazioni manuali del livello considerato.

3. **La classificazione delle annotazioni**: una volta evidenziate le parti salienti di un referto per la classificazione di un livello, queste vengono rappresentate tramite il modello Bag-of-words ed elaborate da un algoritmo di Machine Learning (Decision Tree, Support Vector Machine, Multi-layer Perceptron, Naive Bayes e Random Forest) con il compito di decidere quale valore del livello assegnare all'annotazione. Ad esempio, l'espressione *lesione solida nella lingula in sede ilo-peri-ilare* rilevante per la Natura Lesione viene classificata con il valore *Neoplastico*. Nella costruzione del training set, tutte le espressioni annotate per un livello ottengono il valore di classificazione del referto per quel livello.

4. **La derivazione della classificazione del referto**. Date quindi una serie di annotazioni, ognuna con la propria classificazione, una serie di regole permettono di derivare il valore complessivo del referto. Ad esempio, se nel referto sono presenti al terzo livello un'annotazione di valore *Neoplastico* e una di valore *Natura Dubbia*, si considera il caso peggiore e quindi il valore finale del referto per il livello Natura Lesione è *Neoplastico*.

### 4.2.1 Problemi relativi al sistema

Nonostante i risultati iniziali fossero promettenti, una volta che il sistema è stato testato nel funzionamento quotidiano del reparto di Radiologia degli *Spedali Civili di Brescia*, le prestazioni sono state peggiori del previsto.

Ad un'analisi più dettagliata [72], condotta all'inizio del mio dottorato, sono stati riscontrati i seguenti problemi:

- Data la difficoltà e la quantità di tempo richiesta per l'annotazione di un referto, la selezione di 346 referti annotati da un solo radiologo non è sufficiente per cogliere tutti gli aspetti del problema, che richiederebbe un dataset più grande e soprattutto annotato da più radiologi.

- Dato che il paziente ha la possibilità di leggere il referto medico relativo al proprio esame, il medico tende a scrivere il testo del referto in modo prettamente tecnico, annotando i rilievi riscontrati ma spesso senza formulare una diagnosi netta.

- L'incompletezza dell'annotazione del referto, ovvero l'esperto ha la tendenza ad annotare esclusivamente quanto basta per giustificare la classificazione, tralasciando altre parti di testo che avrebbero potuto portare alla medesima conclusione. Ad esempio, sempre facendo riferimento al referto nella Figura 4.1, sia la *lesione solida nella lingula in sede iloperi-ilare* che l'*incremento dimensionale di un nodulo* possono essere importanti per l'attribuzione del valore *Neoplastico* al livello Natura Lesione, eppure in molti casi solo una delle due viene annotata (molto spesso, la prima che compare nel testo). Questo fa sì che in un refer-

to non annotato, la seconda espressione possa non venire riconosciuta come rilevante.

- L'incompletezza delle espressioni annotate, ovvero il fatto che un'espressione che descrive un rilievo radiologico, e quindi un concetto importante per l'analisi del referto, non viene annotata completamente. Ad esempio, mentre l'espressione *nodulo a margini netti* è sovente l'esempio di una lesione di tipo *Non Neoplastico* o di un referto *Negativo*, l'espressione *nodulo a margini irregolari e spiculati* o il fatto che le sue dimensioni siano aumentate possono invece indicare la presenza di una lesione di tipo *Neoplastico*. Molto spesso, tuttavia, l'annotazione manuale riguarda esclusivamente il termine *nodulo* escludendo le sue caratteristiche. Questo significa avere la stessa espressione classificata con valori diversi e quindi la presenza di incertezza nell'algoritmo di machine learning per la classificazione delle annotazioni.

- Questo è oltretutto aggravato dal fatto che l'algoritmo CRF per l'annotazione automatica ha una capacità limitata di riconoscere espressioni lunghe [44]. Per cui, mentre è sicuramente in grado identificare il rilievo (*nodulo*, *lesione*, ecc.), ha maggiori difficoltà nel collegarne le caratteristiche, seppure queste siano essenziali per l'analisi.

## 4.3 Il sistema basato su deep learning

La fase di test del sistema basato sulle annotazioni prevedeva che, subito dopo la scrittura del referto, il sistema fornisse la classificazione automatica in pochi secondi e che il radiologo validasse questo risultato apportando le eventuali modifiche con l'ausilio di un'interfaccia grafica. Dal novembre 2017 al gennaio 2019 quindi sono stati classificati dai radiologi, validando o correggendo la classificazione automatica fornita dal sistema, 5752 referti presi direttamente dall'attività quotidiana del reparto, il cosiddetto *dataset di produzione*. Questi nuovi documenti si vanno ad aggiungere ai 346 utilizzati per il sistema basato sulle annotazioni.

| | | Dataset Annotato | Dataset Produzione |
|---|---|---|---|
| Tipo Esame | Primo Esame | 136 | 2061 |
| | Follow-Up | 210 | 3691 |
| Ris. (P.E.) | Positivo | 120 | 798 |
| | Negativo | 16 | 1263 |
| Ris. (F.U.) | Prog. Rec. | 54 | 696 |
| | Stabile | 79 | 784 |
| | Negativo | 77 | 2211 |
| Natura Les. | Neoplastica | 161 | 1005 |
| | Natura Dub. | 29 | 609 |
| | Non Neopl. | 54 | 691 |
| Sito Les. | Polmone | 188 | 2042 |
| | Pleura | 19 | 152 |
| | Mediastino | 40 | 300 |

Tabella 4.1: Distribuzione dei referti nelle diverse classi dello schema per il dataset annotato (346 referti in totale) e per quello di produzione (5752 referti).

Questi nuovi referti non potevano essere tuttavia utilizzati per ampliare il training set del sistema basato sulle annotazioni manuali, in quanto presentano solo le etichette di classificazione poste dai medici ma non le annotazioni delle espressioni più rilevanti. Stanti anche i problemi relativi alle annotazioni descritti nella Sezione 4.2.1, si è deciso quindi di creare un nuovo sistema che non necessitasse di un procedimento lungo e costoso quale l'annotazione manuale e che prendesse in input direttamente il testo del referto. Per fare ciò, è stato necessario l'impiego di tecniche di deep learning quali il word embedding e le reti LSTM.

Nella Tabella 4.1, viene mostrata la distribuzione dei referti, per tutti i livelli e le classi definite nello schema di classificazione. Possiamo subito notare una significativa differenza: nel dataset annotato i referti negativi sono la minoranza (specialmente nel caso dei primi esami, con solo 16 esempi su 136), mentre sono oltre il 60% nel dataset di produzione. Anche per quanto riguarda il livello Natura Lesione, c'è una significativa differenza: nel dataset annotato i referti neoplastici sono oltre il 65% del totale, mentre in quello di produzione non raggiungono il 45%. Non c'è invece significativa differenza

tra i due dataset per quanto riguarda il numero di parole: la lunghezza media è di 120 parole e l'80% dei referti ne contiene meno di 250. Tuttavia, circa il 5% dei referti è molto più lungo, con 400 parole o più.

In questa sezione, presentiamo i componenti principali del nuovo sistema, che abbiamo descritto approfonditamente in [72], e come sono stati costruiti i modelli di deep learning che fornissero una classificazione gerarchica che rispettasse le indicazioni poste dai radiologi. Per l'addestramento del sistema di deep learning è stato utilizzato il dataset di produzione, sia perchè i valori di classificazione sono stati assegnati dagli stessi medici che andranno poi ad utilizzare il sistema, sia per la diversa distribuzione dei due dataset e, infine, per poter usufruire del dataset annotato come ulteriore test set, completamente indipendente dai referti usati per il training.

### 4.3.1 Pre-processing e rappresentazione dell'input

Per riuscire a passare dal testo libero del referto ad un input per una rete neurale, occorre svolgere una fase di pre-processing che si compone delle seguenti fasi:

1. **Identificazione delle sezioni**: dato che, come abbiamo visto nell'esempio in Figura 4.1, un referto si può comporre di diverse sezioni, ognuna dedicata a una specifica parte del corpo, e che la nostra applicazione si concentra solo sul torace, la prima fase riguarda l'individuazione delle sezioni rilevanti. Con un algoritmo basato sulle espressioni regolari e progettato appositamente per la struttura dei referti, vengono selezionate: l'intestazione del referto, contenenti le specifiche di come è stato eseguito e con cosa viene eventualmente confrontato, la sezione relativa al torace o comunque ogni sezione non etichettata come appartenente ad un'altra parte del corpo, e le eventuali conclusioni (presenti tuttavia solo in una minoranza dei referti considerati).

2. **Suddivisione in frasi, tokenizzazione e POS-tagging**: successivamente, è stato utilizzato il tool di NLP spaCy[1] per procedere con

[1] https://spacy.io

la suddivisione del documento nelle singole frasi che lo compongono, con la tokenizzazione e il Part-of-Speech tagging. Non essendo disponibili moduli di NLP specializzati per testi clinici o biomedici in lingua italiana e non avendo a disposizione alcun testo con la suddivisione in token e il POS per allenare un nostro modello, è stato utilizzato "it_core_news_sm" realizzato per la lingua italiana e allenato con un corpus generico di articoli di giornale. Trattandosi di un modello generico, alcune parole ed espressioni relativa al gergo radiologico non vengono tokenizzate o etichettate con il PoS tag in modo corretto. Per questo motivo, al modello sono state aggiunte alcune espressioni regolari e associazioni tra parola e PoS per evitare errori in termini ritenuti fondamentali per la nostra applicazione. Ad esempio, alla parola *adenopatia* è stato associato il PoS *NOUN* in quanto il termine è un nome comune.

3. **Standardizzazione della lunghezza**: dato che la necessità pratica del funzionamento dei modelli di classificazione richiede un input di dimensione fissa, abbiamo definito un numero massimo di parole che un referto può contenere. Dato che nel nostro dataset il 95% dei referti contiene meno di 450 parole, abbiamo fissato la lunghezza massima a 450. Se un referto supera questa lunghezza, vengono rimossi articoli, congiunzioni e preposizioni fino a raggiungere il numero di parole desiderato. Qualora il referto, dopo questa procedura, ecceda ancora la lunghezza massima, vengono selezionate le prime 450 parole. Questo tuttavia accade solo nello 0.7% dei casi.

L'input dei nostri modelli di classificazione viene quindi costruito utilizzando un modello di word embedding. Non avendo a disposizione un modello pre-addestrato per i testi clinici in lingua italiana, ma essendo in possesso di un dataset composto da oltre 6.000 referti classificati e oltre 10.000 referti non classificati forniti dall'ospedale, abbiamo allenato un modello Word2Vec (nella versione Skip-Gram, che è quella considerata dalla libreria Gensim [2]) [57] che permettesse la rappresentazione delle parole come word vector nell'ambi-

[2]https://radimrehurek.com/gensim/

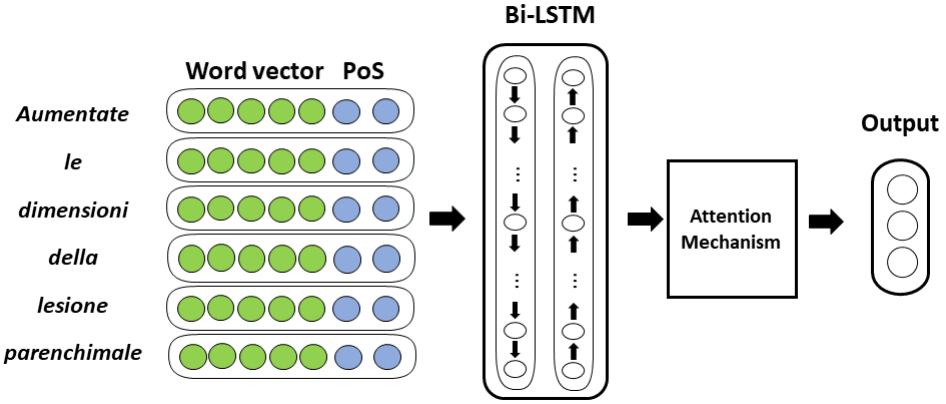


Figura 4.3: Blocco Classificatore. La matrice di word vectors e POS embedding viene elaborata dal livello bidirezionale LSTM e dall'Attention Mechanism. L'uscita è calcolata con un livello fully-connected.

to della radiologia in lingua italiana. Ogni parola viene quindi rappresentata come un vettore di 200 numeri reali. A questo vettore viene concatenato il POS vector, ovvero un vettore di 4 posizioni rappresentante il Part-of-Speech della parola. Il modello per il POS embedding è stato costruito esattamente come quello di word embedding utilizzando però la sequenza di POS, al posto della sequenza di parole, come esempi di training.

Per ogni referto quindi costruiamo una matrice di dimensione $450 \times 204$ in cui ad ogni parola, cioè ad ogni riga della matrice, corrisponde un word vector e un POS vector. Nel caso un referto sia composto da meno di 450 parole, lo spazio rimanente viene riempito da vettori di 204 zeri con la cosiddetta tecnica di *padding*.

### 4.3.2 Il Blocco Classificatore

Lo schema di classificazione proposto dai radiologi (Figura 4.2) propone una forma di analisi del referto che comprenda diversi aspetti e diversi concetti da estrarre dal testo, collegati tra loro tramite regole gerarchiche.

Per questo motivo, seguendo l'esempio del sistema basato sulle annotazioni e stante anche il numero relativamente basso di referti, si è deciso di non costruire un singolo modello che fornisse l'intera classificazione livello

per livello. Invece, si è deciso di costruire un modello fatto da *blocchi*, ognuno responsabile di catturare un concetto, da cui successivamente derivare la classificazione secondo lo schema.

Come visibile nella Figura 4.3, ognuno di questi blocchi classificatori è composto da:

- un livello LSTM bidirezionale (Sezione 3.3.2) che elabora l'intero referto nell'ordine originale delle parole e al contrario;
- un Attention Mechanism (Sezione 3.4) che assegna dei pesi a ognuno dei token che compongono il referto, potenzialmente in base alla loro importanza;
- un livello fully-connected, con attivazione softmax o sigmoide a seconda dei casi, responsabile di fornire in output la predizione finale.

Essendo la rappresentazione già costruita su referti radiologici dello stesso tipo, la matrice di word vector in input al blocco classificatore non è modificata durante il training del modello.

### 4.3.3 I modelli gerarchici

In questa sezione, presentiamo tre modelli, di crescente complessità, per la classificazione gerarchica di referti radiologi, composti tramite la combinazione di blocchi classificatori.

#### 4.3.3.1 Modello 1

Il Modello 1 (Figura 4.4) è il più semplice dei tre e unisce le due configurazioni del Risultato, quella per i *Follow-Up* e quella per i *Primo Esame*. Si compone di 3 blocchi, uno per livello:

- il **Blocco Tipo Esame** che identifica se un referto è *Primo Esame* o *Follow-Up*, e viene allenato utilizzando l'intero training set;
- il **Blocco Risultato Esame** che indica se il referto è *Negativo*, *Positivo*, *Stabile* o *Progressione Recidiva*, anche questo allenato con l'intero training set;

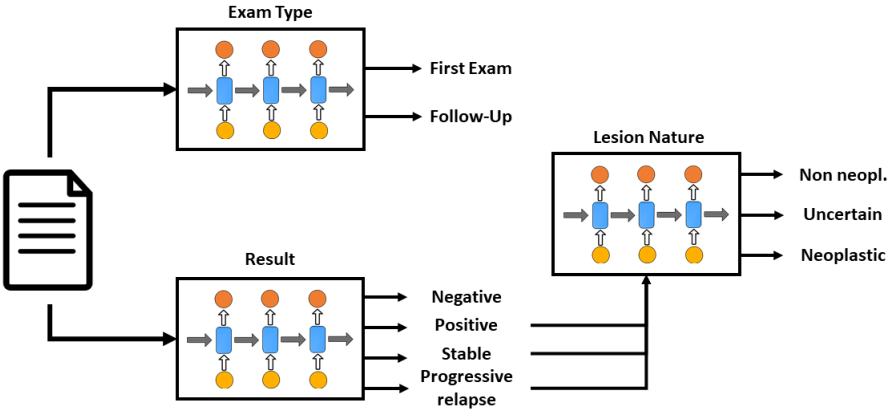


Figura 4.4: Modello 1. Ogni quadrato rappresenta un blocco classificatore. Due blocchi indipendenti classificano il Tipo Esame e il Risultato. I Non Negativi vengono elaborati dal blocco Natura Lesione.

- il **Blocco Natura Lesione**, che indica se il referto è *Non Neoplastico*, *Natura Dubbia* o *Neoplastico*, allenato solo con i referti di training il cui risultato non è *Negativo*. Se un referto di test viene classificato come *Positivo*, *Stabile* o *Progressione Recidiva*, allora viene analizzato anche da questo blocco.

Da notare come questo modello non rispetti la gerarchia, in quanto i primi due blocchi sono completamente indipendenti tra loro. Infatti, nulla vieta al modello di fornire una classificazione come *Follow-Up* per il livello Tipo Esame e *Positivo* per il livello Risultato, cosa che è impossibile secondo lo schema. Tuttavia, nella sua semplicità, il Modello 1 può essere facilmente implementabile e, per questo motivo, può anche essere considerato come punto di partenza con cui confrontare modelli più complessi. Inoltre, la sua conformazione può essere facilmente adattata per altri problemi basati su referti medici che non prevedono due risultati diversi se è stato svolto un primo esame oppure un follow-up.

#### 4.3.3.2 Modello 2

In accordo a quanto definito nello schema di classificazione (Figura 4.2), il Modello 2 divide la predizione del Risultato in due blocchi distinti (Figura

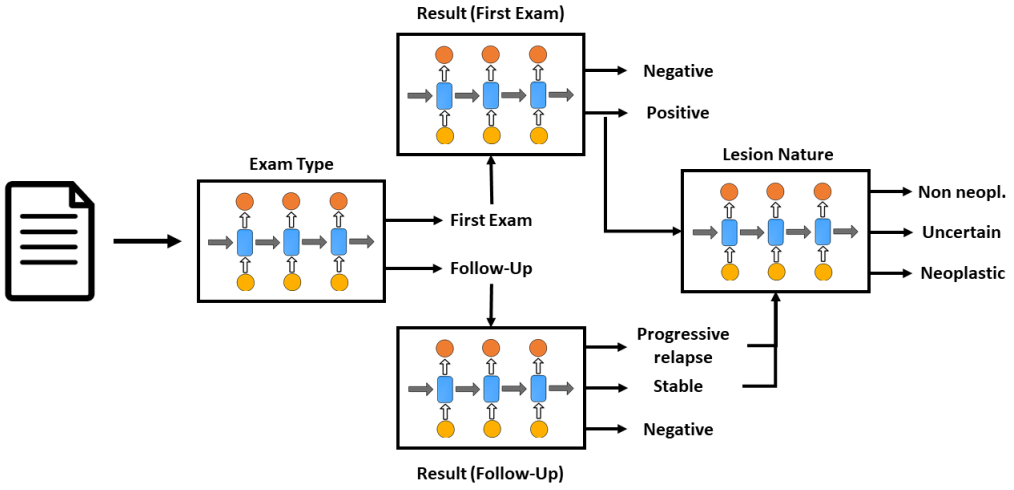


Figura 4.5: Modello 2. In questo modello, due blocchi diversi classificano il Risultato in base alla classificazione Blocco *Primo Esame*.

4.5):

- il **Blocco Primo Esame**, che indica se un referto è *Positivo* o *Negativo* ed è allenato solo con i *Primo Esame* del training set;
- il **Blocco *Follow-Up***, che indica se un referto è *Negativo*, *Stabile* o *Progressione Recidiva*, ed è allenato solo con i *Follow-Up* del training set.

Nella fase di test, se un referto è classificato come *Primo Esame* dal Blocco Tipo Esame, allora viene analizzato dal Blocco Primo Esame, altrimenti dal Blocco Follow-Up. Affinchè i referti vengano poi esaminati dal Blocco Natura Lesione, è necessario che siano etichettati come Positivi, Stabili o Progressione Recidiva.

In questo modello, contrariamente al Modello 1, la particolare conformazione dello schema è stata incorporata nella struttura, introducendo quindi un blocco in più. Un potenziale vantaggio di questo modello è che il Blocco Primo Esame e il Blocco Follow-Up possono specializzarsi. Se nel nostro caso non esistono particolari differenze tra i referti di una categoria e l'altra, in vista di un'applicazione in cui queste differenze sussistono il Modello 2 può essere una scelta adeguata.

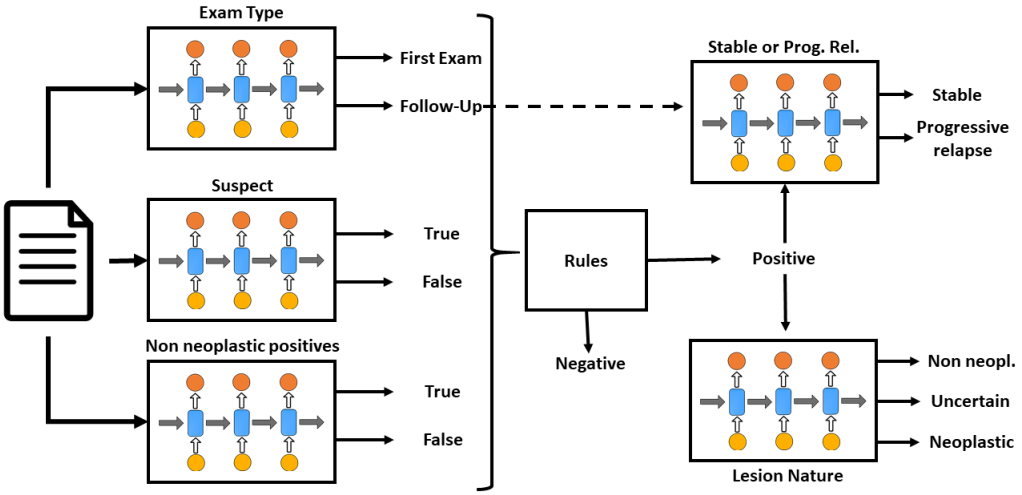


Figura 4.6: Modello 3. Il Risultato è calcolato mediante la combinazione, attraverso le regole definite dai radiologi, dei risultati di tre blocchi: quello Tipo Esame, quello Sospetto e quello dei Positivi Non Neoplastici, e nel caso dei *Follow-Up* dal blocco *Stabile* o *Progressione Recidiva*.

#### 4.3.3.3 Modello 3

Mentre i precedenti due modelli sono solo vagamente basati sulla conoscenza fornita dai radiologi, il Modello 3 (Figura 4.6) segue le regole che abbiamo descritto nella Sezione 4.1.1 ed è il più complesso dei tre.

Innanzitutto, come accade negli altri due modelli, il **Blocco Tipo Esame** identifica se un referto è un *Primo Esame* o un *Follow-Up*. Successivamente, seguendo le regole definite nella Sezione 4.1.1, abbiamo introdotto il **Blocco Sospetto** con l'obiettivo di identificare se un referto è sospettato di contenere la descrizione di una lesione di tipo neoplastico. Questo ci permette di individuare i referti più problematici senza le complicazioni dovute ai valori diversi del risultato per Primo Esame e Follow-Up e alla valutazione della stabilità delle condizioni del paziente. Questo blocco è allenato con l'intero training set ri-etichettato come segue:

- se il Risultato del referto è *Negativo*, sia per i *Primi Esami* che per i *Follow-Up*, il referto è considerato **Non Sospetto**;

- se il Risultato è *Stabile* o *Progressione Recidiva*, il referto è considerato **Sospetto**;
- se il Risultato è *Positivo*, si controlla il livello Natura Lesione: se il valore è *Natura Dubbia* o *Neoplastico*, allora il referto è considerato **Sospetto**, se il valore è *Non Neoplastico* **Non Sospetto**.

Seguendo la Regola 3 della Sezione 4.1.1, se un *Follow-Up* è classificato Non Sospetto, possiamo automaticamente derivare che il valore del Risultato è *Negativo*.

Seguendo la Regola 1, se un *Primo Esame* è classificato come Sospetto, il Risultato è *Positivo*. Tuttavia, dato che alcuni *Primo Esame* sono considerati Positivi anche senza il sospetto di una lesione di tipo neoplastico (Regola 2), abbiamo progettato il **Blocco Positivi Non Neoplastici** per riconoscere questi casi particolari, anche questo allenato con tutto il training set opportunamente ri-etichettato. Quindi, se un *Primo Esame* è riconosciuto come Sospetto o come *Positivo Non Neoplastico*, il risultato può essere automaticamente derivato come *Positivo*; altrimenti, il risultato è *Negativo*.

Successivamente, per i *Follow-Up* non classificati come *Negativo*, è necessario capire se siano di valore *Stabile* o *Progressione Recidiva*. Per fare ciò, abbiamo introdotto un ulteriore blocco (**Blocco Stabile o Prog. Recidiva**) allenato con tutti i *Follow-Up* esclusi quelli negativi. Senza la necessità di dover valutare in generale le condizioni del paziente (tutti questi referti infatti prevedono una lesione sospetta di neoplasia), questo blocco può semplicemente focalizzare la propria attenzione sul testo che confronta il referto corrente con i precedenti, verificando miglioramenti e peggioramenti.

Come per gli altri modelli, se un referto non ha risultato *Negativo*, viene analizzato anche dal **Blocco Natura Lesione**.

Il Modello 3 rappresenta un modo completo di introdurre le regole definite dai radiologi nella struttura dei blocchi classificatori. Se questa conoscenza del dominio può portare miglioramenti in termini di risultati, è evidente che, nel caso di una modifica dello schema o di adattamento di questo modo di procedere in altri contesti ospedalieri, debba essere rifatto praticamente da zero, seguendo le nuove indicazioni fornite dagli esperti.

#### 4.3.3.4 Classificazione del Sito Lesione

Il Sito Lesione indica dove si trovano le lesioni descritte nel referto. Per riuscire in questa identificazione, il modello di deep learning deve non solo individuare le parole che indicano parti del corpo, ma anche identificare le espressioni che indicano la presenza di una lesione.

Dato che un referto non negativo può avere lesioni in più di un sito, abbiamo la necessità di eseguire una classificazione multi-label. Per fare questo, potrebbe bastare un singolo blocco classificatore con attivazione sigmoide capace, eventualmente, di fornire in output anche una predizione di più classi contemporaneamente. Tuttavia, probabilmente a causa della doppia complessità del task e del dataset piuttosto esiguo (non avendo utilità per l'apprendimento i referti con risultato *Negativo*), questo ha ottenuto risultati peggiori rispetto all'addestramento di tre blocchi separati: uno per identificare le lesioni nel *Polmone*, uno per la *Pleura* e uno per il *Mediastino*.

Questi blocchi sono comuni a tutti e i tre modelli illustrati precedentemente, tuttavia non sono stati inclusi nelle Figure 4.4, 4.5 e 4.6 per mantenere la schematizzazione più comprensibile possibile.

## 4.4 Valutazione delle prestazioni

In questa sezione, riportiamo la valutazione delle prestazioni ottenute dai modelli di deep learning descritti nella Sezione 4.3.3, in particolare confrontandoli con il sistema precedente basato sulle annotazioni.

I modelli sono stati implementati usando la libreria Keras e Tensorflow come backend e sono stati valutati usando come metriche l'*accuracy*, ovvero il numero di predizioni corrette diviso per il numero complessivo di predizioni fatte e la *macro-average F-Measure (FM)*, ottenuta considerando la media aritmetica non pesata delle F-Measure [90] di ogni singola classe. La F-Measure, o F-Score, è definita come la media armonica tra la *precision* (P) e la *recall* (R), due misure che definiscono la capacità del modello di individuare la classe corretta evitando rispettivamente i falsi positivi e i falsi negativi. La F-Measure è calcolata nel seguente modo:

| | Annotazioni | | Modello 1 | | Modello 2 | | Modello 3 | |
|---|---|---|---|---|---|---|---|---|
| | Acc | FM | Acc | FM | Acc | FM | Acc | FM |
| `Tipo Esame` | 96.0 | 95.8 | 96.2 | 96.0 | 96.2 | 96.0 | **96.2** | **96.0** |
| `Risultato P.E.` | 77.3 | 76.1 | **81.7** | **80.8** | 80.4 | 79.6 | 78.3 | 76.3 |
| `Risultato F.U.` | 73.9 | 65.6 | 76.3 | 70.7 | 76.3 | 70.7 | **81.9** | **71.9** |
| `Natura Lesione` | 66.3 | 62.3 | 72.9 | 70.7 | 72.8 | 70.4 | **73.2** | **71.2** |
| `Polmone` | **93.2** | 71.9 | 90.3 | 75.6 | 90.8 | 75.8 | 90.9 | **76.6** |
| `Pleura` | 93.2 | 75.5 | 94.3 | 75.5 | **94.4** | **76.3** | **94.4** | 75.8 |
| `Mediastino` | **92.9** | **81.0** | 88.3 | 72.7 | 88.3 | 72.7 | 88.3 | 72.9 |

Tabella 4.2: Valutazione dei risultati dei modelli di deep learning, confrontati a quello basato sulle annotazioni, in 10-fold cross validation.

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, FM = 2 * \frac{P * R}{P + R} \tag{4.1}$$

in cui $TP$ (*true positives*) è il numero di istanze di quella classe correttamente predette dal modello, $FP$ (*false positives*) è il numero di predizioni sbagliate dal modello per quella classe e $FN$ (*false negatives*) è il numero di istanze di quella classe non correttamente predette dal modello.

Il sistema è stato testato in 10-fold cross validation, per cui nella tabella 4.2 viene riportata la media aritmetica delle metriche ottenute in ogni fold. La deviazione standard è compresa tra lo 0.7% e il 2.8%, il che significa che le prestazioni sono stabili tra un fold e l'altro. Per non complicare ulteriormente la lettura della tabella, non abbiamo riportato la relativa deviazione standard per ogni metrica.

Per ogni blocco classificatore che compone uno qualsiasi dei tre modelli, è stata svolta una semplice Random Search [7] per trovare i parametri del modello neurale (come ad esempio il numero di neuroni del livello LSTM) ricavando il 20% del training set e impiegandolo come validation set specifico per questo compito.

### 4.4.1 Risultati sperimentali

Nella Tabella 4.2 riportiamo la valutazione complessiva delle prestazioni dei modelli di classificazione gerarchica, confrontandoli con il modello basato

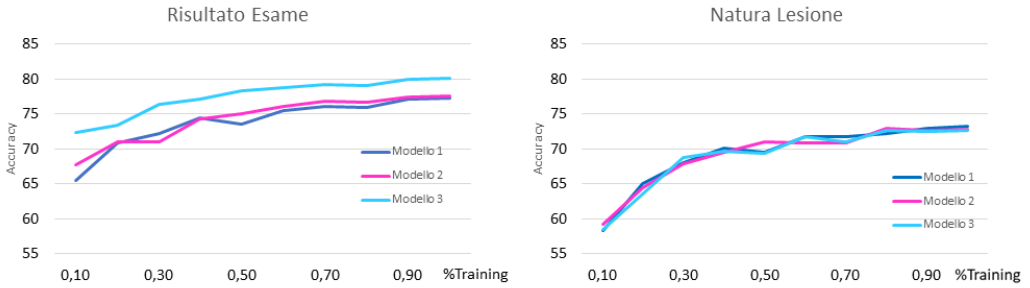


Figura 4.7: Miglioramento dell'accuracy per il livello Risultato Esame (a sinistra), considerando sia i Primi Esami che i *Follow-Up*, e il livello Natura Lesione (a destra) in 10-fold-cross validation per il Modello 1 (in blu), il Modello 2 (in rosa) e il Modello 3 (in azzurro). Sull'asse x, la percentuale di training set utilizzata per l'addestramento del modello.

sulle annotazioni in termini di accuracy e macro-average F-Measure, per tutti i livelli dello schema di classificazione. Mentre le prestazioni per il livello Tipo Esame sono molto simili, possiamo notare come il nuovo sistema ottenga prestazioni migliori per il livello Risultato, sia nel caso dei Primi Esami che per i *Follow-Up*, nonostante non vengano considerate le annotazioni manuali.

Nonostante l'architettura sia più semplice, il Modello 1 ottiene i risultati migliori per il Risultato Primo Esame. Dato che, nel nostro dataset, i Primi Esami sono circa il 35% del totale, riteniamo che questa differenza sia dovuta al fatto che nel Modello 1 il blocco per calcolare il Risultato è allenato sull'intero dataset. Nel Modello 2 invece, il blocco deputato per quel livello è allenato solo con i Primi Esami, risentendo quindi della minore quantità di dati.

Per quanto riguarda il Risultato *Follow-Up* è invece il Modello 3 che ottiene le migliori prestazioni, con una differenza significativa di oltre 5 punti di accuracy. Questo risultato è particolarmente importante in quanto i *Follow-Up* compongono circa il 65% dell'intero dataset. Per questo motivo, l'accuracy complessiva del livello Risultato Esame (considerando entrambe le configurazioni) per il Modello 3 è dell'80%, mentre per il Modello 1 77.5% e per il Modello 2 77.2%. Dato che il Modello 3 impiega le informazioni e le regole fornite dai radiologi, possiamo quindi osservare che l'utilizzo di cono-

| | Acc | FM |
|---|---|---|
| `Tipo Esame` | 96.0 | 96.2 |
| `Positivi Non Neoplastici` | 95.7 | 82.5 |
| `Sospetto` | 86.3 | 82.7 |
| `Stabile o Prog. Recidiva` | 81.8 | 81.6 |

Tabella 4.3: Prestazioni dei Blocchi Classificatori che compongono il Modello 3 in termini di accuracy (acc) e F-Measure (FM)

scenza del dominio, al fine di creare un'architettura gerarchica più complessa, migliora le prestazioni.

Inoltre, come è possibile vedere nella Figura 4.7, per il livello Risultato Esame, rispetto agli altri due allenati utilizzando l'intero training set, il Modello 3 ottiene prestazioni migliori pure addestrato considerando solo il 50% del training set. Dato che il Natural Language Processing in ambito biomedico è spesso condotto su dataset di dimensioni molto ridotte (confronta la Sezione 2.2), consideriamo questo risultato particolarmente significativo.

Data la complessità del Modello 3, riportiamo le prestazioni dei suoi componenti nella Tabella 4.3. In particolare, riconosciamo se un referto descrive un sospetto di lesione neoplastica con un'accuracy dell'86.3%. Il principale svantaggio di questo modello è il riconoscimento dei Positivi Non Neoplastici che, essendo tutti dei Primi Esami, sono la principale ragione per cui il Modello 3 ottiene prestazioni leggermente inferiori rispetto agli altri due nel calcolo del Risultato *Primo Esame*. Questo problema verrà affrontato nel dettaglio nella Sezione 4.4.2.

Per quanto riguarda il livello Natura Lesione, che ricordiamo utilizza la stessa procedura per tutti e tre i modelli, il nuovo sistema basato su deep learning ottiene risultati considerevolmente migliori (oltre 9 punti di F-Measure) rispetto al sistema basato sulle annotazioni. In maniera più dettagliata, questo confronto è riportato nella Tabella 4.4 riportando le F-Measure di ogni classe. Possiamo notare che, a fronte di un piccolo calo per la classe *Non Neoplastico*, le prestazioni relative alla *Natura Dubbia* e a *Neoplastico* crescono di oltre 14 punti. Tuttavia, la classe *Natura Dubbia* ottiene le peggiori prestazioni delle tre, con una F-Measure solo del 57.2%. Anche questo problema

| | Annotazioni | Deep Learning |
|---|---|---|
| `Non Neoplastico` | **76.6** | 73.5 |
| `Natura Dubbia` | 43.1 | **57.2** |
| `Neoplastico` | 66.9 | **81.0** |
| `Media` | 62.2 | **71.2** |

Tabella 4.4: Confronto dettagliato per il livello Natura Lesione tra il sistema basato sulle annotazioni e il sistema basato su deep learning, in termini di F-Measure

| | No Attention | | Attention | |
|---|---|---|---|---|
| | Acc | FM | Acc | FM |
| `Tipo Esame` | 94.6 | 94.0 | **96.2** | **96.0** |
| `Risultato P.E.` | 75.8 | 63.0 | **78.3** | **76.3** |
| `Risultato F.U.` | 76.5 | 59.3 | **81.9** | **71.9** |
| `Natura Lesione` | 64.8 | 61.3 | **73.2** | **71.2** |

Tabella 4.5: Confronto delle performance (in 10-fold cross validation) del Modello 3 senza e con l'attention mechanism nei blocchi classificatori.

verrà discusso nella Sezione 4.4.2.

Le prestazioni per il livello Sito Lesione sono riportate nella Tabella 4.2. Come mostrato nella Sezione 4.3.3.4, questo livello è calcolato allo stesso modo per tutti e tre i modelli, fornendo quindi risultati molto simili. Il nuovo sistema supera in prestazioni quello basato sulle annotazioni per la classe *Polmone* in termini di F-Measure, ottiene un leggero miglioramento per la classe *Pleura* ma peggiora per la classe *Mediastino*, di cui fa parte solamente il 5% dei referti per cui il riconoscimento di questi casi con metodologie di deep learning è molto complicato. Ricordiamo inoltre che mentre il sistema basato sulle annotazioni aveva specifiche feature per riconoscere le parole indicanti un potenziale sito per una lesione, anche in questo caso il nostro modello si basa sull'intero testo con l'obbligo non solo di riconoscere quali espressioni possano indicare un sito ma anche quelle che indicano la presenza di una lesione.

A sottolineare l'importanza dell'attention mechanism nella nostra architettura, nella Tabella 4.5 mostriamo un confronto tra le prestazioni (considerando i primi tre livelli di classificazione) del Modello 3 e un modello con la

stessa struttura ma in cui nei Blocchi Classificatori che lo compongono non è compreso l'attention mechanism. Se le performance per il Tipo Esame subiscono solo un leggero calo, per il Risultato Esame (in particolare in termini di F-Score e per i Follow-Up) i risultati sono nettamente inferiori. Lo stesso avviene per la Natura Lesione, il cui F-Score cala di quasi dieci punti.

### 4.4.2 Discussione e analisi degli errori

Come detto precedentemente, la classe *Natura Dubbia* per il livello Natura Lesione è quella più problematica. Questo è dovuto sia al fatto delle poche istanze di quella classe ma anche all'intrinseca difficoltà di stabilirne i confini, come dimostrato dall'*inter-annotator agreement* descritto in [24] e [72]. In questa analisi, il test set iniziale del sistema basato sulle annotazioni, composto da 68 referti, è stato ri-classificato da un altro radiologo. Per il livello Tipo Esame, i radiologi sono concordi nel 100% dei casi, per Risultato Esame nel 93% e per Natura Lesione solo nel 73%.

Andando più nel dettaglio, si può vedere come la maggior fonte di disaccordo per quest'ultimo livello sia proprio nella classe *Natura Dubbia*. Infatti, il 78% dei referti in cui i due radiologi non concordano sono classificati, dall'uno o dall'altro, come *Natura Dubbia*. La maggior difficoltà è stata riscontrata nella differenza tra *Natura Dubbia* e *Neoplastico*, che riguarda il 61% dei referti incerti per quel livello. Questa difficoltà è stata riscontrata da un ulteriore test che abbiamo svolto: allenando un blocco classificatore per la predizione del livello Natura Lesione con le classi *Neoplastico* e *Natura Dubbia* unificate, abbiamo ottenuto un'accuracy del 84.2% rispetto a 73.2% e una F-Score del 80.5% invece di 71.2%.

Ipotizziamo quindi che questi referti contengano, dal punto di vista medico e di rapporto col paziente, le informazioni maggiormente sensibili e che il loro linguaggio sia quello più vago e criptico. Allo stesso tempo, è probabile che alcuni casi possano essere identificati come *Neoplastico* o *Natura Dubbia* a seconda dell'opinione del medico.

Abbiamo inoltre analizzato perché il Modello 3 ottenga prestazioni peggiori rispetto agli altri due, per il Risultato *Primo Esame*. Riteniamo che

questo sia dovuto alla difficoltà di identificazione dei Positivi Non Neoplastici, ovvero quei Primi Esami che sono considerati positivi senza però la probabile presenza lesione neoplastica. Questi referti compongono solo il 6% dell'intero dataset e quindi la rete neurale ha difficoltà nella predizione, ottenendo un F-Score solo del 67%.

La dimensione del dataset è un ulteriore problema. Infatti la mancanza di referti influisce negativamente sui risultati specialmente per i livelli come Natura Lesione e Sito, che sono presenti solo per i 2.248 referti non negativi. Nella Figura 4.7 mostriamo come, allenando diversi modelli con sempre più referti (dal 10% al 100% del training set) e testandoli sullo stesso test set, l'accuracy migliori progressivamente. Analogamente, anche la scarsa disponibilità di referti non classificati contribuisce a peggiorare le prestazioni: infatti, mentre i comuni modelli di Word2Vec sono allenati utilizzando milioni di documenti [55], la nostra rappresentazione delle parole è stata ottenuta solo con circa 15mila referti. Inoltre, questa rappresentazione (come detto nella Sezione 4.3.2) non viene modificata durante il training dei diversi blocchi classificatori. Un interessante sviluppo futuro potrebbe valutare come diverse rappresentazioni (con più o meno referti, tenute fisse o modificate durante il training del modello) influenzano i risultati dei singoli blocchi e degli interi modelli.

### 4.4.3 Confronto tra i due sistemi

Indubbiamente, i risultati contenuti nella Tabella 4.2 provano che il sistema di deep learning offre complessivamente prestazioni molto migliori, specialmente per quanto riguarda gli aspetti più complicati e delicati della classificazione: decidere il risultato dell'esame e il tipo di lesione riscontrata.

Da un punto di vista applicativo, sia il sistema basato sulle annotazioni che quello basato su deep learning possono essere implementati su un server e connessi al reparto di radiologia. Mentre il precedente sistema si basava sull'integrazione di più tecnologie e linguaggi, la soluzione corrente può essere vista come un unico modulo software. Inoltre, mentre TextPro è un tool di qualche anno fa e non perfettamente documentato, spaCy fa parte dello

stato dell'arte per le fasi di pre-processing di analisi testuale ed è mantenuto e aggiornato regolarmente.

Tuttavia, mentre il sistema basato sulle annotazioni ha richiesto la raccolta e la classificazione di poche centinaia di referti, quello di deep learning ha richiesto un dataset 10 volte più grande, ottenuto in più di un anno di attività del reparto di radiologia. Questa limitazione, se si intendono pianificare progetti simili per l'analisi di patologie meno frequenti, può essere un serio svantaggio che è purtroppo comune a moltissime applicazioni di NLP in ambito clinico.

La Figura 4.7 mostra quanto aumentare il training set sia fondamentale per ottenere buoni risultati ma, allo stesso tempo, quanto l'introduzione di conoscenza del dominio e la creazione di modelli che ne tengano conto nella formulazione della predizione sia utile in mancanza di grandi quantità di dati. Il Modello 3 infatti riesce ad avvicinarsi all'80% di accuracy per il Risultato Esame anche solo considerando metà del dataset e quindi impiegando la metà del tempo e delle risorse per la raccolta e la classificazione dei dati.

Va inoltre ricordato che, se da un lato i referti del primo sistema erano solo poche centinaia, dall'altro necessitavano di essere annotati. Questo processo è chiaramente molto più lungo e costoso della classificazione, in quanto richiede una lettura più approfondita e l'individuazione di una spiegazione nel dettaglio della classificazione che si intende dare. La sola classificazione invece può essere data in pochi secondi, subito dopo la stesura del referto, semplicemente selezionando i valori dello schema da un'interfaccia grafica.

Nella prospettiva di un utilizzo di questo sistema nel contesto ospedaliero o di un eventuale adattamento per l'analisi di un altro tipo di documenti clinici (per esempio, referti di un'altra parte del corpo, cartelle cliniche, ecc.), è possibile addestrare un modello anche con sole poche centinaia di referti e renderlo subito disponibile ai radiologi. Dando inoltre la possibilità di validare le predizioni fornite dal sistema sui nuovi referti, sarebbe possibile raccogliere progressivamente un numero sempre maggiore di documenti con le rispettive etichette di classificazione. Con l'incremento delle dimensioni del training set, il sistema può essere periodicamente ri-allenato in modo tale da (come dimostra la Figura 4.7) migliorarne le prestazioni.

Come mostrato nella Sezione 4.4.2, per alcune classi come *Natura Dubbia*, i margini di errore o di discrezionalità dei radiologi nell'apporre l'etichetta di classificazione influiscono negativamente sulle prestazioni. L'analisi delle presenti etichette e la creazione di un *gold standard* che fornisca delle etichette quasi insindacabili, fornite dalla maggioranza dei medici o da quelli maggiormente esperti, chiaramente migliorerebbe le prestazioni di entrambi i sistemi ma con notevole dispendio di tempo ed energie.

Un'altra questione molto importante nel valutare le differenze tra i due sistemi è quella dell'interpretabilità. Le annotazioni manuali, o la loro replicazione in forma automatica, permettono di fornire un testo che giustifica la predizione data dal sistema. Infatti, come descritto nella Sezione 4.2, il sistema basato sulle annotazioni procede proprio da quelle annotazioni, e non dal resto, per definire la classificazione del referto. Tuttavia, né il processo di annotazione manuale né CRF hanno saputo coniugare l'interpretabilità con le prestazioni. Se abbiamo mostrato in questa sezione come il sistema di deep learning abbia ottenuto un notevole miglioramento delle prestazioni, la prossima sarà dedicata a quanto questo sistema può essere interpretabile.

## 4.5 Interpretabilità dell'Attention Mechanism

Di per sé, un sistema complesso come una rete LSTM è sostanzialmente una *black box* che produce risultati secondo logiche interne che poco hanno a che vedere con il ragionamento umano. Mentre il radiologo isola i rilievi descritti nel testo e li collega per formulare una diagnosi, la rete neurale opera per via matematica, rappresentando il significato del referto solo tramite vettori di numeri reali che, se da un lato possono essere elaborati dalla rete neurale, dall'altro sono totalmente incomprensibili all'essere umano.

D'altro canto però, l'Attention Mechanism descritto nella Sezione 3.4 opera su un livello molto più comprensibile: assegna infatti dei pesi agli elementi della sequenza, cioè alle parole. Una ragionevole aspettativa del funzionamento dell'attention mechanism sarebbe che dia peso maggiore alle parti che descrivono i rilievi e meno al contorno del referto (ad esempio come è stato

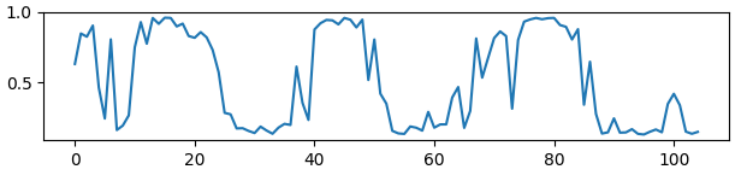


Figura 4.8: Visualizzazione dell'attention per un referto. Sull'asse x, gli indici $i$ delle parole nella sequenza originale del testo, sull'asse y il valore del peso dell'attention $w_i$

eseguito, frasi di routine ecc.), allo stesso modo in cui il ragionamento umano focalizza la propria concentrazione solo su alcuni parti del testo.

Nella Sezione 3.4.3 abbiamo parlato del ruolo dell'attention mechanism nell'interpretabilità, presentando i diversi punti di vista che si sono delineati negli ultimi anni di ricerca e letteratura sull'argomento. In questa sezione invece descriviamo il comportamento di questo meccanismo nella nostra applicazione, la sua efficacia e qual è il suo rapporto con le annotazioni manuali. Le analisi e i risultati più importanti di questa sezione sono stati recentemente pubblicati nei proceedings della conferenza Artificial Intelligence in Medicine 2021 [73].

### 4.5.1 La funzione di gate

Per l'esperimento che segue, abbiamo allenato un blocco classificatore di tipo *Sospetto* (sempre utilizzando il dataset di produzione) ed estratto i pesi $\alpha_i$ con $i \in [1, N]$ e $\sum_{i=1}^{N} \alpha_i = 1$ dell'attention mechanism, in cui $N$ è il numero di parole del referto. La scelta del blocco Sospetto è dovuta al fatto che il distinguere quali referti descrivono una lesione neoplastica e quali no è uno degli aspetti maggiormente importanti della nostra applicazione.

La visualizzazione dell'attention è spesso fatta riproponendo il testo evidenziato con un colore più o meno forte a seconda del peso assegnato alla parola (nel caso il peso sia minimo, la parola viene lasciata su sfondo bianco), come si farebbe sottolineando un libro. Riteniamo che, nell'ambito della nostra applicazione e della nostra analisi, questa visualizzazione non sia pienamente adeguata. Le sfumature di colore infatti possono essere fuorvianti,

mostrando come diversi due pesi simili oppure non mostrando pienamente le differenze tra un peso e l'altro. Abbiamo ritenuto invece più esplicativa una rappresentazione cartesiana in cui all'asse x corrispondono le parole, nell'ordine in cui compaiono nel documento, e sull'asse y vengono rappresentati i pesi dell'attention.

Con questa visualizzazione è possibile vedere nella Figura 4.8 come, mentre in linea teorica l'attention assegna semplicemente un vettore nell'intervallo continuo tra 0 e 1, all'atto pratico l'attention segnali settori *importanti*, a cui assegna un peso molto alto, settori *non importanti* con peso molto basso, e solo una minoranza del contenuto del referto stia nel mezzo. Possiamo facilmente notare, in questo esempio di referto con poco più di 100 parole, i settori importanti come quello successivo alla quarantesima parola e quelli non importanti come quello attorno alla trentesima parola e il numero ridotto di parole con peso medio-basso, che di fatto sono rappresentate solo dal picco attorno a 40 e da quello sul finale del referto.

Visto che l'effettivo numero $\alpha_i$ dipende dalla lunghezza del referto, sia nella Figura 4.8 che successivamente, il peso, che chiameremo $w_i$ per differenziarlo dall'originale, viene scalato utilizzando la seguente formula standard:

$$w_i = \frac{\alpha_i - \alpha_{min}}{\alpha_{max} - \alpha_{min}} \tag{4.2}$$

in cui $\alpha_{min}$ e $\alpha_{max}$ sono rispettivamente il valore minimo e massimo dei pesi $\alpha_i$ all'interno del referto.

Questa visualizzazione conferma quella che [91] ha chiamato, per i task a singola sequenza di parole come effettivamente è il nostro caso, la funzionalità di *gate* dell'attention mechanism, cioè la divisione tra una parte di testo che viene considerata e una che viene scartata.

Quello che è inoltre intuibile dalla Figura 4.8 è come l'attention, quando ha a che fare con un documento composto da almeno un centinaio di parole, non seleziona singole parole ma piuttosto intere espressioni, frasi o anche porzioni di testo più lunghe. In particolare, prendiamo in considerazione le frasi e calcoliamo il **peso complessivo della frase** come media dei pesi $w_i$ delle parole che compongono quella frase: da questa analisi possiamo

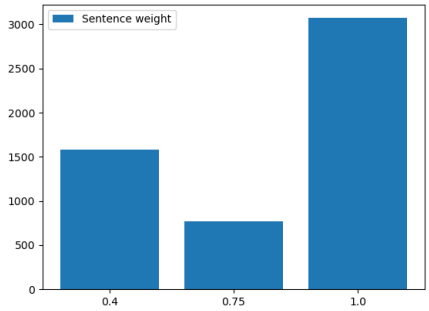


Figura 4.9: Distribuzione dei pesi delle frasi da scartare (peso inferiore a 0.4, prima colonna), delle frasi intermedie (peso compreso tra 0.4 e 0.75, seconda colonna) e delle frasi importanti (peso superiore a 0.75, terza colonna).

identificare tre categorie di frasi: le frasi importanti, con peso superiore a 0.75, quelle intermedie, con peso compreso tra 0.4 e 0.75, e quelle da scartare che hanno un peso inferiore a 0.4. Le soglie 0.4 e 0.75 sono state fissate dopo aver analizzato manualmente diversi grafici e diverse distribuzioni dei pesi dell'attention in modo tale che:

- frasi con la prevalenza di token a peso basso, in cui ci sono solo pochi token con peso relativamente alto, non siano considerate importanti o intermedie;
- frasi importanti relativamente corte con pochi token a peso basso (come congiunzioni o altre stop words) non siano considerate intermedie o poco importanti.

Un possibile sviluppo futuro del lavoro è analizzare ulteriormente la distribuzione dei pesi dell'attention considerando ulteriori soglie.

Analizzando la distribuzione delle frasi, per l'intero test set, possiamo dire che l'attention segnala il 56.5% delle frasi come importanti, il 29.1% come da scartare e il rimanente 14.4% come intermedie. Una visualizzazione di questo dato si può trovare in Figura 4.9. A questo però va aggiunto che molte delle frasi intermedie o di quelle da scartare con peso leggermente più alto delle altre lo sono per la presenza di un frammento di 4-5 parole a peso molto alto

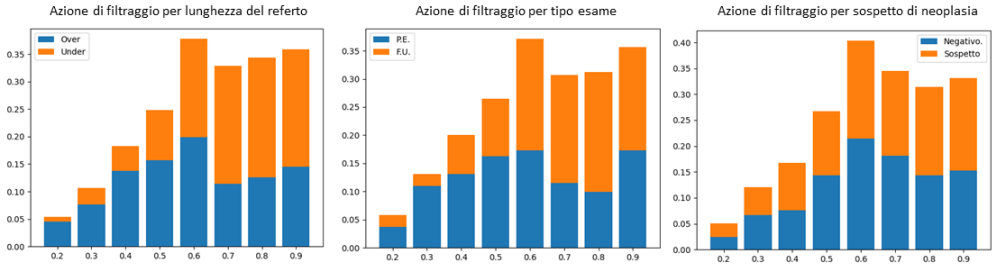


Figura 4.10: Visualizzazione dell'azione di filtraggio dell'attention dividendo i referti in lunghi (sopra le 120 parole) o corti (sotto le 120 parole), primi esami e follow-up, sospetti o megativi. Sull'asse x, la frazione di frasi importanti, rispetto all'intero documento, segnalate dall'attention. Sull'asse y la frazione di referti, per le due categorie.

rispetto al resto, di peso più basso: è estremamente difficile infatti trovare una lunga sequenza di parole con $w_i$ attorno a 0.5.

Questo ci porta a definire alcuni punti fondamentali:

- l'attention mechanism opera non a livello di singolo token ma all'interno di intere espressioni (come possono essere i rilievi e le loro caratterstiche) o, ancora più spesso di frasi.

- l'attention mechanism tende a dividere la sequenza di input in modo binario: il testo da considerare e quello da scartare, senza quantificare numericamente il peso della sequenza per la classificazione;

- l'attention mechanism non opera un filtraggio molto selettivo, in quanto, mediamente, più della metà delle frasi è ritenuta importante. Questo fa sì che la sua utilità in termini di spiegazione della predizione fatta sia limitata: mostrare infatti più della metà del testo di un referto come motivazione, invece che alcune espressioni chiave, può essere ritenuto piuttosto ridondante e non sufficientemente informativo.

### 4.5.2 Differenze di comportamento in base alle caratteristiche del referto

Tuttavia, il comportamento dell'attention mechanism non è uguale per tutti i referti. Nella Figura 4.10, possiamo vedere come l'azione di filtraggio dell'attention, cioè il riconoscimento che parte del testo non è utile ai fini della classificazione, varia in base alla lunghezza del referto, al tipo esame o al sospetto di lesione neoplastica.

Una distinzione molto importante è tra referti lunghi e corti. Identificando il confine tra le due categorie a 120 parole (la mediana della distribuzione delle lunghezze dei referti del test set), possiamo vedere nel primo istogramma come ci siano pochissimi referti corti di cui è segnalato come importante solo il 20% delle frasi. Questa percentuale anzi aumenta, e infatti la maggior parte dei referti corti mantiene oltre il 60% del suo testo secondo i criteri dell'attention mechanism. Al contrario, più aumenta l'azione di filtraggio, considerando importanti solo il 50% o meno delle frasi, più possiamo notare una prevalenza dei referti lunghi, contrassegnati dal colore blu nel primo grafico della Figura 4.10.

Un comportamento molto simile si ha per la distinzione tra Primi Esami e *Follow-Up*, nel secondo istogramma. I Primi Esami, di solito più descrittivi e spesso più lunghi, vengono filtrati mediamente di più rispetto ai *Follow-Up* che, in diversi casi, si limitano a poche righe che di solito l'attention seleziona in toto.

L'azione dell'attention nel selezionare le frasi importanti è invece del tutto indipendente dalla distinzione tra referti Sospetti e Non Sospetti. Come è possibile vedere nel terzo istogramma, non c'è significativa differenza nella distribuzione dei referti in base alla percentuale di frasi ritenute importanti.

A nostro parere, questa analisi conferma, come detto in [37], che l'attention debba essere valutato sia in relazione al compito da svolgere che all'input. Nel nostro caso, la necessità di semplificare può portare l'attention a filtrare significative porzioni del referto nel caso di referti molto lunghi e con descrizioni non particolarmente significative, ma nel caso di referti di poche righe immediatamente interpretabili può decidere di mantenere tutto il testo.

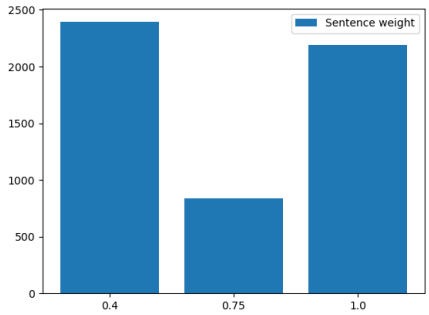


Figura 4.11: Distribuzione dei pesi delle frasi per la classificazione tra *Primo Esame* o *Follow-Up*.

### 4.5.3 Differenze di comportamento in base al livello

Proprio per mostrare quanto il compito richiesto al modello di classificazione sia fondamentale nel comportamento dell'attention, proviamo a ripetere l'esperimento di addestrare un modello e di estrarre i pesi dell'attention. Stavolta tuttavia, invece che per classificare un referto Sospetto o Non Sospetto, il compito sarà di capire il Tipo Esame, ovvero se un referto è *Primo Esame* o *Follow-Up*.

Se confrontiamo la Figura 4.11 con quella analoga per il blocco Sospetto (Figura 4.10), possiamo notare subito come l'attention mechanism, allenato con lo stesso dataset e verificato con lo stesso test set ma per un altro compito di classificazione, si comporti in modo molto diverso. Infatti, il numero di frasi ritenute da scartare è aumentato di più di 1.000 unità, mentre quelle importanti sono diminuite di oltre 500.

Questo comportamento può essere facilmente spiegato dal fatto che per riconoscere che un referto è un *Follow-Up* è sufficiente identificare correttamente nell'intestazione espressioni come *confrontato con precedente in data...* oppure *si confronta con TC del...* e, dall'inserimento prima della descrizione del rilievo di parole come *invariato* oppure *incremento*, che appunto suggeriscono che quanto si vede nella TC corrente si rapporta a quanto analizzato precedentemente. Per questo motivo, l'analisi del dettaglio delle descrizioni dei rilievi è totalmente inutile per questo compito.

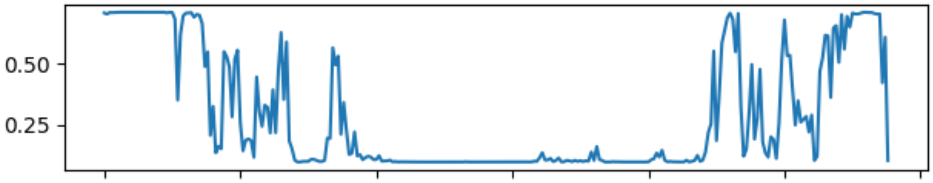


Figura 4.12: Visualizzazione dell'attention per un referto per la predizione del Tipo Esame. Sull'asse x, gli indici $i$ delle parole nella sequenza originale del testo, sull'asse y il valore del peso $w_i$

Nella Figura 4.12, possiamo vedere come l'attention selezioni solamente la prima parte (che nel documento corrisponde a *TORACE-ADDOME COMPLETO indagine condotta prima e dopo somministrazione di mdc non ionico ev(Xenetix 350 , 120 ml). Esame confrontato con precedente del 25 novembre 2015*) per poi saltare gran parte del corpo del referto e poi segnalare solo alcune tra le ultime frasi, tra cui *Invariata la piccola lesione* e *Invariato il resto*. In altri casi, tuttavia, come nel caso del blocco Sospetto, il referto è sufficientemente corto da non richiedere un particolare filtraggio da parte dell'attention, il che spiega comunque il numero piuttosto alto di frasi ritenute importanti visibili in Figura 4.11.

### 4.5.4 Confronto con le annotazioni manuali

Finora ci siamo occupati di come l'attention pesa le sezioni del referto a seconda della loro importanza, mostrando affinità e divergenze rispetto all'input e al compito richiesto dal modello di classificazione. Mentre questa analisi va nella direzione di un'interpretazione *fedele*, cioè di descrizione dei processi interni del modello senza alcuna relazione con il risultato finale della predizione, ora ci occuperemo di quanto i settori selezionati dall'attention possano essere realmente quelli utili ai fini della classificazione, ovvero dell'interpretazione *plausibile* come descritto in [37] e nella Sezione 2.3.1.

Chiaramente, per valutare se le frasi ritenute importanti dall'attention lo siano anche effettivamente dal punto di vista diagnostico serve conoscenza medica. Di conseguenza, abbiamo confrontato i nostri risultati con l'unica fonte di conoscenza medica applicata all'analisi del testo che abbiamo a di-

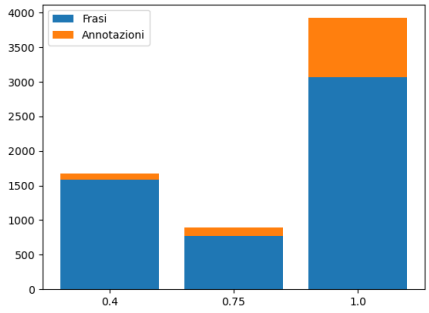


Figura 4.13: Confronto tra annotazioni manuali e pesi dell'attention. In blu, la suddivisione delle frasi, in arancione il numero di annotazioni contenute nelle frasi importanti, intermedie o non importanti.

sposizione: le annotazioni manuali, utilizzando quindi il dataset annotato di 346 referti (che, come detto nella Sezione 4.5.1, non sono stati inclusi nei dati di training).

Va subito chiarito che si tratta di un confronto che presenta dei limiti. Mentre un'annotazione manuale è composta mediamente da 4 token, il filtraggio dell'attention riguarda settori di referto decisamente più lunghi. Non è praticamente possibile quindi che, nel nostro caso di studio, il testo del referto ritenuto importante dall'attention mechanism coincida esattamente con le annotazioni manuali Tuttavia, questo può portare anche dei vantaggi: nella Sezione 4.2.1 abbiamo infatti sottolineato come molto spesso i radiologi annotassero quasi solamente espressioni molto brevi, senza includere dettagli fondamentali come le caratteristiche di un nodulo di una lesione, oppure annotassero solamente un rilievo tra i diversi menzionati nel referto. L'attention, concentrandosi su settori più lunghi, può evidenziare anche questi dettagli e fornirli come parte di una spiegazione.

Piuttosto, è ragionevole supporre che, se il testo rilevato dall'attention è quello su cui correttamente un radiologo baserebbe il proprio ragionamento per dare la classificazione, le annotazioni manuali siano contenute in esso. Abbiamo quindi eseguito questo esperimento:

- per ogni referto, abbiamo estratto le espressioni annotate per il Risultato Esame e per la Natura Lesione, ovvero quelle parti di testo che, se-

condo i radiologi, mostrano la presenza di una lesione e se quest'ultima è di tipo neoplastico, dubbio o non neoplastico;

- abbiamo allenato un blocco Sospetto ed estratto i pesi dell'attention mechanism;
- in base ai pesi dell'attention, abbiamo calcolato il peso delle frasi e divise in *non importanti*, *intermedie* e *importanti*, analogamente a quanto descritto nella Sezione 4.5.1;
- abbiamo verificato, per ogni annotazione manuale, se fosse contenuta in una frase importante, intermedia o non importante.

In Figura 4.13 il risultato di questo esperimento. Come è possibile notare sommariamente anche nell'istogramma, l'80.2% delle annotazioni manuali è contenuto nelle frasi importanti, che ricordiamo sono però solo il 56% del totale. Invece, solo l'8.9% delle annotazioni è contenuta in una frase ritenuta non importante, mentre il restante 10.8% è contenuto in una frase intermedia. Si può quindi vedere come ci sia una correlazione forte tra quanto selezionato dall'attention, ovviamente in modo molto meno selettivo, e quanto invece dai radiologi.

Questo può non bastare a considerare l'attention mechanism uno strumento di interpretazione plausibile nella classificazione di un referto radiologico che descrive una lesione di sospetta natura neoplastica. Tuttavia può fornire un indizio del fatto che il suo ragionamento, da correlare inoltre con il processo di analisi del testo fatto dal livello LSTM e dall'ultimo livello fully-connected con attivazione softmax che compongono il blocco classificatore, abbia un fondamento anche rispetto alla conoscenza medica.

L'impatto della scelta della threshold nella percentuale di frasi ritenute importanti e delle annotazioni riconosciute è mostrato nella Tabella 4.6. Come prevedibile, all'aumentare della soglia diminuisce la percentuale di annotazioni riconosciute (da 88.9 considerando 0.50 a solo il 58.2% considerando 0.95) e delle frasi ritenute importanti. Per tutte le soglie, è verificabile il fatto che la percentuale di annotazioni riconosciute è ben più alta di quella delle

| Soglia | % Annot. | % Frasi |
|---|---|---|
| 0.50 | 88.9 | 67.5 |
| 0.55 | 87.5 | 65.3 |
| 0.60 | 85.6 | 62.9 |
| 0.65 | 83.7 | 61.2 |
| 0.70 | 81.4 | 59.9 |
| 0.75 | 80.2 | 56.6 |
| 0.80 | 75.5 | 53.2 |
| 0.85 | 72.4 | 49.9 |
| 0.90 | 67.3 | 46.0 |
| 0.95 | 58.2 | 39.3 |

Tabella 4.6: Valutazione dell'impatto della scelta della soglia (prima colonna) per considerare una frase importante. La seconda colonna rappresenta la percentuale di annotazioni contenuta nelle frasi importanti, la terza la percentuale di frasi ritenute importanti dall'attention.

frasi ritenute importanti, quasi sempre di oltre 20 punti percentuali. Vale comunque la pena sottolineare che anche considerando soglie molto alte viene riconosciuta quasi la metà delle frasi come importanti (ad esempio, il 46% prendendo 0.90). Questo a conferma, come detto nella Sezione 4.5.1, della tendenza dell'attention mechanism ad assegnare pesi normalizzati molto vicini a 1.

#### 4.5.4.1 Annotazioni non rilevate

Delle 1128 annotazioni considerate, 107 sono contenute in frasi non considerate importanti dall'attention mechanism. Leggendole e analizzandole, è possibile individuare alcuni casi ricorrenti e provare a spiegare i motivi per cui non sono state prese in considerazione:

- *Annotazioni contenute in frasi negative*: nel 16% dei casi sono stati annotati concetti importanti come lesioni o noduli ma all'interno di frasi che ne escludono la presenza. Ad esempio, la frase *Non lesioni focali riferite a localizzazioni secondarie* contiene due annotazioni: *non lesioni focali* e *localizzazioni secondarie* che non sono state riconosciute. Un motivo probabile per cui questi casi non sono riconosciuti può essere

quello che i referti Non Sospetti sono la classe di maggioranza, con circa il 73% delle istanze del test set: il modello quindi, in assenza di testo rilevante che gli possa dimostrare il contrario, tende a predire Non Sospetto, non avendo quindi la necessità di dare peso maggiore alle parti del testo in questo senso.

- *Adenopatie e linfonodi*: il 35% dei casi presenta riferimenti ad adenopatie o più in generale di infiammazioni dei linfonodi. In mancanza di ulteriore specificazione, queste manifestazioni non hanno un diretto collegamento con la neoplasia e quindi sono presenti sia nei referti Sospetti che Non Sospetti. Va anche detto però che, in alcuni casi, l'insorgere di nuove adenopatie o il loro incremento volumetrico può essere dovuto alla presenza di una lesione neoplastica per cui anche questa parte di testo merita attenzione. È comunque probabile che, in questi casi, nel referto siano presenti riferimenti ben più evidenti (la descrizione di lesioni e noduli, ad esempio) che possono essere sufficienti per la classificazione.

- *Conclusioni*: in circa l'8% del training set è presente in coda al referto la sezione delle conclusioni che riassume il referto e, se presa in considerazione, potrebbe essere molto importante per la predizione. Ad esempio, se dopo la descrizione di un nodulo di cui vengono riportate solamente le dimensioni (insufficienti per capirne l'origine e quindi per la predizione), nelle conclusioni compare la frase *I rilievi riscontrati sono di natura flogistica*, possiamo automaticamente capire che, essendo l'origine dei rilievi infiammatoria, il referto non presenta alcun sospetto di neoplasia. Le poche annotazioni di questo tipo sono raramente evidenziate dall'attention, probabilmente per la scarsità di esempi di training a cui fare riferimento.

Altri casi più rari o meno caratteristici possono variare dal semplice errore del modello di machine learning e dalla mancata annotazione di un concetto fondamentale come una lesione, o dalla presenza di espressioni molto rare, seppur importanti, come *esiti di radioterapia* che è sicuramente probante nel

capire che il referto tratta una neoplasia (la radioterapia è infatti una terapia per la cura del tumore), ma che tuttavia è descritta solo nell'1% dei referti del training set.

### 4.5.5 Frammenti importanti

Per ora, abbiamo semplicemente considerato se un'annotazione è contenuta in un'intera frase che, complessivamente, è stata ritenuta importante dall'attention mechanism. Nel nostro contesto applicativo, come abbiamo visto precedentemente, l'attention tende a lavorare a livello di frase e non di espressioni di poche parole o addirittura singoli token.

Tuttavia, non è raro trovare qualche token con peso molto alto all'interno di frasi intermedie o non importanti. Considerando anche questi *frammenti* (che sono circa 1500 per un test set di poco più di 500 referti, per cui mediamente 3 a referto) come parte del testo considerato importante dall'attention, non si hanno grandi miglioramenti: infatti viene riconosciuto l'82.3% delle annotazioni, rispetto all'80.1% senza questa inclusione.

Anche qui, possiamo cercare di capire come si comporta l'attention a livello più basso e riconoscere alcuni casi ricorrenti:

- *La parola "non"*: nell'8% dei casi, percentuale che riteniamo comunque essere piuttosto significativa, viene annotata un'espressione composta da al massimo 3 parole che inizia con "non" come ad esempio semplicemente *non* o *non lesioni*. Riconoscere le negazioni è fondamentale in applicazioni di NLP, tanto che il sistema basato sulle annotazioni aveva un apposito strumento per farlo, per cui non sorprende che (specialmente nel nostro caso in cui diverse frasi iniziano con "Non" e queste caratterizzano molti referti negativi) parte della concentrazione dell'attention sia focalizzata lì.

- *"Invariato" o "immodificato"*: analogamente alla negazione, parole che indicano la sostanziale stabilità delle lesioni sono molto importanti, in quanto l'incremento dimensionale di un rilievo può essere dovuto a una lesione neoplastica. Per questo motivo, molte frasi presenti nei referti

sono atte a escludere questa ipotesi e molto spesso iniziano con *Invariato*, *Immodificate le dimensioni* o espressioni simili. Queste espressioni, che riteniamo importanti quanto il riconoscimento delle negazioni, rappresentano l'11% dei frammenti.

- *Espressioni per correlazione con un esame precedente*: un altro 9% di casi si riferisce alla sottolineatura, da parte dell'attention, delle espressioni che caratterizzano i *Follow-Up* quali *confrontato con precedente del 26/10/2015*, a volte anche solo semplicemente della data, all'interno dell'intestazione del referto. Sebbene queste espressioni siano più importanti per capire il Tipo Esame rispetto ai livelli successivi, è pur vero che una menzione di un nodulo in un *Primo Esame*, e quindi senza queste espressioni, si accompagna molto più spesso a un sospetto di lesione neoplastica. In un *Follow-Up* di solito la sola menzione, senza descrizioni particolari o segnalazioni di un aumento di dimensione, di solito non dà adito a sospetti. Per questo motivo, ci sembra corretto che l'attention mechanism attribuisca un peso alto a queste espressioni.

- *Espressioni complesse in frasi molto lunghe*: il 12% dei casi ha invece un comportamento più simile a quello descritto nella Sezione 4.5.1, cioè di annotazioni di settori composti da più di 10 token che tuttavia sono compresi in frasi molto lunghe e articolate. Ad esempio, nella frase composta da 47 token *Al controllo attuale compare infiltrazione dell'arco anteriore della I costa di destra con erosione della corticale profonda ed infiltrazione della midollare ossea da parte della lesione espansiva solida apicale destra che presenta ampia superficie di contatto con la pleura costale antero-laterale.* viene evidenziata dall'attention solo la parte che effettivamente descrive la lesione, cioè gli ultimi 18 token, punto compreso. È tuttavia difficile valutare automaticamente la correttezza di questa operazione: se in questo esempio la selezione è corretta, in altri casi alcune parti tralasciate (ad esempio un *immodificato nel tempo* dopo la descrizione di un nodulo) potrebbero essere utili ai fini della predizione o della spiegazione.

Accanto a questi casi ricorrenti, che compongono circa il 40% dei frammenti segnalati dall'attention, tra i singoli token sottolineati possiamo riscontrare la punteggiatura, che comunque ha un ruolo molto importante nel comprendere un testo, ma anche casi meno comprensibili come articoli e preposizioni oppure nomi e aggettivi senza una particolare rilevanza apparente, come dimostra lo scarso contributo che l'inclusione di questi frammenti ha prodotto rispetto alla percentuale di annotazioni riconosciute.

### 4.5.6 Valutazione generale e applicabilità

Mentre i sistemi di classificazione, quello basato sulle annotazioni e quello di deep learning, possono essere facilmente implementati e resi disponibili ai radiologi, integrare questi sistemi con meccanismi che forniscano una spiegazione alla predizione fornita dall'algoritmo è una questione decisamente più complessa.

Il sistema basato sulle annotazioni addestra i modelli utilizzando solamente alcune parti di testo, che possono eventualmente essere restituite o evidenziate al radiologo come giustificazione. Sebbene nel sistema siano presenti anche componenti meno intellegibili (come le regole per derivare la classificazione del referto da quella delle singole annotazioni), le annotazioni automatiche potrebbero già rappresentare un aiuto per spiegare il funzionamento del sistema. Oltre al fatto che le prestazioni di questo sistema non sono state ritenute soddisfacenti, anche la presenza di problemi nelle annotazioni manuali (e quindi anche in quelle automatiche, come visto nella Sezione 4.2.1), pongono seri limiti al suo utilizzo nella pratica medica.

Il sistema di deep learning invece, a partire dal fatto che l'input è trasformato in un vettore di numeri reali totalmente oscuro, è ancora meno comprensibile. D'altro canto però questo sistema ottiene prestazioni nettamente superiori e l'utilizzo dell'attention mechanism permette di fornire qualche indicazione sul comportamento del modello. In particolare, il funzionamento abbastanza semplice della funzione di gate, per cui per gran parte del testo siamo in grado di dire se è importante o meno ai fini della computazione della predizione, può escludere immediatamente una buona porzione di

testo non rilevante. Tuttavia a nostro parere non può essere immediatamente impiegato per l'interpretabilità, in quanto la sua tendenza ad evidenziare cospicue parti del referto (in media il 56%) non permette infatti di ritornare il testo evidenziato dall'attention come spiegazione, semplicemente perché troppo grande. Paradossalmente, per alcuni referti verrebbe prodotto come giustificazione l'intero testo e sapere che l'algoritmo si sia basato, per dare la predizione, su quanto scritto nel referto non è certo di grande utilità.

In generale, la funzione di gate dell'attention mechanism non permette di isolare piccole porzioni di testo che forniscano una risposta, concentrandosi su intere frasi. Nei casi particolari in cui vengono evidenziate poche parole (quasi sempre all'interno di frasi poco importanti) non vi è garanzia dell'esattezza del procedimento. Infatti, a volte le parole evidenziate sembrano effettivamente sensate (come il riconoscimento di negazioni o del non peggioramento di un rilievo), altre volte invece sembrano non essere particolarmente rilevanti e dovute alla percentuale di errore connaturata a qualsiasi sistema di machine learning. Tuttavia, analizzando il comportamento in relazione alle annotazioni manuali e ad alcuni concetti medici descritti nei referti, si può vedere come l'attention mechanism possa mettere in luce alcune caratteristiche del funzionamento del modello. Come mostrato nelle sezioni precedenti, la correlazione tra le annotazioni manuali e le frasi ritenute importanti dall'attention rinforzano l'impressione che le prestazioni del sistema di deep learning siano dovute a un processo che ha alcuni punti in comune col ragionamento medico.

Se le nostre valutazioni si sono svolte considerando il peso medio di 0.75 come valore oltre il quale si definisce una frase come importante, nella Tabella 4.6, abbiamo mostrato come l'aumento della soglia non modifichi sostanzialmente la situazione. Un valore più alto, come 0.90 ad esempio, considererebbe ugualmente come importante quasi la metà delle frasi, diminuendo la percentuale di annotazioni riconosciute da oltre l'80% a poco più del 67%. Si ritiene quindi necessario combinare l'attention mechanism con altre metodologie di interpretabilità, quali la sottrazione o la perturbazione delle feature, per arrivare a spiegazioni più precise che possano essere applicate anche nella pratica clinica.

## 4.6 Approcci preliminari con BERT

I modelli basati quasi esclusivamente sull'attention mechanism come Transformer e BERT (vedi Sezione 3.4.2) hanno ottenuto ottime prestazioni nei principali task di Natural Language Processing. Anche per la lingua italiana, sono stati addestrati alcuni modelli di BERT come quello presentato in [68], tuttavia solo su corpus generici o presi dai social network.

Nell'analisi di un linguaggio specifico come quello clinico, questo può rappresentare un problema. Tuttavia, nonostante l'indisponibilità di modelli specifici, abbiamo provato il nostro task di classificazione di referti radiologici con il modello standard di BERT per la lingua italiana[3].

Come spiegato nella Sezione 3.4.2, BERT viene allenato con due compiti paralleli. Dato un corpus di documenti, vengono mascherate un certo numero di parole ed un certo numero di frasi che il modello dovrà riuscire a predire dato il contesto (sia esso composto dalle parole rimanenti nella frase o dalle frasi non mascherate). Questi due compiti sono utili per capire il funzionamento del linguaggio e il significato delle parole, ma di per sé non bastano per risolvere uno specifico task di Machine Learning. Nel nostro caso infatti, la risposta che ci deve fornire il modello è, ad esempio, se il referto descrive una lesione di tipo neoplastico o meno; non siamo interessati al fatto che sia in grado di predire una parola mancante all'interno della frase. Per questo motivo, il modello pre-allenato deve essere adattato tramite un processo di *fine tuning* in cui:

- Il modello pre-allenato funziona da encoder, rappresentando la frase come un vettore di numeri reali. Dato che il modello è in grado di predire frasi e parole in base al contesto, questa rappresentazione riesce a catturare il significato della frase in termini numerici.

- Questa rappresentazione viene analizzata da un livello fully-connected, definito in base allo specifico task di machine learning che il modello è chiamato a risolvere. Per il livello di Natura Lesione, ad esempio, avremo quindi 3 neuroni con attivazione softmax.

[3]https://huggingface.co/dbmdz/bert-base-italian-cased

- Viene quindi allenato il modello composto dall'encoder di BERT, i cui pesi sono stati già appresi durante l'addestramento precedente, e dal livello fully-connected (inizializzato casualmente), utilizzando il training set del task di machine learning. La backpropagation andrà quindi a modificare sia i pesi dell'encoder che quelli dell'ultimo livello, andando quindi a produrre un nuovo modello di BERT, da cui eventualmente è possibile estrarre l'encoder. Essendo teoricamente il modello di BERT già allenato e capace di comprendere il linguaggio, questo fine tuning va a modificare solo leggermente il suo funzionamento, specializzandolo per il task.

### 4.6.1 Risultati sperimentali

Data la relativa novità dell'approccio, ci si è potuti concentrare su questi esperimenti solo negli ultimi mesi. Per questo motivo, non sono ancora stati replicati i modelli presentati nella Sezione 4.3.3 ma sono stati solamente testati tre blocchi classificatori. Più in dettaglio, gli sforzi si sono concentrati su:

- in primo luogo, sul **Blocco Tipo Esame**, che svolge un task molto semplice e che quindi può essere usato come esperimento di base. Nel caso in cui il modello basato su BERT abbia risultati notevolmente inferiori rispetto a quelli basati su LSTM, è infatti molto probabile un calo ancora maggiore per i livelli successivi.

- il **Blocco Sospetto** del Modello 3, in quanto parte centrale dell'applicazione che è rivolta a capire, in modo automatico, quali referti descrivono possibili neoplasie.

- il **Blocco Natura Lesione**, comune a tutti i modelli e il task più complicato dei tre, sia per la ridotta dimensione del dataset che per la presenza dei referti di tipo Natura Dubbia.

Anche in questo caso, le prestazioni vengono valutate in 10-fold cross validation. Se questo è decisamente più oneroso dal punto di vista computazionale (infatti bisogna svolgere il fine tuning di BERT per dieci volte), d'altro

| | BERT | | LSTM | |
|---|---|---|---|---|
| | Acc | FM | Acc | FM |
| `Tipo Esame` | 95.9 | 95.6 | **96.2** | **96.0** |
| `Sospetto` | 85.8 | 81.7 | **86.3** | **82.7** |
| `Natura Lesione` | 71.8 | 69.9 | **73.2** | **71.2** |

Tabella 4.7: Confronto delle prestazioni, per tre blocchi classificatori, del modello basato su BERT e quello basato su LSTM in termini di accuracy (Acc) e macro-averaged F-Measure (FM).

canto ci garantisce una maggiore stabilità e validità dei risultati rispetto a un unico esperimento (vedi anche la Sezione 5.6.2).

Nella Tabella 4.7 mostriamo il confronto tra i risultati ottenuti da un modello basato sulla versione standard di BERT per la lingua italiana e quelli addestrando ex novo un modello LSTM, come descritto nelle sezioni precedenti. Mentre i risultati per il Tipo Esame sono sostanzialmente invariati (il leggero calo può essere dovuto a una fluttuazione statistica), la differenza per il Blocco Sospetto e la Natura Lesione è più netta, con circa un punto di F-Score in meno per entrambi. Mentre l'accuracy del Blocco Sospetto cala solo di 0.5 punti, quella della Natura Lesione nel modello di BERT cala di circa 1.4.

Questo calo è spiegabile con la difficoltà di BERT nell'adattarsi al linguaggio biomedico, in quanto il modello è stato pre-allenato su un corpus del tutto generico. Tuttavia, le prestazioni con il solo fine tuning sono comunque di ottimo livello, e questo depone chiaramente a favore di BERT come strumento per l'analisi del linguaggio, anche in domini specifici.

Riteniamo molto probabile che arricchendo il dataset oppure proseguendo l'addestramento di BERT con dell'altro testo clinico, le prestazioni possano raggiungere e superare quelle dei modelli basati su LSTM. Per questo motivo, la ricerca proseguirà in questa direzione oltre che nell'implementazione dei modelli completi e non solo dei singoli blocchi classificatori, in modo tale da svolgere un confronto più completo. Allo stesso tempo, tuttavia, i problemi relativi all'incertezza e alla qualità delle etichette di classificazione descritte nella Sezione 4.4.2, pongono necessariamente un limite superiore al miglioramento anche per i sistemi basati su BERT.

Per quanto riguarda l'interpretabilità, il modello standard di BERT per la lingua italiana possiede 12 livelli di encoder, ognuno dei quali comprende altrettanti self-attention. La via quindi dell'analisi dell'attention come è stata impostata in questa tesi non è più percorribile e fornire una spiegazione del perché BERT ha fornito una determinata risposta è un problema ancora aperto. Recenti studi, che analizzano sia come BERT rappresenta al proprio interno le parole, sia il comportamento dei diversi attention mechanism per ogni livello di encoder, mostrano come questi modelli riescano a immagazzinare e rappresentare informazioni sia linguistiche che relative al contesto applicativo [16, 56, 89]. Per questo motivo, il nostro obiettivo sarà di estrapolare queste ultime e confrontarle, ancora una volta, con la spiegazione che fornirebbe il medico.

# Capitolo 5

# Estrazione di interazioni tra farmaci

Dal 1997, il database di letteratura scientifica MedLine e il suo motore di ricerca PubMed sono disponibili per la consultazione online, fornendo milioni di abstract e riferimenti riguardanti la ricerca in ambito medico. Una buona parte di questi studi riguarda la farmacovigilanza, ovvero l'analisi dei rischi correlati all'assunzione di farmaci (come effetti collaterali e potenziali reazioni avverse) in particolare quando più farmaci vengono assunti nello stesso periodo di tempo.

La realizzazione di database strutturati, come DrugBank o IUPHAR/BPS, per contenere le informazioni relative alle reazioni avverse dei farmaci (ADR, da *adverse drug reactions*) e soprattutto il loro continuo aggiornamento, alla luce di nuovi farmaci introdotti o di nuovi studi, richiede tuttavia un notevole sforzo nell'analisi di una quantità enorme di testo, preso dalla letteratura scientifica, riguardante la farmacovigilanza.

Per questo motivo, l'utilizzo di tecniche automatiche per l'estrazione di relazioni tra farmaci è stato l'obiettivo di diverse competizioni di machine learning e Natural Language Processing, con l'obiettivo di realizzare modelli sempre più precisi e accurati. Tra queste, una delle competizioni più famose è la DDI-2013 [35], realizzata nel 2013 ma considerata ancora importante e diventata uno degli standard del NLP in ambito biomedico.

In questo capitolo verranno descritti i modelli da noi realizzati, con una particolare concentrazione sugli attention mechanism e sul loro effetto in termini di prestazioni e interpretabilità, per l'estrazione di *drug-drug interactions* (DDI) dal corpus DDI-2013. Il lavoro su questo dataset è stato descritto anche in [70] e [71]. Il codice con cui sono stati svolti gli esperimenti è disponibile online[1].

## 5.1 Descrizione del problema e dei dati

Nell'ambito del NLP, l'estrazione di DDI è un task di Relation Extraction (vedi Sezione 2.1.3): date due entità corrispondenti ai due farmaci, bisogna riconoscere se nel testo è specificata una relazione tra di essi: la reazione avversa data dalla loro assunzione contemporanea.

Il dataset DDI-2013 [35] si compone di frasi in lingua inglese estratte dalla letteratura medica, completate dalle annotazioni *gold standard* dei nomi dei farmaci e dalle etichette che segnalano, per ogni possibile coppia di farmaci, se nel testo è presente una relazione o meno e di che tipo.

Dal punto di vista del machine learning, questo task è assimilabile a una classificazione. Infatti, è possibile considerare ogni possibile coppia di farmaci come un'istanza e il tipo di relazione come etichetta da associare al testo. In totale, ci sono 5 classi:

- **unrelated**, se non è presente alcuna interazione tra i due farmaci;
- **effect**, se il testo descrive un effetto collaterale dovuto alla DDI;
- **advise**, se il testo consiglia di non assumere i due farmaci contemporaneamente;
- **mechanism**, se il testo descrive come l'assorbimento di un farmaco venga danneggiato dalla contemporanea assunzione di un altro;
- **int**, se il testo descrive genericamente un'interazione tra i due farmaci.

[1] https://github.com/lucaputelli/ddi_attention_experiments

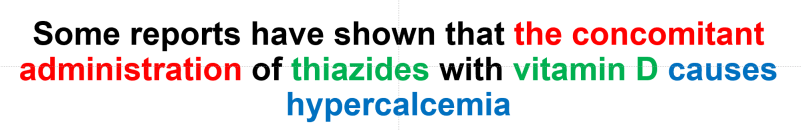


Figura 5.1: Esempio di una frase del corpus DDI-2013. In verde, i nomi dei farmaci, in rosso le parole che indicano l'assunzione in contemporeanea e in blu l'effetto collerateralе causato.

| | Training Set | Test Set |
|---|---|---|
| unrelated | 22474 | 4461 |
| effect | 1685 | 360 |
| mechanism | 1316 | 302 |
| advise | 826 | 221 |
| int | 188 | 96 |

Tabella 5.1: Numero di frasi per classe appartenenti al training set o al test set DDI-2013

Questo corpus è composto da singole frasi, non da interi documenti. In ogni frase può essere presente un numero $n$ di farmaci molto variabile (da uno fino anche a oltre venti), da cui è poi necessario generare tutte le possibili $\binom{n}{2}$ coppie-istanze. Ciascuna istanza può appartenere a una sola delle 5 classi, per cui possiamo definire questo task come una classificazione multiclass.

Nella Figura 5.1 possiamo vedere un esempio di una frase estratta dal corpus. I due farmaci, *thiazides* e *vitamin D*, assunti nello stesso periodo di tempo (*concomitant administration*) danno luogo all'ipercalcemia (*causes hypercalcemia*), ovvero un livello troppo alto di calcio nel sangue.

Nella Tabella 5.1 possiamo vedere il numero di frasi, nel training set e nel test set, per ogni classe. Si può immediatamente notare come quelle unrelated siano la stragrande maggioranza e come la classe *int* abbia un numero estremamente limitato di esempi di training, solo 188 su oltre $20,000$. La lunghezza media delle frasi è di 50 token, e oltre l'85% delle frasi è composta da meno di 100 token, mentre il 25% circa ne ha meno di 30.

## 5.2 Preparazione del dataset

Per il nostro task di Relation Extraction, la trasformazione dalle frasi e la preparazione delle istanze di classificazione è una procedura che si articola nelle seguenti fasi:

1. **Tokenizzazione e POS-tagging**: dato che siamo in presenza di un testo biomedico anche molto complicato, in cui i nomi propri dei farmaci possono comprendere anche più di una decina di token, questa fase è estremamente delicata. Per questo motivo, il tokenizzatore standard appartenente al modello "en_core_web_sm" del tool spaCy è stato modificato allo scopo di dividere il più possibile il testo in ingresso. Caratteri come trattini, barre e parentesi sono stati forzatamente isolati come token singoli. Solo successivamente, e in base alle annotazioni gold standard (che forniscono l'intervallo di caratteri che identifica il nome del farmaco), i token che formano i nomi dei farmaci vengono riunificati in un'unica entità. Questo evita che, a causa di errori di tokenizzazione, si abbia un token che contenga dei caratteri appartenenti al nome del farmaco e altri no. Successivamente alla tokenizzazione, spaCy assegna anche il part-of-speech tag ai singoli token.

2. **Sostituzione**: data una coppia di farmaci, l'istanza del problema di relation extraction si ottiene sostituendo opportunamente ai nomi della coppia due termini standard: `PairDrug1` e `PairDrug2` e ai farmaci restanti il generico termine. Questo permette di differenziare il testo di tutte le istanze e ha anche un altro vantaggio. Infatti, se in una frase sono presenti più coppie con classificazioni diverse, con questo procedimento ogni coppia sarà associata ad un testo diverso, evitando quindi che il modello veda esattamente la stessa frase ma con etichette diverse. Inoltre, i nomi propri dei farmaci sono un'ulteriore complicazione (in quanto termini altamente specifici) non necessaria, dato che l'interazione è eventualmente descritta nella parte restante della frase. Nel caso la coppia sia formata da due menzioni diverse dello stesso far-

maco, alle due menzioni viene sostituito il termine `NoPair` per indicare immediatamente che non è presente nessuna interazione.

3. **Calcolo dello shortest dependency path**: a sostituzione già avvenuta, il tool spaCy viene utilizzato anche per calcolare l'albero sintattico della frase, ovvero una struttura dati in cui ad ogni nodo corrisponde una parola e ad ogni arco la relazione grammaticale di dipendenza tra una parola e l'altra. Prendendo l'esempio in Figura 5.1, dal predicato verbale *causes* è presente un arco *dobj* verso *hypercalcemia* che è il complemento oggetto e uno di tipo *subj* verso *administration* che è il soggetto. Viene calcolato quindi, all'interno dell'albero sintattico in versione non orientata, il cammino minimo (*shortest dependency path)* tra `PairDrug1` e `PairDrug2`, ovvero la sequenza di parole che, nella struttura della frase, collegano un farmaco all'altro e che quindi possono descrivere l'interazione.

### 5.2.1 Il filtro delle negative

Nel corpus DDI-2013 oltre l'80% delle istanze appartiene alla classe unrelated, ovvero non descrive nessuna interazione tra farmaci (da qui in avanti, queste istanze saranno anche chiamate *negative*). Lo sbilanciamento del dataset è un problema classico del machine learning in quanto questi hanno la tendenza di predire la classe maggiormente frequente nel dataset. Infatti, un modello del tutto non addestrato può facilmente raggiungere oltre l'80% di accuracy sempre predicendo la classe unrelated e, data un'istanza a caso, è molto più probabile sia negativa piuttosto che rappresenti un'interazione. Questo, chiaramente, porta basse pretazioni per le classi meno frequenti che, nel nostro dataset, tuttavia sono proprio quelle più interessanti [97].

Per mitigare questo problema, e visto che studi precedenti [14, 43] hanno dimostrato come ridurre il numero di istanze negative migliori le prestazioni anche per questo specifico corpus, abbiamo implementato un meccanismo per filtrare una parte delle istanze negative.

Prima di tutto, dato che lo scopo di questo progetto è scoprire interazioni tra farmaci diversi, se la coppia è formata da due menzioni esattamente iden-

tiche dello stesso farmaco, oppure i due nomi differiscono di una sola lettera come nel caso del cambio da singolare a plurale (ad esempio *antidepressant* e *antidepressants*) viene immediatamente identificata come negativa.

Oltre a questo, possiamo filtrare un'istanza se due farmaci compaiono in una struttura di coordinazione, come ad esempio in un elenco. Se prendiamo ad esempio la frase "The majority of patients in RA clinical studies received one or more of the following concomitant medications with ORENCIA:*MTX, NSAIDs, corti-costeroids, TNF blocking agents, azathioprine, gold, hydroxychloroquine, lefluno-mide, sulfasalazine, andanakinra*", i nomi di farmaco in corsivo sono in una struttura coordinata ed è estremamente improbabile che possano avere una relazione tra loro (la potranno avere invece con altri elementi nella frase fuori dall'elenco).

Mentre tuttavia lavori come [43] e [49] utilizzano un approccio basato specificatamente su questo dataset, come l'impiego di particolari espressioni regolari, la nostra scelta per il filtraggio delle negative è di tipo generale, al fine di creare un meccanismo valido anche per altri contesti di relation extraction. Per filtrare le coppie di farmaci che appaiono in una struttura di coordinazione quindi viene sfruttato l'albero sintattico: se il dependency path calcolato precedentemente contiene esclusivamente nomi di farmaci e congiunzioni, l'istanza viene filtrata.

Questo filtraggio è applicato sia per il training set che per il test set. Nel primo caso, le coppie di farmaci con lo stesso nome o in una struttura di coordinazione vengono semplicemente escluse dagli esempi con cui l'algoritmo viene addestrato. Per il test set, chiaramente nessuna di esse può essere esclusa: tutte le istanze riconosciute negative dal filtro vengono invece classificate come unrelated e verranno poi valutate alla luce delle etichette di classificazione del gold standard.

### 5.2.2 Creazione dell'input

L'input dei modelli di machine learning per l'estrazione di interazioni tra farmaci è quindi composto da tre elementi: il word embedding, il POS embedding e le offset features.

Per ottenere il word embedding, ovvero un vettore di 200 numeri reali, è stato fatto un fine-tuning sul modello disponibile in [55]. Questi word vectors sono stati infatti pre-allenati con un corpus di tipo biomedico, con documenti estratti da PubMed e PMC, per cui forniscono già una buona base per rappresentare il significato di termini medici per la lingua inglese. Tuttavia, essendo il nostro dataset fortemente specializzato sulla farmacologia, abbiamo eseguito nuovamente Word2Vec, utilizzando i word vectors già disponibili come base di partenza, sul nostro training set.

Il POS-embedding invece è stato calcolato solamente utilizzando il corpus DDI-2013 ed è composto da vettori di 20 posizioni ottenuti applicando l'algoritmo Word2Vec alla sequenza dei PoS-tag, nello stesso ordine delle parole.

Oltre a questi due componenti, che sono degli standard nelle applicazioni di deep learning e NLP, abbiamo aggiunto le offset features, $D_1$ e $D_2$ [104]. Data una parola all'interno di un'istanza, $D_1$ e $D_2$ si calcolano rispettivamente come la distanza, in termini di parole, da `PairDrug1` e `PairDrug2`. Per esempio, nella frase *Intravenous PairDrug1 was shown to double the bioavailability of oral PairDrug2*, la parola *shown* ha $D_1$ uguale a 2, e $D_2$ uguale a -7, cioè compare sette parole prima di `PairDrug2`. Essendo questi numeri, nel caso di frasi di oltre 100 parole, potenzialmente anche molto grandi, le offset features vengono anch'esse rappresentate da un vettore di numeri reali, stavolta di lunghezza 3. Per calcolarli, non viene utilizzato Word2Vec: questa rappresentazione è infatti appresa contestualmente all'addestramento del modello con un *embedding layer*.

Dato che il livello LSTM prevede un input di dimensione fissa e che la frase di lunghezza massima comprende 150 parole, è stato scelto di creare una matrice di dimensione $150 \times 226$ (200 di word embedding, 20 di PoS embedding, 6 per le offset features) come input. Nel caso una frase sia composta da meno di 150 parole, vengono aggiunti vettori di 226 zeri per uniformare la dimensione finale.

## 5.3 Modelli utilizzati

Essendo questa competizione aperta dal 2013, in letteratura è possibile trovare approcci classici [14] così come basati su reti convoluzionali [75] o su reti ricorrenti [43]. Il focus del nostro lavoro, descritto in [70] e [71], è principalmente sulle diverse forme di Attention Mechanism e su come influiscono sui risultati.

Tutti i modelli utilizzati hanno quindi in comune:

- la presenza di almeno un livello LSTM bidirezionale ad elaborare la sequenza delle parole;
- la presenza di un attention mechanism, in particolare uno di quelli descritti nella Sezione 3.4;
- l'ultimo livello: un livello fully-connected con 5 neuroni e attivazione softmax che fornisce la classificazione finale.

Le differenze tra un modello e l'altro sono esclusivamente nella scelta dell'attention utilizzato e nella preparazione dell'input.

### 5.3.1 Modello con Self-interaction attention

Come visto nella Sezione 3.4, il Self-interaction attention [108] è un attention mechanism abbastanza diverso dagli altri. Infatti, se nella versione classica [4] l'attention mechanism calcola i pesi per ogni parola e fornisce poi un vettore finale di rappresentazione del documento, il Self-interaction attention considera tutte le possibili coppie di parole e assegna un peso alla loro interazione. Data quindi una frase di $N$ parole e un livello LSTM con $D$ neuroni, mentre l'attention mechanism classico fornisce in output un vettore di dimensione $D$, il self-interaction attention produce una matrice $S$ di dimensione $N \times D$, ovvero un vettore di dimensione $D$ per ogni parola, ottenuto dalla media pesata dei pesi di tutte le interazioni:

$$s_i = \sum_{k=1}^{K} \alpha_{i,k} u_i \tag{5.1}$$

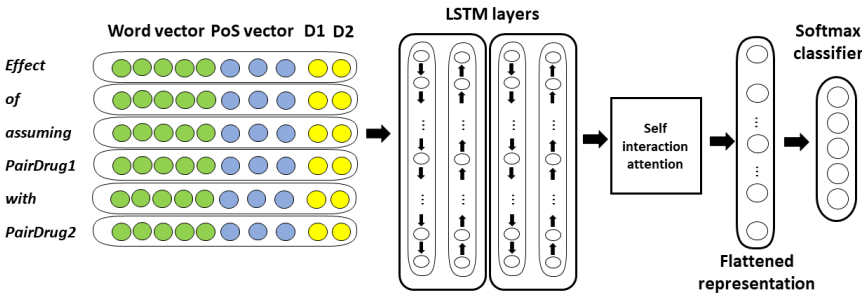


Figura 5.2: Modello con Self-Interaction Attention.

in cui $\alpha_{i,k}$ è il peso dell'interazione tra la parola $i$-esima e la parola $k$-esima e $u_i$ si calcola come:

$$u_t = \tanh(W_a h_t + b_a) \tag{5.2}$$

ovvero applicando un livello fully-connected con attivazione tanh all'uscita del livello LSTM, che produce infatti $h_t \in \mathbb{R}^D$ con $t \in [1, N]$.

Mentre l'approccio originale in [108] prevede di ridurre ulteriormente $S$ a un vettore attraverso un'operazione di pooling [83], nel nostro modello abbiamo deciso di preservare l'intera matrice che viene poi elaborata dal livello di output creando un unico vettore di lunghezza $D * N$ concatenando tutti quelli che compongono la matrice (*flattened representation).*

Una schematizzazione del modello in forma grafica è visibile in Figura 5.2. La configurazione migliore che è stata trovata prevede, per questo modello, 2 livelli di LSTM bidirezionali.

Per confrontare l'effetto del Self-Interaction Attention, abbiamo costruito altri tre modelli:

- un modello composto solo da livelli LSTM (**No Attention)**, passando quindi solamente l'uscita dell'ultima cella $h_N$ al livello fully-connected con attivazione softmax;
- un modello comprendente anche l'**Attention** mechanism classico [4];
- un modello comprendente la versione **Context Attention** [101] (vedi anche Sezione 3.4).

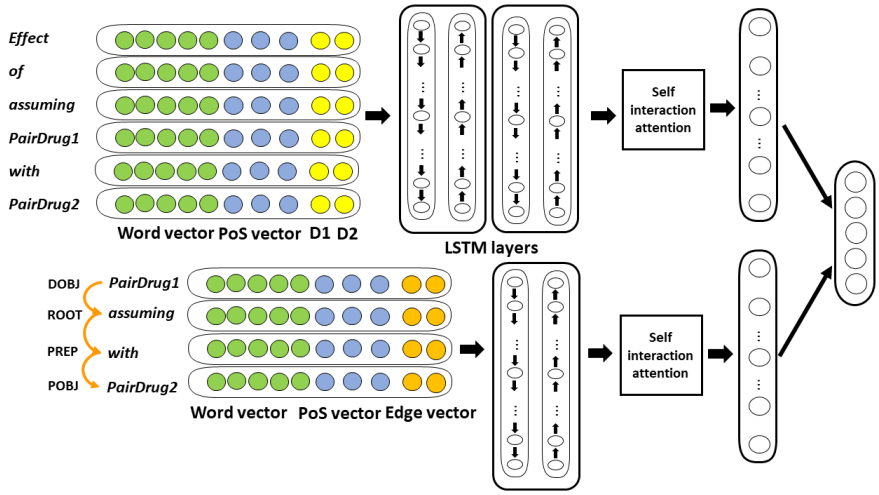


Figura 5.3: Modello con Self-Interaction Attention e inclusione dello Shortest Dependency Path

### 5.3.2 Modello con Shortest Dependency Path

L'inclusione delle relazioni grammaticali e della struttura della frase in un modello di deep learning può essere vista come un'ulteriore aggiunta di conoscenza che potrebbe essere utile per la comprensione del testo.

In particolare, è ragionevole supporre che il cammino minimo nell'albero sintattico tra `PairDrug1` e `PairDrug2` contenga informazioni riguardo alla loro interazione. Lo shortest dependency path quindi, oltre a essere utilizzato per il filtraggio delle negative, in questo modello viene anche passato al modello come input aggiuntivo.

Nella Figura 5.3 si può vedere il modello a due canali provato per l'inclusione dello shortest dependency path (SDP). Accanto a un primo canale esattamente uguale a quanto descritto precedente, ne viene affiancato un altro con le seguenti caratteristiche:

- L'input è formato solamente dallo SDP, seguendo l'ordine del cammino e non quello della frase.
- Al posto delle offset features, è stata inclusa una rappresentazione embedded della relazione grammaticale tra una parola e l'altra. Ad esempio, se `PairDrug1` è complemento oggetto ed è collegato al predicato

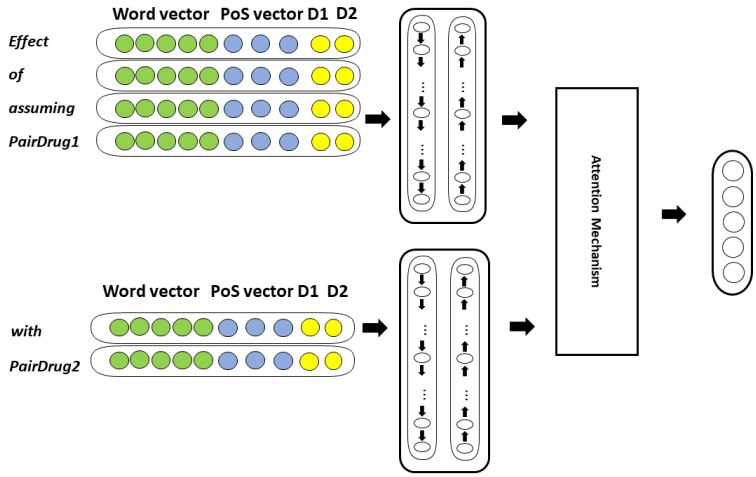


Figura 5.4: Modello a due canali, in versione semplificata. Il primo canale, nella parte sopra della figura, prende in ingresso le prime 60 parole. Il secondo canale, nella parte sotto, le restanti.

verbale dall'etichetta *DOBJ*, questa etichetta viene rappresentata come un vettore di lunghezza 10. Il calcolo di rappresentazione è del tutto analogo a quello per il PoS embedding.

- Viene utilizzato solamente un livello LSTM, in quanto l'input è molto più breve rispetto all'intera frase.
- La rappresentazione finale della frase è data dalla concatenazione delle flattened rappresentations prodotte dai self-interaction attention dei due canali.

### 5.3.3 Modello a due canali

Data la disparità della lunghezza delle frasi come descritto nella Sezione 5.1) in cui la maggior parte è composta da al massimo 60 parole e da alcune problematiche riscontrate nel testing dei modelli precedenti (come vedremo in seguito) proprio nelle frasi più corte, è stato deciso di creare un altro modello a due canali: uno destinato alle prime 60 parole, e che quindi sarà particolarmente responsabile per le classificazione delle frasi più corte, un

altro alle seguenti 60 (la massima lunghezza della frase è 150 ma quelle oltre i 120 sono meno del 5%) “attivato" solamente per le frasi più lunghe.

Come è visibile in Figura 5.4, sia la prima metà che la seconda metà della frase sono elaborati da un livello LSTM di dimensione $D$. Il risultato di questi livelli, ovvero $h_i$ con $i \in [1, 60]$ per il primo e $h_i$ con $i \in [61, 120]$ vengono concatenati verticalmente, creando quindi una matrice $120 \times D$ che viene passata a un unico attention mechanism, responsabile quindi di calcolare la rappresentazione finale della frase intera. Il *context vector* prodotto dall'attention viene quindi elaborato dal solito livello fully-connected con attivazione softmax, esattamente come negli altri modelli.

L'idea principale che sta alla base della realizzazione di questo modello è che il rumore presente nelle frasi più lunghe (ad esempio lunghi elenchi di farmaci) influisca non solo sulla classificazione di queste ultime ma, data la necessità della rete LSTM di adattarsi anche a questi casi, anche l'elaborazione delle frasi più corte. Separando i due canali quindi è possibile che il primo, sui primi 60 token, possa maggiormente specializzarsi e ottenere risultati migliori. La presenza poi di un unico attention mechanism permette al sistema di riunire le due computazioni e considerarle come parte di un'unica frase.

## 5.4 Prestazioni dei modelli basati su LSTM

La competizione fornisce un test set, fisso e uguale per tutti, con cui valutare le prestazioni dei modelli e indica le metriche con cui confrontarsi.

L'accuracy del modello, vista la presenza di una fortissima maggioranza di istanze della classe unrelated (cioè negative), non è significativa. Invece, per ogni classe si calcolano *precision* (P), *recall* (R) e *F-Score* (per una definizione di queste tre metriche si veda la Sezione 4.4).

La competizione fissa inoltre come metriche, per un confronto più rapido delle prestazioni complessive dei modelli, la media aritmetica pesata delle precision, recall e F-Score delle 4 classi che identificano un'interazione (effect, mechanism, advise e int), escludendo quindi la classe unrelated che è di poco

| Input | No Attention | Context-Att | Attention | Self-Int-Att |
|---|---|---|---|---|
| Word | 64.44 | 65.32 | 66.60 | **69.72** |
| Word+PoS | 65.37 | 65.20 | 67.57 | **68.95** |
| Word+PoS+Offset | 60.67 | 65.82 | 69.47 | **70.88** |

Tabella 5.2: Recall media, per le quattro classi positive, con diversi attention mechanism e diverse configurazioni di input.

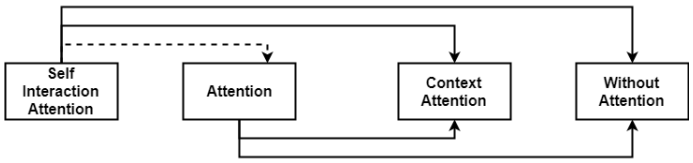


Figura 5.5: Confronto, attraverso il test di Friedman, tra diverse recall per la configurazione migliore di input (Word+PoS+Offset). La linea continua indica che è un modello è significativamente migliore con confidenza superiore al 99%, quella tratteggiata con confidenza superiore al 95%.

interesse per lo scopo di relation extraction (e che ottiene sempre metriche superiori al 90%).

Per trovare la configurazione dei modelli più promettenti è stata eseguita una Random Search [7], con particolare attenzione verso il numero di livelli di LSTM (uno o due), il numero di neuroni di questi livelli, il dropout e il recurrent dropout. Per questa ottimizzazione degli iperparametri, è stato impiegato un validation set composto da circa il 15% dei documenti del training set.

### 5.4.1 Confronto tra attention

L'effetto più evidente dell'applicazione del Self-Interaction Attention è il miglioramento della recall. Nella Tabella 5.2 possiamo vedere la recall totale del modello con gli iperparametri migliori, calcolata come media delle recall delle classi effect, advise, mechanism e int, variando solamente l'attention mechanism posto dopo il livello LSTM. Tutti e quattro i modelli sono stati testati con tre diverse configurazioni di input: quella con il solo word em-

| | Effect | | | | Mechanism | | | |
|---|---|---|---|---|---|---|---|---|
| Input | No | C-Att | Att | Self-Int | No | C-Att | Att | Self-Int |
| W | 0.68 | 0.71 | **0.72** | 0.70 | 0.69 | 0.72 | 0.72 | 0.70 |
| W+P | 0.67 | 0.70 | 0.70 | 0.69 | 0.71 | 0.73 | 0.74 | 0.70 |
| W+P+O | 0.65 | 0.70 | 0.70 | 0.69 | 0.68 | 0.73 | 0.74 | **0.76** |
| | Advise | | | | Int | | | |
| Input | No | C-Att | Att | Self-Int | No | C-Att | Att | Self-Int |
| W | 0.77 | 0.71 | 0.74 | 0.78 | 0.53 | 0.49 | 0.45 | 0.45 |
| W+P | 0.78 | 0.73 | 0.77 | 0.77 | **0.55** | 0.50 | 0.45 | 0.43 |
| W+P+O | 0.74 | 0.75 | **0.79** | 0.78 | 0.50 | 0.52 | 0.50 | 0.49 |

Tabella 5.3: Confronto tra le F-Score dei diversi modelli, con tutte e tre le configurazioni di input. Per ogni classe, la migliore F-Score è segnata in grassetto.

bedding, quella aggiungendo anche il PoS embedding e quella completa, cioè compresa delle offset features.

Il comportamento è simile in tutti e tre i casi: le prestazioni in termini di recall media sono più alte per il Self-Interaction Attention di più un punto percentuale. Questa differenza è anche statisticamente significativa secondo il test di Friedman [23], con una confidenza sempre superiore al 95% come visibile nella Figura 5.5. Riteniamo questa differenza in termini di recall particolarmente importante, in quanto è la metrica più influenzata dalla presenza della classe di maggioranza unrelated. Dato che, rispetto ad altri approcci [43, 49], il nostro filtro delle negative è più generale, possiamo vedere il Self-Interaction Attention come un ulteriore metodo per ridurre l'effetto delle istanze negative sulle prestazioni.

Nella Tabella 5.3 possiamo confrontare le F-Score medie delle 4 classi positive, cambiando attention mechanism e per le tre diverse configurazioni di input. I punti principali desumibili da questo confronto sono:

- Il miglioramento delle prestazioni medie includendo anche il PoS embedding e le offset features, per i modelli che includono un attention mechanism. Il modello senza attention, al contrario, non riesce a trarre beneficio dalle offset features.

| Input | Effect | Mechanism | Advise | Int |
|---|---|---|---|---|
| Word | 0.68 | 0.72 | 0.76 | 0.50 |
| Word+Tag | 0.69 | 0.69 | 0.74 | 0.53 |
| Word+Tag+Edge | 0.66 | 0.71 | 0.70 | 0.45 |

Tabella 5.4: Prestazioni del modello con inclusione dello Shortest Dependency Path

- Per le classi effect e mechanism, il miglioramento netto dato dall'inclusione di un attention. Le prestazioni sono invece abbastanza simili per la classe advise (tranne per la variante Context Attention che ottiene risultati peggiori) e addirittura peggiori per la classe int.

- Le prestazioni dell'attention classico e del Self-Interaction Attention, che come visto precedentemente è significativamente superiore in termini di recall, sono molto simili in termini di F-Score. La variante Context Attention invece, pure con più complessità rispetto all'attention classico, ottiene risultati quasi sempre inferiori.

Nella Tabella 5.4 possiamo vedere le prestazioni dettagliate del modello con l'inclusione dello shortest dependency path, cioè con due canali formati da livelli LSTM e self-interaction attention messi in parallelo.

L'informazione aggiuntiva della struttura grammaticale non ha prodotto risultati e anzi, la maggiore complessità del modello ha prodotto un decremento delle prestazioni in termini di F-Score. Mentre sono rimasti mediamente gli stessi per la classe effect e leggermente migliori per int (che comunque, essendo una classe composta da poco più di 150 elementi di test è molto sensibile a minuscole variazioni nella predizione), il calo è abbastanza netto per mechanism (0.72 contro 0.76) e advise (0.76 contro 0.78).

Anche l'inclusione, tramite un vettore di 10 numeri reali, delle etichette tra un nodo e l'altro rappresentanti la relazione grammaticale tra due parole, non ha prodotto gli effetti sperati e anzi, ha fatto calare le prestazioni per tutte e quattro le classi rispetto alle altre due configurazioni di input.

Ricordiamo comunque che l'albero sintattico e quindi anche lo shortest dependency path non sono gold standard ma forniti da modelli di machine

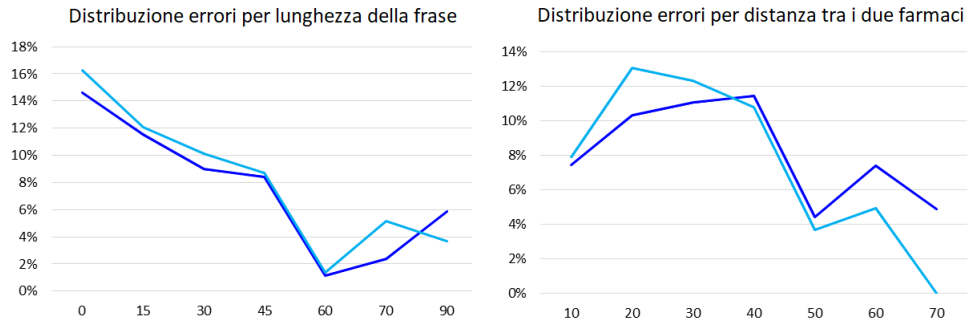


Figura 5.6: Distribuzione degli errori per il modello senza attention mechanism (in azzurro) e con self-interaction attention (in blu) rispetto alla lunghezza della frase (a sinistra) e alla distanza, in termini di parole, tra un farmaco e l'altro (a destra).

learning anch'essi e quindi soggetti ad errore, specialmente nel caso di un linguaggio complesso e con termini sconosciuti come quello biomedico.

### 5.4.2 Analisi degli errori

Per capire pregi e difetti del nostro approccio, abbiamo analizzato gli errori commesso dal modello con self-interaction attention. Il problema del nostro sistema rimane ancora la distinzione tra le istanze negative e le altre classi: infatti, l'83.5% degli errori di predizione riguardano proprio questa casistica. Va comunque ricordato che la presenza del self-interaction mechanism diminuisce questo errore del 14.6%. Al contrario, la distinzione tra le classi effect, advise e mechanism positive ottiene buoni risultati, con soli 30 errori su 861 istanze.

Nella Figura 5.6 possiamo vedere come si distribuiscono gli errori fatti dal modello con e senza self-interaction attention in termini di lunghezza della frase e di distanza, calcolata come numero di parole, tra un farmaco e l'altro. Nonostante in linea teorica dovrebbero essere quelle più semplici da analizzare, le difficoltà maggiori si trovano per le frasi medio-corte e con una distanza non troppo grande tra un farmaco e l'altro. Questo può essere dovuto a vari fattori: innanzitutto, come scritto nella Sezione 5.1, solo il 25% delle frasi si compone di meno di 30 token e quindi è comprensibile che

il modello ottenga prestazioni migliori dove ha una percentuale maggiore di esempi. In secondo luogo, molte frasi lunghe non lo sono perché descrivono ragionamenti intricati e complessi, ma bensì per lunghi elenchi di farmaci. Ad esempio nella frase "*Drugs that have been reported to diminish oral anticoagulant response, ie, decreased prothrom-bin time response, in man significantly include: adrenocortical steroids; alcohol; antacids; antihistamines; barbiturates; carbamazepine; chloral hydrate; chlordiazepoxide; cholestyramine; diet high in vitamin K; diuretics; ethchlorvynol; glutethimide; griseofulvin; haloperidol; meprobamate; oral contraceptives; paraldehyde; primidone; ranitidine; rifampin; unreliable prothrombin time determinations; vitamin C; warfarin sodium under-dosage*", di oltre 90 token (ricordiamo che vengono contati anche i segni di punteggiatura), sono presenti ben 24 farmaci e quindi 276 coppie di cui predire la predizione, di cui la stragrande maggioranza negative. Dato che le istanze negative sono molto facili da predire, visto che sono la classe maggiormente rappresentata nel training set, questo abbassa notevolmente il tasso di errore per le frasi più lunghe, che si assesta sul 5% (nonostante l'indubbia complessità di gestire tutti questi token), rispetto a una percentuale doppia per le frasi più corte. L'attention mechanism non influisce sulla distribuzione dell'errore rispetto alla lunghezza della frase ma ha un effetto sulla distribuzione per quelle istanze che hanno una distanza tra i due farmaci compresa tra le 10 e le 30 parole, diminuendo nettamente la percentuale di errore, alzandola tuttavia per le frasi con distanza elevata.

La classe più problematica, come si può vedere nelle Tabelle 5.3 e 5.4, è indubbiamente int, che è fortemente minoritaria nel dataset. Infatti, a int appartengono solamente poco più dello 1.5% delle istanze del dataset. Va tuttavia fatto notare che, mentre per le altre classi l'errore più comune è non aver riconosciuto l'interazione e quindi aver predetto unrelated, per la classe int le istanze sbagliate vengono perlopiù etichettate come effect. Riteniamo quindi che il sistema riconosca una sorta di interazione e tenda quindi a classificarla come appartenente alla classe di maggioranza tra le non negative.

| | Modello a due canali | | | Self-int attention | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| Effect | 0.72 | 0.77 | 0.75 | 0.66 | 0.73 | 0.69 |
| Mechanism | 0.81 | 0.64 | 0.71 | 0.80 | 0.73 | 0.76 |
| Advise | 0.91 | 0.71 | 0.79 | 0.75 | 0.81 | 0.78 |
| Int | 0.81 | 0.34 | 0.48 | 0.82 | 0.34 | 0.49 |

Tabella 5.5: Precision, Recall e F-Score per le classi della DDI del modello a due canali.

### 5.4.3 Prestazioni del modello a due canali

Proprio per la difficoltà nelle frasi medio-corte, è stato deciso di costruire il modello a due canali con un livello LSTM che prendesse in considerazione solo le prime 60 parole e un altro per le seguenti, utilizzando l'input migliore disponibile, ovvero quello composto da word embedding, PoS embedding e offset features.

In termini di F-Score totale, questa scelta ha pagato: infatti, mentre il modello self-interaction attention ottiene una F-Score del 72.2%, il modello a due canali riesce ad arrivare al 72.9% con una differenza significativa soprattutto nella precision. Nella Tabella 5.5 è possibile vedere le prestazioni dettagliate di questo modello.

I comportamenti sono abbastanza diversi: il modello a due canali ottiene una precision migliore tranne che per la classe int (in cui è molto simile), con addirittura un miglioramento di 16 punti percentuali per la classe advise. La recall invece tende a peggiorare, a conferma del fatto che comunque il self-interaction attention riesce a sopperire parzialmente allo sbilanciamento del dataset.

I tentativi di coniugare il self-interaction attention con il modello a due canali, sia attraverso un unico attention dopo aver riunito gli output delle due LSTM che attraverso due meccanismi separati, hanno ottenuto prestazioni peggiori dell'attention classico.

| | P(%) | R(%) | F(%) |
|---|---|---|---|
| UTurku (SVM) [8] | 73.2 | 49.9 | 59.4 |
| FBK-irst (SVM) [14] | 64.6 | 65.6 | 65.1 |
| Zeng SCNN [104] | 72.5 | 65.1 | 68.6 |
| Liu CNN [49] | 75.7 | 64.7 | 69.8 |
| Quann LSTM [75] | 76.0 | 65.3 | 70.2 |
| **Context-Att** | **75.9** | **65.8** | **70.5** |
| Sahu LSTM [43] | 73.4 | 69.7 | 71.5 |
| **Self-Int** | **73.0** | **70.9** | **71.9** |
| Yi GRU [102] | 73.7 | 70.8 | 72.2 |
| **Attention** | **75.6** | **69.5** | **72.4** |
| Zhang LSTM [107] | 74.1 | 71.8 | 72.9 |
| **Due canali** | **80.0** | **68.9** | **72.9** |

Tabella 5.6: Confronto in termini di precision (P), recall (R) e F-Score (F) media del modello, con gli altri metodi presenti in letteratura, ordinati per F-Score. I nostri modelli sono in grassetto.

### 5.4.4 Confronto con lo stato dell'arte e riproducibilità

Nella Tabella 5.6 possiamo confrontare i modelli illustrati precedentemente con lo stato dell'arte.

Il modello a due canali raggiunge lo stato dell'arte in termini di F-Score, dovuto principalmente a una precision notevolmente alta. Va sicuramente segnalata anche la recall del modello Self-Interaction Attention, tuttavia superata dal modello descritto in [107]. Le prestazioni invece dell'attention classico (72.4 % di F-Score) e soprattutto della variante Context Attention, che è inferiore di un punto rispetto alla media degli approcci con LSTM, non sono particolarmente notevoli. Le prestazioni rispetto ai metodi più classici basati su Support Vector Machine [14, 8] e alle reti convoluzionali più semplici [49, 104] sono decisamente superiori.

Per il dataset DDI-2013, almeno a nostra conoscenza, non sono stati svolti esperimenti con Graph Neural Networks o Graph Convolutional Networks [100], ovvero modelli di rete neurale particolarmente adatti per analizzare l'informazione contenuta nell'albero sintattico di una frase [106]. Alcuni nostri esperimenti iniziali, seguendo l'idea di affiancare le Graph Convolutional

Networks alle LSTM presentata in [53], non hanno prodotto risultati degni di nota. Tuttavia, anche alla luce di nuove idee che coniugano l'attention mechanism con la struttura a grafo [48, 93], riteniamo che un confronto tra i risultati da modelli classici come quelli basati su reti ricorrenti e quelli basati su Graph Neural Networks possa essere un'interessante linea di ricerca per il futuro.

Il confronto tra il nostro approccio e il lavoro descritto in [109] di Zheng et al. è stato trattato in maniera approfondita in [71]. Il loro modello utilizza le stesse feature di input ma applica direttamente ai word vector un attention mechanism proposto da loro, il *drug oriented input attention*, che di fatto calcola direttamente il peso dell'interazione $\alpha_i^j$ tra una parola $i$ con word embedding $w_i$ e un farmaco, rappresentato dall'embedding $w_{d_j}$ con $j \in \{1, 2\}$:

$$\alpha_i^j = softmax(\frac{w_i \times w_{d_j}}{m}) \tag{5.3}$$

in cui $\times$ rappresenta l'operazione di prodotto scalare tra i due vettori e $m$ è la lunghezza del word embedding. Il peso totale dell'interazione con i due farmaci $\alpha_i$ è dato dalla media dei due pesi $\alpha_i^1$ e $\alpha_i^2$. Successivamente, vengono calcolati dei nuovi word embedding $u_i$ delle parole alla luce della loro interazione con i farmaci moltiplicando $w_i$ con $\alpha_i$:

$$u_i = w_i * \alpha_i \tag{5.4}$$

Questi nuovi word embedding vengono quindi concatenati al PoS embedding e alle offset features ed elaborati dal consueto livello LSTM bidirezionale. Nel modello non è presente alcun altro attention mechanism al di fuori di quello drug-oriented posto subito dopo l'input.

Abbiamo replicato questa architettura, compresa di drug-oriented input attention, utilizzando la nostra rappresentazione word embedding come base e non siamo riusciti ad ottenere risultati né simili a quelli riportati nell'articolo, nè superiori ai nostri e anzi, provando tutte le configurazioni di input, l'aggiunta del loro meccanismo non ha migliorato le prestazioni.

Dato che, a parte questa innovazione, la loro architettura è molto semplice (non è presente alcun attention mechanism), né ci sono più canali come

in [75] o [107], né è descritto un filtraggio delle negative particolarmente efficace, riteniamo che questi risultati siano dovuti o a una rappresentazione word embedding iniziale decisamente migliore (l'articolo non riporta se è stato utilizzato un modello pre-allenato o costruito ad hoc) oppure a qualche tecnica di pre-processing non adeguatamente descritta.

Gli approcci basati su reti convoluzionali molto profonde descritti in [19] di Dewi et al. e [86] di Sun et al. presentano problemi di riproducibilità ancora più evidenti. I loro risultati, superiori alla media di oltre 10 punti di F-Score, sono dovuti a un'architettura molto più complessa di tutte le altre. Nel primo, vengono utilizzati 5 diversi modelli di word embedding di dimensione 200 precedente allenati con documenti presi da PMC, PubMed, entrambe le fonti, Wikipedia e PubMed insieme e MedLine, formando quindi un tensore di dimensione $5 \times 150 \times 200$ in cui 150 è la lunghezza massima della frase. Il tensore viene quindi elaborato da 8 livelli convoluzionali con 7 operazioni parallele di convoluzione, con dimensione della finestra pari a 4, 5, 6, 7, 8, 9 e 10, ognuna delle quali con 400 filtri diversi, seguiti da un'operazione di max pooling. La rete poi aggiunge dropout, regolarizzazione o rumore gaussiano per migliorare le prestazioni. Replicando questa architettura molto complessa, non siamo riusciti né a ottenere i risultati nell'articolo né, soprattutto, ad ottenerne di significativi. Un qualsiasi modello di machine learning applicato a questo task deve innanzitutto superare lo scoglio della classe di maggioranza, ovvero non predire ogni istanza come negativa e *accontentarsi* della buona accuracy; la nostra replica del modello proposto in Dewi et al. non è riuscita a superare questo ostacolo e cominciare l'apprendimento vero e proprio.

La stessa cosa è avvenuta per il modello descritto in [86]. Qui la rete è ben più profonda (fino a 24 livelli convoluzionali) ma senza il parallelismo tra finestre di grandezza diversa, nè il word embedding multiplo (ne viene utilizzato uno ma non viene specificato quale, per cui nel tentativo di replicare l'esperimento abbiamo utilizzato il nostro). Il numero dei filtri per livello è, come spesso accade nelle reti convoluzionali, a raddoppiare progressivamente da 64 a 512. La replica del modello con i parametri messi nell'articolo, tra cui un numero di epoche molto basso (solamente 10), anche in questo caso

non è riuscita ad andare oltre la predizione della sola classe di maggioranza.

Sia per quanto riguarda il modello basato su LSTM che per gli ultimi due basati su CNN, abbiamo contattato gli autori non ricevendo risposta oppure ricevendo (su richiesta ad esempio degli iperparametri o del modello di word embedding utilizzato) solamente risposte evasive.

Replicare un risultato riportato in un articolo non è un problema banale. Negli ultimi anni, con l'esplosione del machine learning e del deep learning per il NLP, il trend per la pubblicazione di lavori scientifici è molto orientato al risultato numerico, talvolta senza una chiara intuizione o spiegazione di come è stato ottenuto [15].

Dato che le prestazioni di un modello di deep learning possono dipendere tanto dall'input e dal pre-processing fatto, quanto dalla configurazione del modello stesso, dagli iperparametri o perfino dal software e dall'hardware utilizzato per l'addestramento, è molto difficile ottenere lo stesso risultato perfino quando gli autori forniscono il codice sorgente della loro applicazione [99].

In molti casi, inoltre, come gli approcci che abbiamo descritto in precedenza e tentato di riprodurre, gli autori non forniscono il codice sorgente e danno solo un'indicazione di massima su come hanno eseguito il training, come hanno configurato il modello, ecc. Anche a fronte di nostra esplicita richiesta, non abbiamo ottenuto informazioni sufficienti per replicare i loro risultati. In uno studio pubblicato in [99] riguardante 397 articoli pubblicati tra il 2011 e il 2016 alla conferenza annuale della Association for Computational Linguistics, una delle più importanti conferenze nell'ambito del Natural Language Processing, nel 41% dei casi gli autori non forniscono alcun riferimento per replicare i loro risultati, nemmeno dopo un'esplicita richiesta.

Lo stesso studio ha ulteriormente selezionato 10 articoli e, partendo dal codice sorgente fornito dagli autori, ha provato a replicare i risultati. Solo in un caso, è stato possibile riprodurre il risultato esatto, mentre per altri tre si è riuscito ad ottenere prestazioni comparabili. A conferma anche di quanto possano contare elementi software e le fasi di pre-processing, l'aggiornamento di MALLET, un famoso tool per NLP, a una versione più recente ha fatto calare le prestazioni di oltre 10 punti: dal 95% di accuracy ottenuto con la

versione originale indicata dagli autori all'83.3% con quella più recente.

Nel lavoro in [17], sono indicati tre tipi di riproducibilità: (i) quella di un valore numerico, come abbiamo appena descritto; (ii) quella di una scoperta, per cui un algoritmo funziona meglio di un altro in determinati contesti oppure che l'inclusione di un componente o di un input aggiuntivo migliora i risultati per certi compiti; (iii) quella di una conclusione, per cui di un'affermazione generale e di ampia portata. L'approccio per la competizione DDI-2013 riportato nel lavoro di Zheng et al. è quindi notevole non solo per il 77.3% di F-Score ma anche per l'introduzione del drug oriented input attention che, a loro dire, avrebbe migliorato i risultati. Nel nostro tentativo di riprodurre questa scoperta, non siamo riusciti ad ottenere alcun miglioramento nei nostri modelli che pure agivano in un contesto estremamente simile, ovvero con lo stesso dataset e utilizzando la stessa architettura con un livello LSTM.

Per quanto riguarda invece i due modelli basati su reti convoluzionali molto profonde citati precedentemente, la netta differenza tra i risultati degli autori, superiori di gran lunga a qualsiasi altro tentativo, e il nostro tentativo di replica che ha prodotto un F-Score dello 0%, desta qualche perplessità. In genere l'addestramento di reti neurali di questa complessità necessita di un training set molto grande, mentre nel corpus DDI-2013 non si arriva alle ventimila istanze di cui molte che variano solo per la coppia di farmaci considerata. Ad esempio, l'articolo che ha introdotto questo modo di procedere per la classificazione di testi [18] utilizza per i propri test dei dataset contenenti tra i centomila e i tre milioni di documenti. Inoltre, i corpus considerati riguardano argomenti generici quali recensioni, articoli di giornale ecc. e quindi con un tasso di difficoltà minore rispetto a quelli dell'ambito biomedico.

## 5.5 Analisi del ruolo dell'Attention

A differenza di quanto fatto per la classificazione dei referti radiologici (vedi Sezione 4.5), per questo dataset non è possibile valutare la qialità della porzione di testo evidenziata dall'attention. Infatti, mentre nel caso dei

referti era possibile fare un confronto con le annotazioni manuali delle espressioni più importanti fatte dagli esperti del dominio, in questo caso le uniche informazioni presenti sono:

- l'annotazione di quali token formano il nome di un farmaco;
- per ogni possibile coppia di farmaci, il valore della classe a cui appartiene.

Nessuna informazione quindi su quali parti del testo indicano la presenza della relazione o meno, con cui sarebbe stato utile confrontare i pesi dell'attention mechanism in un'analisi qualitativa per l'individuazione di un'interpretazione plausibile.

In questa sezione quindi i ragionamenti saranno più a livello quantitativo, cercando di capire come varia il comportamento dell'attention in diversi casi e come questo può essere ricondotto a concetti noti del Natural Language Processing. Il modello di riferimento è quello a singolo canale con l'attention mechanism più semplice definito in [4, 76].

### 5.5.1 L'effetto gate all'interno della frase

Un'altra differenza fondamentale tra questo dataset e quello dei referti radiologici è che siamo in presenza non più di interi documenti, ma di singole frasi. Non è quindi scontato che la funzione di gate, cioè che l'attention tenda non a quantificare l'importanza in termini numerici ma invece a selezionare o scartare in maniera binaria, sia verificata anche per questa applicazione.

Infatti, nella Sezione 4.5.1 abbiamo verificato come la funzione di gate molto spesso si concentri su intere frasi o comunque su parti consistenti di testo. Essendo in presenza, nel dataset DDI-2013, di istanze composte da una sola frase, è chiaro che l'attention, per essere efficace, deve lavorare con maggiore precisione per individuare le parole più importanti. Anche se l'attention si limitasse a selezionare porzioni consistenti di testo, è da verificare quale effetto abbia la netta diminuzione di parole di input sul suo comportamento (la lunghezza media dei documenti di questo dataset è di 60

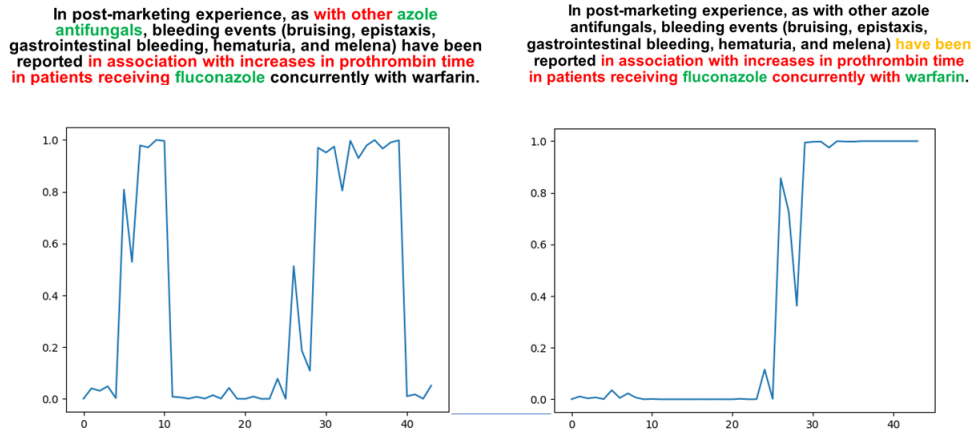


Figura 5.7: Visualizzazione dell'effetto gate per due istanze di test, relative alla stessa frase, per il dataset DDI-2013. In rosso, le parole a cui l'attention assegna un peso normalizzato superiore a 0.8. In arancio, quelle superiori a 0.6.

parole, quella di un referto, come indicato nel capitolo precedente, di oltre 200).

Come nel caso dei referti radiologici, anche in questo caso abbiamo ritenuto (dato che i pesi dell'attention $\alpha_i$ variano in base alla lunghezza della frase, visto che la loro somma dev'essere uguale a uno) di scalarne i valori, ottenendo quindi dei nuovi pesi normalizzati tra 0 e 1 ($w_i$), attraverso la seguente operazione:

$$w_i = \frac{\alpha_i - \alpha_{min}}{\alpha_{max} - \alpha_{min}}$$

Nella parte sinistra della Figura 5.7, è possibile vedere un esempio di frase (piuttosto corta, visto che sono poco più di 40 parole) e dei pesi che sono stati assegnati dall'attention mechanism. Della frase *In post-marketing experience, as with other azole antifungals, bleeding events (bruising, epistaxis, gastrointestinal bleeding, hematuria, and melena) have been reported in association with increases in prothrombin time in patients receiving fluconazole concurrently with warfarin.*, considerando la coppia di farmaci (in verde) *azole antifungals* (*PairDrug1*) e *fluconazole* (*PairDrug2*), l'attention mechanism

evidenzia solamente i due farmaci e le parole immediatamente circostanti (in rosso). Al resto della frase, salvo il singolo token *have*, viene assegnato il peso minimo. Esattamente come accadeva nel caso dei referti radiologici, quasi non esistono pesi intermedi tra il minimo e il massimo. Questa è un'indicazione del fatto che l'attention non lavora solo a livello di frase ma anche per insiemi di poche parole o, potenzialmente, anche per singoli token.

Il fatto che l'effetto dell'attention si concentri maggiormente nell'area che circonda i due farmaci è lampante nella parte destra della Figura 5.7. Pur mantenendo la stessa frase, in questo altro esempio abbiamo considerato non più la coppia tra *azole antifulgas* e *fluconazole* bensì tra quella tra quest'ultimo (che quindi viene sostituito da (*PairDrug1*) e *warfarin* (*PairDrug2*). Come si può facilmente notare, la parte relativa ad *azole antifulgas* non viene più evidenziata, mentre l'attention prolunga il suo effetto anche su *concurrently with warfarin*. Vista la vicinanza con la parte maggiormente interessante, si alzano anche i pesi dei token *have been*, praticamente gli unici casi di un peso intermedio nei due esempi.

La tendenza dell'attention di isolare parti rilevanti e di escluderne completamente altre è confermata anche quantitativamente. Innanzitutto, definiamo come importanti i token con peso normalizzato superiore a 0.8. Considerando quindi tutti l'insieme $T$ dei token non importanti e il loro peso $w_j$, possiamo calcolare la media dei pesi:

$$\bar{w} = \sum_{j \in T} w_j \tag{5.5}$$

Il peso medio dei token non importanti è 0.19, un valore quindi molto basso che conferma come la maggioranza di questi ultimi sia molto vicina allo 0 con solamente una minoranza di pesi intermedi, come abbiamo visto nei due esempi. Se poi consideriamo solo le istanze di cui viene considerata importante meno del 50% del testo, il peso medio dei token non evidenziati dall'attention scende a 0.11.

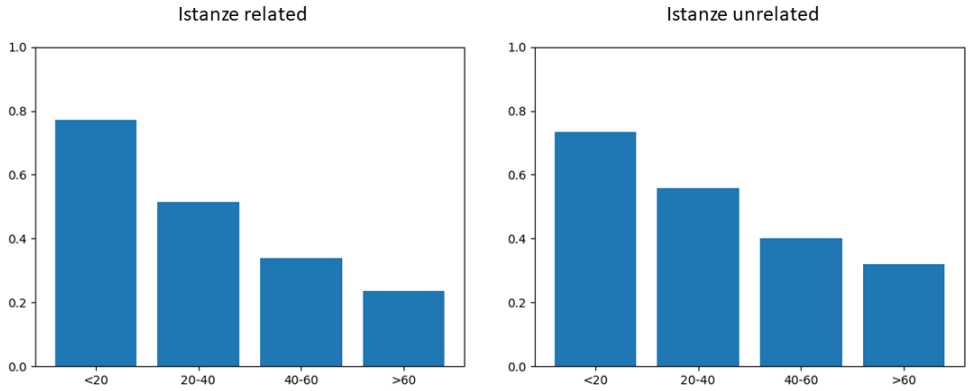


Figura 5.8: Percentuale dei token ritenuti importanti dall'attention per frasi che descrivono una relazione (related, a sinistra) o appartenenti alla classe unrelated (a destra). Sull'asse x, il numero di token della frase.

### 5.5.2 Funzione di gate e filtraggio

La funzione di gate può essere vista come un'azione di filtraggio, che mantiene solamente quella parte del testo candidata, secondo le logiche interne del modello, a descrivere o a escludere la relazione tra farmaci.

Dato che abbiamo due macro-tipologie di istanze nel dataset, abbiamo separato anche la nostra analisi per: le istanze appartenenti alla classe unrelated, ovvero quelle che non descrivono un'interazione tra la coppia di farmaci; e le istanze related, ovvero quelle in cui è presente un'interazione, senza entrare nel dettaglio delle singole classi.

Nella Figura 5.8, possiamo vedere l'azione di filtraggio per i due casi: a sinistra, possiamo vedere come varia la percentuale di token importanti (ovvero con peso normalizzato maggiore a 0.8) in base alla lunghezza della frase (meno di 20 parole, tra 20 e 40, 40 e 60 oppure più di 60). A destra, è possibile valutare lo stesso per le istanze unrelated.

I risultati di questa analisi confermano l'idea che l'attention "si attivi" maggiormente per le frasi più lunghe. Questo fenomeno, che era stato notato anche nel caso dei referti radiologici (vedi la Sezione 4.5.2) può essere spiegato intuitivamente tenendo conto di diversi fattori:

- Ogni farmaco non appartenente alla coppia presa in considerazione nel-

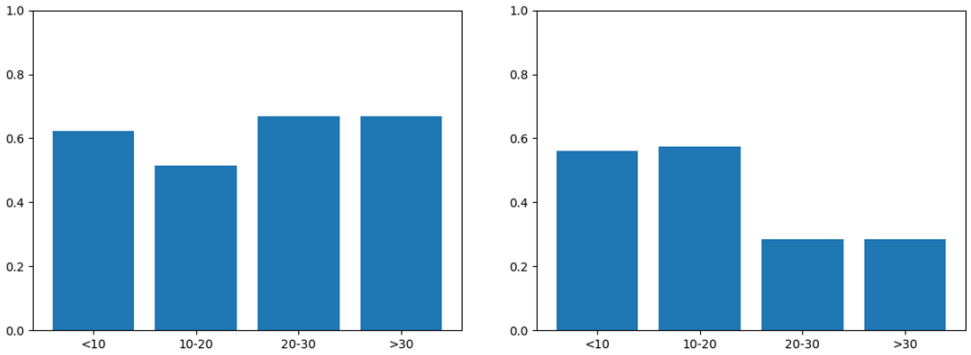


Figura 5.9: Frazione dei token ritenuti importanti dall'attention per frasi che descrivono una relazione (related, a sinistra) o appartenenti alla classe unrelated (a destra). Sull'asse x, il numero di token che intercorrono tra i due farmaci.

l'istanza (sostituito dal termine generico *Drug*) è totalmente irrilevante ai fini del task di relation extraction. Le frasi più lunghe in genere quindi contengono anche un numero maggiore di farmaci che l'attention tende inevitabilmente a scartare (vedi anche la Figura 5.10 per un caso limite).

- La relazione è spesso descritta tramite pochi token. Ad esempio, nella frase in Figura 5.7, la parte relativa a *In post-marketing experience* oppure quali siano stati i "bleeding events" (*bruising, epistaxis, gastrointestinal bleeding, hematuria, and melena*) sono dettagli che possono essere tralasciati dall'algoritmo. Chiaramente però, nelle frasi più corte il numero di dettagli o informazioni di contesto è molto minore, ergo la scarsa azione di filtraggio del testo.

- Se per le frasi più corte il livello LSTM può essere già sufficiente per comprendere il significato del testo, per quelle più lunghe le difficoltà di memoria presenti in questo layer ricorrente necessitano di un'azione maggiore dell'attention per selezionare le parti più importanti [76].

Non si vedono invece grandi differenze nell'azione della funzione di gate tra istanze classificate unrelated e related. La tendenza a filtrare maggior-

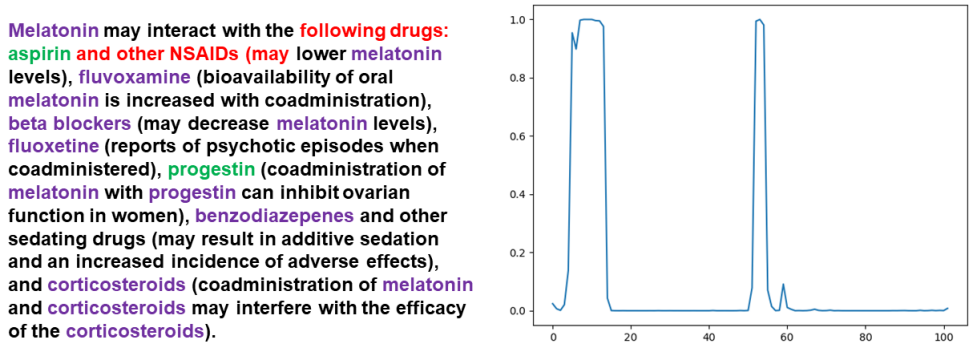


Figura 5.10: Visualizzazione del comportamento dell'attention mechanism per una frase unrelated con ampia distanza tra un farmaco e l'altro. In verde, la coppia di farmaci dell'istanza considerata, in rosso le altre parole selezionate dall'attention, in viola tutti gli altri farmaci menzionati.

mente le frasi più lunghe è infatti visibile in entrambi i casi, con differenze poco significative. Va comunque detto che, tranne per le frasi molto corte, le istanze unrelated tendono a essere filtrate in maniera minore (0.55 per le frasi tra 20 e 40 parole contro lo 0.51 delle related, 0.40 contro 0.34 tra 40 e 60, 0.32 contro 0.23 per le frasi con più di 60 parole).

Il comportamento dell'attention tra istanze related e unrelated è invece completamente diverso rispetto alla distanza, in termini di token, tra un farmaco e l'altro. Infatti, come si può vedere nella Figura 5.9, per le coppie di farmaci con più di 30 token tra un farmaco e l'altro il comportamento è praticamente opposto. Mentre per le istanze predette related l'attention tende ad evidenziare all'incirca sempre la stessa quantità di testo (attorno al 60%), per quelle unrelated questo vale solo per le frasi con pochi token tra i due farmaci. Infatti, se la distanza tra i due farmaci della coppia aumenta, l'attention tende a filtrare molto di più, considerando importante solamente circa il 20% delle frasi per le frasi con più di 20 token tra un farmaco e l'altro.

Uno dei motivi per cui questa differenza è così marcata è la presenza di istanze come quella in Figura 5.10. Sebbene sia stato scelto volutamente un caso limite, analizziamo per un attimo il significato di questa lunghissima frase. Questa descrive la possibilità che il farmaco *melatonin* interagisca con un lunghissimo elenco di altri farmaci, tra cui *aspirin* e *progestin*. Prendendo

però in considerazione come istanza questi ultimi due farmaci, il testo non descrive nessuna interazione tra di loro, che vengono solamente citati assieme in un elenco. L'attention quindi correttamente individua la zona relativa al primo farmaco, evidenziando anche un certo numero di parole negli immediati dintorni "*following drugs: aspirin and other NSAIDS (may*" ma poi esclude tutto il resto, salvo giusto *progestin* che è il secondo membro della coppia presa in considerazione. Mentre molte di queste istanze, specialmente se i farmaci sono separati solamente da virgole e congiunzioni, vengono classificate ancora prima di entrare nella rete neurale dal filtro delle negative, le più complesse come quella di questo esempio chiaramente influenzano le statistiche sul comportamento dell'attention.

### 5.5.3 Relazione tra attention e contesti locali

Proviamo ora ad analizzare perché, invece, nelle istanze related l'attention tende a considerare importanti un numero alto di token anche quando la distanza tra i due farmaci è considerevole.

Prima dell'introduzione del deep learning per il Natural Language Processing, il metodo allo stato dell'arte per la Relation Extraction era basato sulle funzioni kernel e sulle Support Vector Machines. In particolare, lavori come [28, 60] costruivano appositi kernel basandosi sull'evidenza empirica che, in genere, una relazione può essere descritta nel testo seguendo tre pattern:

- **Fore-Between**, come nella frase *l'associazione di [P1] e [P2]*, in cui la relazione è descritta nei token prima della prima entità e tra l'una e l'altra.
- **Between**, in cui la relazione è descritta con i token tra le due entità, come ad esempio nella frase *[P1] interagisce con [P2].*
- **Between-After**, in cui la relazione si descrive con i token tra l'una e l'altra e quelli che seguono la seconda, come nella frase *[P1] e [P2] interagiscono.*

Analizzeremo ora come si comporta l'attention in tre settori diversi della frase: prima della prima entità, tra l'una e l'altra, dopo la seconda entità.

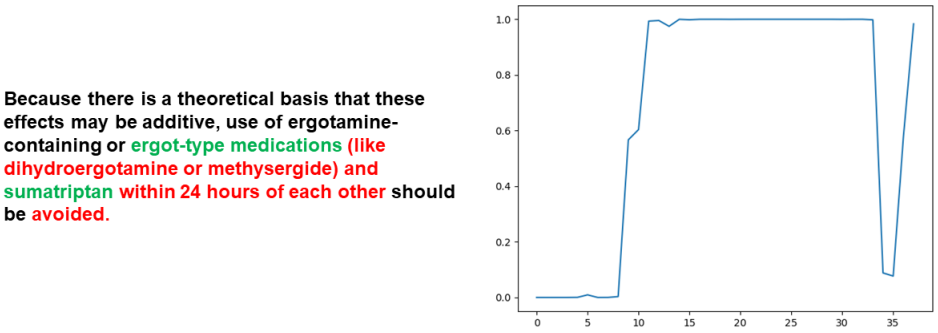


Figura 5.11: Visualizzazione del comportamento dell'attention mechanism per una frase related. In verde, la coppia di farmaci dell'istanza considerata, in rosso le altre parole selezionate dall'attention mechanism.

| | Related | Unrelated |
|---|---|---|
| Before | 0.59 | 0.35 |
| Between | 0.80 | 0.83 |
| After | 0.28 | 0.47 |

Tabella 5.7: Peso medio dell'attention, per istanze related e unrelated nei tre settori: prima del primo farmaco (Before), tra un farmaco e l'altro (Between) e dopo il secondo farmaco (After). Sono state considerate solo istanze in cui viene filtrato almeno il 40% del testo.

Nella Figura 5.11 si può vedere un esempio per una frase related. Dalla frase in figura possiamo ricavare tre settori diversi, chiamati anche contesti locali: Before (*Because there is a theoretical basis that these effects may be addictive, use of ergotamin-containing or*), Between (*ergot-type medications [P1] (like dihydroergotamine or methysergide) and sumatripan [P2]*) e After (*within 24 hours of each other should be avoided*). Possiamo però anche vedere come si comporta l'attention, che di fatto tralascia completamente il contesto Before, a favore invece di quello Between e After, che in effetti ci informano che è da evitare l'assunzione dei due farmaci nell'arco delle stesse 24 ore.

Per valutare il comportamento da un punto di vista quantitativo, abbiamo calcolato il peso medio dell'attention nei tre settori, sia per le istanze related che unrelated in cui viene considerato non importante almeno il 40% del testo (scartando quindi le frasi più corte che di fatto non subiscono l'ef-

fetto dell'attention). Nella Tabella 5.7 sono visibili i risultati. Si può notare immediatamente che il testo compreso tra un farmaco e l'altro ha un peso medio molto alto, sia per le istanze related che non. Gli altri due settori invece hanno un peso decisamente più basso, con le related che si concentrano soprattutto nella parte Before (0.59 contro 0.35 delle unrelated), mentre succede l'esatto contrario nel contesto After (0.47 delle unrelated contro 0.27). La grande importanza assegnata dall'attention al contesto Between conferma quanto osservato da [28] e [60], ovvero che in tutti e tre i pattern è necessario considerare (da sola, oppure con le parole precedenti o seguenti) la parte di testo compresa tra le due entità.

Seguendo anche qui l'intuizione di un metodo originariamente usato per le funzioni kernel, abbiamo anche analizzato i pesi che l'attention assegna al dependency path, che presumibilmente contiene importanti informazioni riguardanti la relazione tra farmaci [9].

Sia per quanto riguarda le istanze related che unrelated, il peso medio che l'attention assegna ai token che compongono il dependency path è in effetti molto alto, rispettivamente 0.86 e 0.88. Considerando le istanze di cui viene filtrato almeno il 40% del testo, il peso medio si abbassa ma rimane molto significativo, con 0.79 in entrambi i tipi di istanza.

Nella sezione precedente abbiamo visto come però l'attention tenda a concentrare la propria azione sul contesto Between. Abbiamo quindi analizzato se il dependency path possa essere visto semplicemente come un sottoinsieme del contesto Between: per entrambi i tipi di istanza, mediamente più dell'80% del dependency path è contenuto in quel settore. Al contrario, il dependency path, sempre mediamente, forma il 65% del contesto Between per le related e il 55% per le unrelated. Da questa breve analisi possiamo quindi dedurre che il dependency path è perlopiù un sottoinsieme di Between, di cui occupa circa il 60%. Non sorprende quindi che venga considerato in maniera così significativa dall'attention.

### 5.5.4 Sintesi sull'interpretazione dell'attention

Se nelle sezioni precedenti abbiamo cercato di valutare su esempi specifici l'azione dell'attention e cercato di capirne il ragionamento sottostante, una valutazione sull'intero dataset richiederebbe l'intervento di esperti nel dominio che identificassero, alla luce della loro conoscenza, le vere parti del testo che contengano le informazioni più importanti ai fini della relation extraction.

I ragionamenti quantitativi però ci permettono di trarre qualche conclusione:

- Anche per testi più corti rispetto a documenti di centinaia di parole, si conferma la tendenza dell'attention, per i task a singola sequenza, a dividere il testo in due categorie distinte: le parole importanti e quelle non importanti. Questa tendenza è accentuata nelle frasi più lunghe, senza particolari differenze se nel testo è descritta o meno una relazione tra i farmaci dell'istanza considerata.

- L'attention è in grado di riconoscere la coppia di farmaci considerata e di scartare gli altri farmaci presenti nella frase e modificare il proprio comportamento di conseguenza (vedi Figura 5.7).

- Come nel caso dei referti radiologici, l'attention spesso lavora per zone o espressioni piuttosto che per singoli token. Il peso medio del contesto Between è alto nonostante siano sicuramente presenti stop-words, articoli, preposizioni ecc. A queste parole (come si può facilmente vedere anche in Figura 5.11 con *and*, *or* e *of*) viene assegnato un peso molto alto più per la posizione che non per l'effettivo significato. Questo è dovuto anche al fatto che l'attention opera sul risultato di un livello LSTM, che lavora non solo sulla parola corrente ma anche su termini precedenti e, nel caso bidirezionale, perfino seguenti.

- L'attention tende a evidenziare in particolar modo la parte di testo compresa tra i due farmaci, il cosiddetto contesto Between. Questo conferma l'evidenza empirica che, in questa zona, si trovino informazioni importanti per la relation extraction. Questo può anche essere

il motivo per cui, per le istanze related con distanza ampia tra i due farmaci (vedi Figura 5.9), l'attention consideri importante oltre il 60% della frase.

Come dimostra l'analisi del dependency path, tuttavia, riteniamo che l'attention non riconosca i singoli token che definiscono la relation extraction. Non è possibile quindi, date queste analisi, implementare un sistema che fornisca direttamente una risposta al perché il modello abbia preso una determinata decisione. Questo è chiaramente un limite per l'interpretabilità. Nonostante ciò, possiamo dire che l'attention è in grado di isolare delle parti di frasi secondo logiche che possono essere individuate e, in linea di massima, spiegate o ricondotte ad evidenze empiriche. Questo è già di per sé un passo verso la comprensione del comportamento del modello.

## 5.6 Modelli basati su BERT

I modelli basati quasi esclusivamente sull'attention mechanism come Transformer e BERT (vedi Sezione 3.4.2) hanno ottenuto ottime prestazioni nei principali task di Natural Language Processing. I primi modelli, analogamente a quanto accaduto con Word2Vec, sono stati addestrati utilizzando documenti del tutto generici [103] quali BookCorpus [110] e la versione inglese di Wikipedia, con l'obiettivo di comprendere il linguaggio nella sua accezione più comune. Questi modelli tuttavia possono non essere sufficienti per task che riguardino un linguaggio strettamente tecnico, come quello scientifico o biomedico. Per questo motivo, modelli come SciBERT [5], BioBERT [46] o BlueBERT [65] sono stati addestrati con lo stesso procedimento delle versioni tradizionali di BERT ma utilizzando articoli scientifici, letteratura biomedica e testi clinici in lingua inglese.

Anche per il corpus DDI-2013 quindi sono stati realizzati alcuni lavori che sfruttano questi modelli pre-allenati per la classificazione delle interazioni tra farmaci, ottenendo un notevole miglioramento rispetto allo stato dell'arte. Tuttavia, mentre in un normale modello di deep learning è possibile sceglierne la configurazione e procedere all'addestramento, nel caso di

| | F(%) |
|---|---|
| BERT-base [103, 21] | 78.0 |
| BlueBERT [65] | 79.9 |
| Character BERT [21] | 80.4 |
| Relation BERT [62] | 80.9 |
| BERTChem-DDI [59] | 83.8 |
| Molecular SciBERT [3] | 84.1 |

Tabella 5.8: Confronto tra i modelli più recenti basati su BERT in termini di Precision (P), Recall (R), negli articoli in cui sono riportate, e in termini di F-Score (F).

BERT si è costretti ad operare in modo diverso. Innanzitutto, la configurazione di BERT (il numero di blocchi, il numero di head di self-attention o la dimensione della rappresentazione delle parole) non è modificabile a meno di non ripetere l'intero addestramento: un'operazione estremamente complessa e costosa e che richiede la disponibilità milioni di documenti. Il modello pre-allenato va quindi considerato quasi come una black box di cui è possibile effettuare solamente qualche aggiustamento (per maggiori dettagli sul funzionamento, si vedano le Sezioni 3.4.2 e 4.6).

Per risolvere al meglio un task di machine learning quindi, volendo sfruttare le potenzialità di BERT, conviene concentrare i propri sforzi sulla preparazione del dataset, arricchendolo di nuove informazioni o con un adeguato preprocessing, e sul come trattare l'encoding fornito dal modello pre-allenato. Se la soluzione più semplice è quella appena descritta di un unico livello fully-connected, è possibile anche l'impiego di procedimenti più complessi.

### 5.6.1 Approcci già esistenti

Nella Tabella 5.8 sono mostrati i modelli basati su BERT che si trovano in letteratura per il task DDI-2013. Mentre BERT-base è la semplice applicazione del modello BERT-base-cased, ovvero quello del tutto generale allenato da [103] e testato da [21] ottenendo una F-Score di 78.0 (non è presente il dettaglio di precision e recall), tutti gli altri introducono miglioramenti al fine di creare un modello più specifico e accurato per task biomedici o proprio

per la Relation Extraction e la Drug-Drug Interaction.

BlueBERT [65] è un modello di BERT allenato utilizzando documenti biomedici, in particolare gli articoli presenti su PubMed e il dataset MIMIC-III, una delle più importanti risorse disponibili per quanto riguarda i testi clinici [40]. Il guadagno rispetto alla versione di base è di quasi due punti in termini di F-Score (79.9). Di questo modello abbiamo anche la recall (79.3) e la precision (80.5). Le metriche riportate in Tabella 5.8 sono state ottenute grazie al codice rilasciato dagli sviluppatori di BlueBERT[2].

Lo stesso modello è sfruttato da [21] con tuttavia un'informazione ulteriore. Alla rappresentazione delle parole pre-allenata da BlueBERT viene infatti aggiunta un'altra rappresentazione, basata sui caratteri che compongono le parole, calcolata da una rete convoluzionale. Questa architettura viene testata su diversi dataset, compreso il DDI-2013, ottenendo una F-Score complessiva di 80.4. L'articolo non riporta precision e recall.

Relation BERT (o R-BERT) [62] è un modello basato su BERT ma direttamente orientato per la Relation Extraction. Questo lavoro, dopo una semplice fase di pre-processing, non utilizza la rappresentazione della frase come input per l'ultimo livello fully-connected, bensì quella dell'ultimo token che segna la fine della frase (comunemente indicato come *CLS*) e dei token che compongono le due entità coinvolte nella relazione. Siccome un'entità può essere composta da un numero variabile di token, viene calcolata una media tra i diversi vettori. Al vettore di *CLS* e a quelli rappresentanti le due entità viene poi applicata la funzione tanh, per essere successivamente passati a un livello fully connected con attivazione lineare. Infine, i tre vettori risultanti vengono concatenati e compongono l'input per l'ultimo livello con attivazione softmax che fornisce la classificazione. Questo accorgimento porta miglioramenti sia utilizzando il modello base di BERT (79.1 contro 78.0), sia con un modello pre-allenato su documenti biomedici (80.9 contro 79.9). Il modello biomedico di R-BERT tuttavia non è BlueBERT ma BioBERT [46] per cui non è possibile fare un confronto diretto tra la versione base e quella orientata alla Relation Extraction. Anche per R-BERT non sono riportate precision e recall.

[2]https://github.com/ncbi-nlp/bluebert

I modelli migliori tuttavia operano in un'altra direzione. Mentre le modifiche all'architettura riescono ad ottenere miglioramenti intorno al punto percentuale, BERTChem-DDI [59] e Molecular SciBERT [3] ottengono risultati ancora superiori, con rispettivamente 83.8 e 84.1 di F-Score non tanto operando sull'architettura ma integrando nuove informazioni. Infatti, in entrambi i casi, parallelamente al modello di BERT per l'analisi del testo, un altro modello di deep learning analizza la struttura molecolare dei farmaci citati nel testo, permettendo quindi al sistema complessivo non solo di capire il significato della frase ma anche di trovare regolarità anche su base biologica. Nel primo caso, viene utilizzato un autoencoder non supervisionato [42]. Nel secondo invece un'architettura ancora più complessa basata su Graph Neural Networks che, visto che la struttura molecolare è rappresentabile come un grafo, è particolarmente indicata per questo tipo di analisi [100]. Oltre alla struttura moleculare, il modello proposto in [3] prende in considerazione anche la descrizione, sotto forma di testo estratto dal database DrugBank, dei due farmaci. Mentre per [59] non è esattamente chiaro il modello di BERT utilizzato nè sono disponibili precision e recall, la trattazione di [3] è più completa e indica chiaramente l'uso di SciBERT [5] (modello di BERT allenato su generici testi di tipo scientifico, senza particolare concentrazione sull'ambito biologico) e riporta una precision pari a 85.4 e una recall a 82.8.

### 5.6.2 Alcune note sui risultati

Nel nostro tentativo di riprodurre i risultati forniti da BlueBERT abbiamo potuto notare immediatamente un problema: la variabilità dei risultati forniti dal modello di BERT tra un'esecuzione e l'altra.

Nella Tabella 5.9 mostriamo i risultati di sette diverse esecuzioni di BlueBERT, con gli stessi parametri ( numero di epoche, learning rate, training set, batch size, ecc.). Mentre il risultato medio (79.9 di F-Score) rispetta quanto riportato in [65], possiamo notare come, tra un'esecuzione e l'altra ci sia notevole variabilità sia per tutte e le tre metriche: da un massimo di 82.3 di precision si può passare a un minimo di 78.6, la recall varia da 76.7 a 81.1 e la F-Score tra 77.7 e 81.6. La deviazione standard delle tre metriche

| | P(%) | R (%) | F(%) |
|---|---|---|---|
| Es. 1 | 82.3 | 80.1 | 81.1 |
| Es. 2 | 82.0 | 81.1 | 81.6 |
| Es. 3 | 78.6 | 76.7 | 77.7 |
| Es. 4 | 81.8 | 80.0 | 80.9 |
| Es. 5 | 81.2 | 79.2 | 80.2 |
| Es. 6 | 78.2 | 79.2 | 78.6 |
| Es. 7 | 80.5 | 78.7 | 79.6 |
| **Media** | **80.7** | **79.3** | **79.9** |
| **Dev. St.** | **1.65** | **1.38** | **1.41** |

Tabella 5.9: Risultati di 7 esecuzioni di BlueBERT in termini di Precision (P), Recall (R) e F-Score). Nelle ultime due righe, in grassetto, la media delle metriche e la loro deviazione standard.

è attorno a 1.5, con valori più alti per la precision (1.65) che non per recall e F-Score (1.38 e 1.41 rispettivamente).

In generale, tutti i sistemi di deep learning risentono non solo del training set e degli iperparametri scelti per l'addestramento, ma anche dell'inizializzazione casuale dei pesi, del rumore che è necessario introdurre per aiutare la generalizzazione (come nel caso del dropout) e perfino dell'architettura del calcolatore su cui si svolgono le operazioni [2]. Quando però si ha una variabilità così alta, è difficile confrontare le prestazioni di modelli diversi: alcune istanze della riproduzione di BlueBERT infatti superano di quasi un punto percentuale i risultati forniti da Character BERT e Relation BERT.

Questo problema è stato affrontato dagli sviluppatori di Character BERT, che hanno ripetuto i loro esperimenti 10 volte e riportano i valori di primo quartile, mediana e terzo quartile, oltre che la media e la deviazione standard in forma grafica. Il valore di 80.4 di F-Score riportato anche in Tabella 5.8 è infatti quello della mediana di una distribuzione che ha come primo quartile 79.2 e 81.7 come terzo. Quella della loro riproduzione della versione base di BERT è invece compresa tra 77.7 e 78.8, con 78.0 come mediana. Può anche essere interessante notare come il loro tentativo di riprodurre BlueBERT ottenga risultati decisamente peggiori dei nostri: 77.9 di mediana, addirittura meno della versione base di BERT e circa due punti sotto quanto ottenuto

| | Effect | Mechanism | Advise | Int |
|---|---|---|---|---|
| BlueBERT [65] | 79.4 | 83.0 | 87.3 | 50.4 |
| Relation BERT [62] | 77.8 | 97.4 | 87.7 | 52.5 |
| Molecular SciBERT [3] | 85.2 | 84.0 | 82.7 | 79.0 |

Tabella 5.10: Prestazioni dettagliate dei modelli basati su BERT, in termini di F-Score per ciascuna delle classi related.

da noi.

Mentre la loro analisi si focalizza anche su questi aspetti, gli altri approcci riportati in Tabella 5.8 non ne fanno cenno. Va comunque fatto notare che, rispetto alla situazione descritta nella Sezione 5.4.4, tutti i modelli ad eccezione di BERTChem-DDI forniscono il codice per la riproduzione dei risultati.

Un'altra importante differenza degli approcci basati su BERT rispetto a quanto trattato nelle sezioni precedenti (con modelli basati su algoritmi di machine learning o di deep learning da addestrare da zero) è la possibilità di fare molto rapidamente *transfer learning*, ovvero sfruttare lo stesso modello per diversi task, per diversi dataset ecc. Per questo motivo, diversi articoli (come nel caso di BlueBERT e Character BERT) presentano un modello del tutto generale con ottimizzazioni nell'architettura o nell'addestramento e lo applicano a più dataset, mostrandone pregi e difetti. Per questo motivo, spesso non vengono riportati i risultati dettagliati di ogni classe del dataset DDI-2013 o degli altri dataset considerati, anche solamente per questioni di spazio, preferendo fornire una panoramica più generale.

Nella Tabella 5.10 vengono mostrati i risultati dettagliati per i due articoli che li riportano (non a caso, completamente incentrati sul problema della DDI-2013) e quelli medi di BlueBERT ottenuti nei nostri esperimenti per la riproduzione dei risultati. Come si può notare confrontando questi risultati con quelli dei nostro modelli basati su LSTM e vari tipi di Attention mechanism (Tabelle 5.3 e 5.5), il guadagno in termini di F-Score è evidente. Tuttavia è anche possibile notare notevoli differenze tra i singoli modelli basati su BERT: ad esempio, Relation BERT ottiene un impressionante 97.4 di F-Score per la classe mechanism, superando di più di 10 punti sia BlueBERT

che Molecular SciBERT. Quest'ultimo invece ottiene risultati molto migliori per la classe effect. Discorso a parte merita la classe int: infatti, come descritto nella Sezione 5.4.2, questa classe fortemente minoritaria è quella che causa maggiori problemi al modello di classificazione. La mancanza di esempi fa calare bruscamente le prestazioni e, se per i nostri modelli LSTM eravamo attorno ai 49 punti di F-Score, il guadagno ottenuto da BlueBERT e Relation BERT è piuttosto scarso: poco più di un punto per il primo (50.4) e altri due punti per il secondo (52.5). Molecular SciBERT invece arriva a un sorprendente 79.0 di F-Score, allineando di fatto le prestazioni di questa classe a quelle delle altre, ben più popolate. Sebbene le cause di questo miglioramento non siano analizzate nell'articolo, riteniamo che l'arricchimento delle frasi con le descrizioni dei farmaci e con la loro struttura molecolare siano riuscite a chiarire molte di queste frasi che non solo sono in forte minoranza, ma spesso sono anche piuttosto corte, generiche e non dettagliate.

### 5.6.3 Direzioni ed esperimenti per il miglioramento

I progressi ottenuti per il riconoscimento di interazioni tra farmaci tramite modelli BERT sono molto recenti. Ad esempio, Charachter BERT [21], Relation BERT [62] e Molecular SciBERT [3] sono stati pubblicati, a pochi giorni di distanza l'uno dall'altro, nell'ottobre 2020. Solamente per questioni temporali quindi i nostri sforzi di miglioramento si sono concentrati su BlueBERT [65], disponibile già dalla fine del 2019.

Abbiamo già mostrato nella Tabella 5.9 le nostre prestazioni nel replicare i risultati e la loro variabilità, mentre nella prima riga della Tabella 5.10 sono disponibili le F-Score per ogni classe related. Anche solo a colpo d'occhio, si intuisce che le maggiori possibilità di miglioramento risiedono nella classe int (un'intuizione che poi verrà confermata da Molecular SciBERT). Nonostante infatti i notevoli progressi fatti con i modelli basati su LSTM prima e su BERT poi, già l'approccio basato su Support Vector Machines descritto in [14] otteneva 54.7 di F-Score per quella classe, un risultato addirittura superiore rispetto ai modelli più complessi e molto più accurati pubblicati più recentemente.

#### 5.6.3.1 Data augmentation per la classe int

Invece che lavorare sull'architettura, quindi, si è preferito puntare sulla *data augmentation*, ovvero sulla generazione di nuovi dati manipolando quelli già disponibili. Questa tecnica ha avuto ottimi riscontri per quanto riguarda il trattamento delle immagini [85], in cui rotazioni, variazioni di colore, ridimensionamenti e altre semplici operazioni possono generare facilmente nuovi esempi che aiutino l'algoritmo nell'apprendimento. Per dati testuali, tuttavia, la questione non è così semplice.

In generale, possiamo dividere l'augmentation per dati testuali in due macro-categorie: le operazioni puramente lessicali e quelle basate su word embedding [27]. Nelle prime, si modifica direttamente il testo, per esempio sostituendo una certa percentuale di parole con dei sinonimi, oppure permutandone l'ordine, tagliando parti del documento ecc. Nelle seconde invece si cerca o di modificare numericamente i vettori di word embedding introducendo del rumore, oppure si va a sostituire parole con quelle più prossime nello spazio vettoriale. Dato che un modello di BERT lavora direttamente sul testo, costruendosi da sé le rappresentazioni word embedding ad ogni livello dell'encoder, per una data augmentation è possibile solo operare lessicalmente.

Abbiamo focalizzato la nostra attenzione sulla classe int, con un'augmentation molto mirata, portando la numerosità nel training set per quella classe pari a quella delle più popolose effect, mechanism e advise. Sono stati quindi introdotte 300 frasi *artificiali*, andando di fatto a triplicare il numero di esempi. Queste frasi sono state generate applicando le seguenti trasformazioni:

- **Sostituzione di parole con sinonimi attraverso WordNet**. Questa procedura, di per sé molto semplice, di selezione di una parola e la sua sostituzione con una simile ricavata attraverso il database lessicale WordNet[3], può essere molto delicata. Infatti, per quanto riguarda termini altamente tecnici quali il linguaggio biomedico è sia difficile trovare dei sinonimi che assicurarsi che la differenza non alteri sensi-

[3]https://wordnet.princeton.edu

bilmente il significato della frase, potenzialmente introducendo errori nel dataset. Per questo motivo, ci siamo concentrati più che altro sulla sostituzione dei verbi presenti nella frase che, esprimendo una relazione senza specificarne i dettagli, sono spesso termini del linguaggio comune. Anche termini meno importanti come preposizioni, avverbi ecc. possono venire sostituiti.

- **Manipolazione dei settori non interessanti**. Molto spesso, nel corpus DDI-2013, le frasi sono divise in settori, divise da punti e virgola o due punti. Altrettanto spesso, entrambi i farmaci che compongono la relazione si trovano solamente in uno di questi. Ritenendo che anche l'interazione sia espressa nello stesso settore, sul resto della frase è possibile operare più drasticamente: tagliandoli direttamente, oppure sostituendo con dei sinonimi in maniera più grossolana. Inoltre, queste parti non interessanti possono essere aggiunte ad altre frasi, che manterranno la relazione tra i due farmaci citati ma con in aggiunta del rumore.

Le due procedure sono intese per svolgere due funzioni parallele: mentre la prima serve per migliorare la generalizzazione dell'algoritmo sui modi ed i termini per esprimere un'interazione tra farmaci, la seconda lo rende più stabile rispetto a tutto il resto della frase, che può contenere termini medici, altri nomi di farmaci, unità di misura ecc. che possono fuorviare il modello di deep learning, aiutandolo a focalizzarsi solamente sul settore che esprime l'interazione.

Nella Tabella 5.11 mostriamo il confronto delle sette esecuzioni di BlueBERT già mostrate nella sezione precedente con altre sette esecuzioni dello stesso modello ma allenato con il dataset su cui è stata svolta l'augmentation della classe int. Sebbene si ottenga un guadagno minimo in termini di F-Score (80.1 contro 79.9), si può notare una differenza marcata nel bilanciamento tra Precision e Recall dei due modelli. Infatti, ad eccezione dell'Esecuzione 6, tutte le esecuzioni del modello base hanno una Precision più alta della Recall, a volte anche di quasi due punti o più. Nel modello allenato con il dataset aumentato, invece, accade esattamente il contrario: la Recall è sempre più

| | Base | | | Aumentato | | |
|---|---|---|---|---|---|---|
| | P(%) | R (%) | F(%) | P(%) | R (%) | F(%) |
| Es. 1 | 82.3 | 80.1 | 81.1 | 78.5 | 79.7 | 79.1 |
| Es. 2 | 82.0 | 81.1 | 81.6 | 77.2 | 81.7 | 79.4 |
| Es. 3 | 78.6 | 76.7 | 77.7 | 79.9 | 82.5 | 81.2 |
| Es. 4 | 81.8 | 80.0 | 80.9 | 79.7 | 80.8 | 80.3 |
| Es. 5 | 81.2 | 79.2 | 80.2 | 78.7 | 79.0 | 78.8 |
| Es. 6 | 78.2 | 79.2 | 78.6 | 78.8 | 81.5 | 80.2 |
| Es. 7 | 80.5 | 78.7 | 79.6 | 81.6 | 82.0 | 81.8 |
| **Media** | **80.7** | **79.3** | **79.9** | **79.2** | **81.0** | **80.1** |
| **Dev. St.** | **1.65** | **1.38** | **1.41** | **1.37** | **1.30** | **1.10** |

Tabella 5.11: Confronto dei risultati di 7 esecuzioni del modello base di BlueBERT e di quello con l'augmentation della classe int. Le metriche considerate sono Precision (P), Recall (R) e F-Score. Nelle ultime due righe, in grassetto, la media delle metriche e la loro deviazione standard.

alta, a volte di anche molto (più di 4 punti nell'Esecuzione 2). Andando a confrontare i risultati medi, di fatto il bilanciamento tra Precision e Recall si inverte: da 80.7 e 79.3 nel modello base, si passa a 79.2 e 81.0 in quello aumentato. Non ci sono invece grandi differenze nella deviazione standard che, tuttavia, risulta leggermente minore per il modello aumentato.

Andando ad analizzare i risultati ancora più nel dettaglio abbiamo notato che questa differenza di comportamento dei due modelli è dovuta proprio alla classe int. Nella Tabella 5.12 mostriamo Precision, Recall e F-Score della classe int sia per il modello base che per il modello allenato con il dataset aumentato. Come si può facilmente vedere, il modello base ha una Precision media molto alta (66.9) e anche molto variabile (arrivando a un massimo di 88.6 nell'Esecuzione 4, e con una deviazione standard di 12.4) a fronte di una Recall quasi sempre molto bassa (media a 41.3 che varia tra 38.9 e 44.0). Questo significa che il modello non riesce a identificare la maggior parte delle istanze della classe int (solo 4 su 10). Tuttavia, quando ne predice una frase come appartenente a quella classe ha buone prestazioni, individuandone circa 2 su 3.

Il modello aumentato ha un comportamento del tutto diverso. Ad un netto calo della Precision (55.7, quindi 11 punti in meno) corrisponde un

| | Base | | | Aumentato | | |
|---|---|---|---|---|---|---|
| | P(%) | R (%) | F(%) | P(%) | R (%) | F(%) |
| Es. 1 | 77.0 | 42.9 | 55.0 | 55.5 | 56.4 | 55.0 |
| Es. 2 | 65.8 | 40.2 | 48.7 | 56.0 | 46.9 | 50.8 |
| Es. 3 | 54.7 | 41.2 | 47.2 | 58.3 | 55.1 | 56.4 |
| Es. 4 | 88.6 | 40.8 | 56.1 | 53.7 | 55.7 | 51.9 |
| Es. 5 | 57.0 | 38.9 | 46.0 | 49.2 | 46.1 | 48.1 |
| Es. 6 | 66.7 | 41.3 | 51.4 | 47.4 | 47.3 | 47.9 |
| Es. 7 | 57.2 | 44.0 | 49.1 | 68.6 | 49.0 | 57.8 |
| **Media** | **66.9** | **41.3** | **50.4** | **55.7** | **50.4** | **52.6** |
| **Dev. St.** | **12.4** | **1.70** | **3.82** | **7.11** | **3.95** | **3.95** |

Tabella 5.12: Confronto dei risultati di 7 esecuzioni del modello base di BlueBERT e di quello con l'augmentation della classe int. Le metriche considerate sono Precision (P), Recall (R) e F-Score. Nelle ultime due righe, in grassetto, la media delle metriche e la loro deviazione standard.

aumento della Recall (50.4, 9 punti in più). In termini di F-Score media, il nuovo modello guadagna più di 2 punti, passando dai 50.4 del modello base ai 52.6 con l'augmentation. Mentre la variabilità complessiva è pressoché costante (3.82 per la F-Score del modello base, 3.95 per quella del modello aumentato), quella della Precision è quasi dimezzata (da 12.4 a 7.1 di deviazione standard). Questo riequilibrarsi delle due metriche sta a significare che il modello è più in grado di riconoscere circa la metà delle frasi appartenenti alla classe int. Allo stesso tempo però, una sua predizione è meno affidabile rispetto a prima, solo poco più del 55% infatti risultano corrette.

Mentre per le altre classi related non si hanno modifiche del comportamento, la netta differenza tra i due modelli per la classe int si riflette anche nelle prestazioni generali, come mostrato nella Tabella 5.11. Se il lieve miglioramento in termini di F-Score potrebbe anche essere dovuto a variazioni casuali, riteniamo che il comportamento sia imputabile all'effetto dell'introduzione, proprio per la classe int, di 300 ulteriori esempi artificiali.

Dal punto di vista delle prestazioni il miglioramento ottenuto non è particolarmente significativo. Questo può essere dovuto a diversi fattori, quali la portata limitata dell'augmentation. Le frasi artificiali ottenute infatti possono differire anche solo leggermente da esempi già presenti nel dataset,

sostituendo poche parole, e quindi portando un contributo solo marginale. D'altro canto, senza la conoscenza del dominio e del linguaggio biomedico è molto difficile stabilire a priori cosa possa essere alterato di una frase che descrive una relazione tra farmaci preservandone il significato. La modifica del comportamento, tuttavia, ci indica come la strada dell'augmentation possa essere intrapresa con metodi più sofisticati per migliorare anche le prestazioni. L'introduzione indiscriminata di frasi artificiali inoltre, senza ragionare sulla qualità e sulla quantità di quanto aggiunto, potrebbe portare anche a overfitting. Di conseguenza, in future applicazioni di queste tecniche, bisognerà tenere in considerazione anche questi aspetti potenzialmente problematici.

#### 5.6.3.2 Bilanciamento delle classi e altre tecniche

Oltre all'augmentation specifica sulla classe int, è stato svolto anche un esperimento per migliorare le prestazioni generali del modello, in particolare riequilibrando il bilanciamento tra le istanze negative, che ricordiamo sono la stragrande maggioranza all'interno del dataset, e quelle delle classi related.

Il primo approccio segue quello del pre-processing definito nella Sezione 5.2.1, ovvero la rimozione di istanze della classe unrelated tramite lo Shortest Dependency Path. Contrariamente però sia a quanto visto per i nostri modelli LSTM che per gli altri approcci trovati in letteratura (sia basati su SVM [14] che su reti ricorrenti [43, 102]), l'eliminazione di istanze negative non ha portato benefici al modello basato su BlueBERT, fornendo prestazioni estremamente simili.

Il secondo approccio ricalca quello dell'augmentation della classe int, solo che generalizzato per tutte le classi related e aumentando il grado di libertà nelle sostituzioni tramite sinonimi. Questo dataset notevolmente aumentato (circa $10,000$ esempi in più) è stato utilizzato sia come semplice training set per il fine tuning come descritto nella Sezione 5.6, sia con un procedimento a due fasi. In quest'ultimo, successivamente al fine tuning con il dataset aumentato viene svolto un ulteriore addestramento, con il learning rate diminuito, utilizzando il dataset originale. Sebbene l'idea sia che la prima fase serva per rendere più familiare il task in questione, anche con un dataset più

rumoroso, al modello di BERT e che la seconda serva per perfezionarlo ulteriormente, anche in questo caso non si sono trovati miglioramenti rispetto alla baseline posta da BlueBERT. Anche il semplice fine tuning ha ottenuto risultati molto simili.

Seguendo sempre il pre-processing classico di questo task di NLP (vedi Sezione 5.2), è stato provato anche il perfezionamento del procedimento di sostituzione. Infatti, BlueBERT e gli altri modelli simili sostituiscono i token corrispondenti ai nomi dei farmaci con la parola chiave *DRUG* ma lasciano inalterati gli altri nomi presenti nella frase. Negli approcci precedenti invece venivano sostituiti anche questi con espressioni generiche, in modo da rendere il modello completamente indipendente dai nomi specifici e farlo concentrare maggiormente sul contesto della frase. Abbiamo quindi sostituito i nomi dei farmaci non coinvolti nella relazione con la parola chiave *USELESS_DRUG*, ma questo ha causato un netto calo delle prestazioni, portandole al di sotto dei 70 punti in termini di F-Score. La sostituzione degli stessi nomi utilizzando il token speciale di BERT *UNK* ha diminuito il calo. Tuttavia, il modello ha lo stesso un peggioramento di circa 2 punti rispetto alla baseline di BlueBERT.

Sebbene questi tentativi di adattamento della fase di pre-processing al modello di BERT siano solo all'inizio e non è possibile escludere che si riesca a migliorare le prestazioni anche lavorando sul dataset posto in ingresso, possiamo formulare qualche ipotesi del perchè questi approcci non abbiano funzionato. Per quanto riguarda lo sbilanciamento delle classi nel task DDI-2013, è evidente dalla letteratura che BERT ne soffra meno rispetto ai modelli LSTM. Infatti, nessun articolo riporta pre-processing o tecniche particolari per risolvere questo problema (al contrario di quanto accadeva per i modelli precedenti), ottenendo però un netto miglioramento delle prestazioni. Questo può essere chiaramente dovuto al fatto che il modello pre-allenato di BERT ha una comprensione superiore del linguaggio e riesce a distinguere meglio quando una relazione è espressa e quando no. Diminuire quindi il dataset escludendo una certa percentuale di frasi negative quindi potrebbe non fare altro che rendere più difficoltoso il fine tuning.

Per quanto riguarda invece la sostituzione dei nomi dei farmaci non coin-

volti nella relazione, allo stesso modo in letteratura non sono presenti accorgimenti specifici e quindi sembra che BERT abbia già una buona capacità di escludere tali nomi dall'analisi. Tuttavia, mentre l'augmentation generale o la rimozione delle frasi negative non ha modificato il comportamento del modello, in entrambi i casi si è visto un calo delle prestazioni, segno che comunque quell'informazione è stata trattata. Riteniamo quindi probabile che si possano escludere, in altri modi, i farmaci non appartenenti alla relazione e che si possa forzare il modello ancora di più su quelli in esame, con un possibile miglioramento delle prestazioni.

# Capitolo 6

# Conclusioni e sviluppi futuri

In questa tesi sono state descritte le principali attività di ricerca svolte durante il mio corso di dottorato ed i conseguenti risultati sperimentali.

Nel contesto generale dell'applicazione delle tecniche di deep learning (come le Reti Neurali Ricorrenti) per il Natural Language Processing nell'ambito biomedico, il vero punto focale della tesi riguarda l'interpretabilità e il ruolo dell'attention mechanism. La sensibilità dell'implementazione di sistemi di intelligenza artificiale nelle strutture sanitarie, con il supporto a decisioni che interessano direttamente la salute delle persone, non può infatti prescindere da una comprensione del ragionamento interno del modello e delle ragioni per cui è stata fornita una risposta piuttosto che un'altra.

L'analisi del ruolo dell'attention mechanism non è stata quindi svolta solo in termini di prestazioni. Sebbene infatti sia stato verificato sperimentalmente che questo fornisce un contributo per il miglioramento dei risultati predittivi (sia nel caso delle interazioni tra farmaci che per quello della classificazione di referti radiologici), la sua peculiare caratteristica di valutare l'importanza delle parole per il task di machine learning lo rende particolarmente interessante per capire il funzionamento del modello.

I nostri esperimenti nell'ambito del sistema di deep learning per i referti radiologici lo confermano. Anche in questo caso, in cui non vi è una grande disponibilità di documenti (circa 5000 referti), il sistema realizzato ottiene buoni risultati sia in termini prettamente predittivi che di interpretabilità,

proprio grazie all'attention mechanism. Abbiamo infatti verificato che c'è un forte legame tra le annotazioni manuali, ovvero i concetti e i rilievi radiologici maggiormente rilevanti secondo i radiologi, e la parte di testo segnalata come importante dall'attention. Questo conferma come quest'ultimo si concentri sulle frasi e le espressioni più rilevanti anche da un punto di vista medico, giustificando quindi le buone performance del nostro modello. Nonostante questo risultato decisamente positivo, altre caratteristiche del comportamento dell'attention sono meno desiderabili: in molti casi infatti, questo meccanismo, tende ad evidenziare larghe parti del testo. Specialmente per i referti più corti, molto spesso viene ritenuta importante più della metà del contenuto del referto. Uno sviluppo futuro del nostro progetto è senz'altro quindi la riduzione di questa quantità di testo ad una semplice risposta formata dai soli contenuti essenziali. Questa potrebbe essere mostrata ai radiologi come spiegazione della classificazione prodotta dal sistema, aumentandone l'utilità pratica e la sua applicazione nelle strutture sanitarie.

Un lavoro diverso è stato svolto nel contesto delle interazioni tra farmaci, dove si sono confrontate le performance delle diverse varianti dell'attention mechanism e ci si è concentrati sul legame tra il testo ritenuto importante e concetti classici della Relation Extraction come i contesti locali [28] o il dependency path. Se alcune indicazioni sono evidenti, come la concentrazione sulla parte di testo compresa tra i due farmaci considerati per verificarne l'interazione, per proseguire l'analisi e valutare se effettivamente il testo ritenuto importante dall'attention mechanism è quello che giustifica la classificazione presa sarebbe necessario l'intervento di un esperto del dominio.

L'attività più recente del mio dottorato s'è concentrata sui modelli basati su Transformer [92]. Se da un lato le performance sono ancora migliorate, la trattazione dell'interpretabilità di tali modelli è solo agli inizi. Ci sono diverse evidenze che queste nuove architetture estremamente complesse riescano a catturare concetti linguistici [87], ma una spiegazione del funzionamento del modello non è ancora possibile. I recenti tool di visualizzazione degli attention di questi modelli [94] mostrano indubbiamente aspetti interessanti, ma non è certo semplice derivare il testo ritenuto più rilevante del modello da più di 100 *head* di attention diverse e una decina di livelli di encoding. La

mia futura attività di ricerca verterà anche su questo: come estrapolare una spiegazione semplice che possa giustificare la decisione presa da un modello potenzialmente anche molto complesso, in modo tale da poterlo applicare nei settori maggiormente sensibili, come appunto quello medico.

Un problema aperto sottostante a tutta la tesi, e che è ovviamente strettamente legato al machine learning e al deep learning in generale, è quella della disponibilità dei dati e della conoscenza del dominio. Come sottolineato in diversi punti del testo, maggiore è la complessità dell'architettura e il numero dei suoi parametri, maggiore è il costo computazionale (sia in termini di risorse necessarie che di tempo impiegato) e soprattutto maggiore è la quantità di dati necessari per l'addestramento. In ambito biomedico e per lingue diverse dall'inglese, la disponibilità di dati è un problema non banale. Tuttavia, come mostrato nel caso dei referti radiologici, l'introduzione di conoscenza del dominio, con regole logiche per coniugare le predizioni dei diversi classificatori, permette di ottenere buoni risultati anche mediante l'impiego di una minore quantità di dati. L'introduzione di conoscenza, attraverso regole, informazioni aggiuntive o addirittura di vincoli imposti durante l'addestramento [29], può essere un'ulteriore chiave di volta per il Natural Language Processing in ambito biomedico e la sua interpretabilità.

# Bibliografia


[1] J. Alammar. The illustrated transformer, Jun 2018. URL `http://jalammar.github.io/illustrated-transformer/`.

[2] M. Alberti, V. Pondenkandath, L. Vögtlin, M. Würsch, R. Ingold, and M. Liwicki. Improving reproducible deep learning workflows with DeepDIVA. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 13–18, 2019. doi: 10.1109/SDS.2019.00-14.

[3] M. Asada, M. Miwa, and Y. Sasaki. Using drug descriptions and molecular structures for drug-drug interaction extraction from literature. *Bioinformatics*, 2020.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`.

[5] I. Beltagy, K. Lo, and A. Cohan. SciBERT: Pretrained language model for scientific text. In *EMNLP*, 2019.

[6] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*, volume 1. MIT Press Massachusetts, USA:, 2017.

[7] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, Feb. 2012. ISSN 1532-4435.

[8] J. Björne, S. Kaewphan, and T. Salakoski. UTurku: Drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/S13-2108`.

[9] R. Bunescu and R. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, 2005.

[10] S. Carton, Q. Mei, and P. Resnick. Feature-based explanations don't help people detect misclassifications of online toxicity. In M. D. Choudhury, R. Chunara, A. Culotta, and B. F. Welles, editors, *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, pages 95–106. AAAI Press, 2020.

[11] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1636. URL `https://www.aclweb.org/anthology/P19-1636`.

[12] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 882–891. PMLR, 2018. URL `http://proceedings.mlr.press/v80/chen18j.html`.

[13] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014. doi: 10.3115/v1/d14-1179. URL `https://doi.org/10.3115/v1/d14-1179`.

[14] M. F. M. Chowdhury and A. Lavelli. FBK-irst : A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 351–355, 2013. URL `http://aclweb.org/anthology/S/S13/S13-2057.pdf`.

[15] K. W. Church and J. Hestness. A survey of 25 years of evaluation. *Natural Language Engineering*, 25(6):753–767, 2019. doi: 10.1017/S1351324919000275.

[16] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019. URL `http://arxiv.org/abs/1906.04341`.

[17] K. B. Cohen, J. Xia, P. Zweigenbaum, T. J. Callahan, O. Hargraves, F. Goss, N. Ide, A. Névéol, C. Grouin, and L. E. Hunter. Three dimensions of reproducibility in natural language processing. In *LREC... International Conference on Language Resources & Evaluation:[proceedings]. International Conference on Language Resources and Evaluation*, volume 2018, page 156. NIH Public Access, 2018.

[18] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun. Very deep convolutional networks for text classification. In M. Lapata, P. Blunsom, and A. Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*

*2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1107–1116. Association for Computational Linguistics, 2017. doi: 10.18653/v1/e17-1104. URL https://doi.org/10.18653/v1/e17-1104.

[19] I. N. Dewi, S. Dong, and J. Hu. Drug-drug interaction relation extraction with deep convolutional neural networks. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1795–1802, 2017.

[20] F. K. Došilović, M. Brčić, and N. Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[21] H. El Boukkouri, O. Ferret, T. Lavergne, H. Noji, P. Zweigenbaum, and J. Tsujii. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, 2020.

[22] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[23] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. doi: 10.1080/01621459.1937.10503522.

[24] A. E. Gerevini, A. Lavelli, A. Maffi, R. Maroldi, A. Minard, I. Serina, and G. Squassina. Automatic classification of radiological reports for clinical care. In *Proceedings of the 16th Conference on Artificial Intelligence in Medicine, AIME 2017, Vienna, Austria, June 21-24, 2017*, volume 10259 of *Lecture Notes in Computer Science*, pages 149–159. Springer, 2017.

[25] A. E. Gerevini, A. Lavelli, A. Maffi, R. Maroldi, A.-L. Minard, I. Serina, and G. Squassina. Automatic classification of radiological reports for clinical care. *Artificial Intelligence in Medicine*, 91:72 – 81, 2018. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2018.05.006. URL `http://www.sciencedirect.com/science/article/pii/S0933365717305912`.

[26] F. A. Gers, J. Schmidhuber, and F. A. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471, 2000.

[27] P. K. B. Giridhara, C. Mishra, R. K. M. Venkataramana, S. S. Bukhari, and A. Dengel. A study of various text augmentation techniques for relation classification in free text. *ICPRAM*, 3:5, 2019.

[28] C. Giuliano, A. Lavelli, and L. Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[29] M. Gori. *Machine Learning: A constraint-based approach*. Morgan Kaufmann, 2017.

[30] A. Greiner-Petter, A. Youssef, T. Ruas, B. R. Miller, M. Schubotz, A. Aizawa, and B. Gipp. Math-word embedding in math search and semantic extraction. *Scientometrics*, June 2020. ISSN 0138-9130, 1588-2861. doi: 10.1007/s11192-020-03502-9. URL `http://link.springer.com/10.1007/s11192-020-03502-9`.

[31] C. Guan, X. Wang, Q. Zhang, R. Chen, D. He, and X. Xie. Towards a deep and unified understanding of deep neural models in nlp. In *International Conference on Machine Learning*, pages 2454–2463, 2019.

[32] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892, 2012.

[33] Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[34] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[35] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, and T. Declerck. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920, 2013.

[36] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[37] A. Jacovi and Y. Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[38] S. Jain and B. C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

[39] X. Jiang, M. Ringwald, J. A. Blake, C. Arighi, G. Zhang, and H. Shatkay. An effective biomedical document classification scheme in support of biocuration: addressing class imbalance. *Database*, 2019, 04 2019. ISSN 1758-0463. doi: 10.1093/database/baz045. URL `https://doi.org/10.1093/database/baz045`. baz045.

[40] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[41] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.

[42] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

[43] S. Kumar and A. Anand. Drug-drug interaction extraction from biomedical text using long short term memory network. *CoRR*, abs/1701.08303, 2017.

[44] J. D. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[45] R. Leaman, R. Khare, and Z. Lu. Challenges in clinical natural language processing for automated disorder normalization. *Journal of biomedical informatics*, 57:28–37, 2015.

[46] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL `https://doi.org/10.1093/bioinformatics/btz682`.

[47] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL `http://arxiv.org/abs/1612.08220`.

[48] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong Kong, China, Nov. 2019. Association for Computational Lingui-

stics. doi: 10.18653/v1/D19-1488. URL https://www.aclweb.org/anthology/D19-1488.

[49] S. Liu, B. Tang, Q. Chen, and X. Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016.

[50] Z. Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011, 2011.

[51] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[52] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[53] D. Marcheggiani and I. Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1159. URL https://www.aclweb.org/anthology/D17-1159.

[54] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489, 2013.

[55] R. McDonald, G. Brokos, and I. Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. *CoRR*, 2018.

[56] A. Miaschi, D. Brunato, F. Dell'Orletta, and G. Venturi. Linguistic profiling of a neural language model. In D. Scott, N. Bel, and C. Zong,

editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 745–756. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.65. URL https://doi.org/10.18653/v1/2020.coling-main.65.

[57] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

[58] T. M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[59] I. Mondal. BERTChem-DDI: Improved drug-drug interaction prediction from text using chemical structure information. In *Proceedings of Knowledgeable NLP: the First Workshop on Integrating Structured Knowledge and Neural Networks for NLP*, pages 27–32, 2020.

[60] R. J. Mooney and R. C. Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2006.

[61] M. Neumann, D. King, I. Beltagy, and W. Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL https://www.aclweb.org/anthology/W19-5034.

[62] D. P. Nguyen and T. B. Ho. Drug-drug interaction extraction from biomedical texts via relation bert. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–7. IEEE, 2020.

[63] J. Parmar, W. Koehler, M. Bringmann, K. S. Volz, and B. Kapicioglu. Biomedical information extraction for disease gene prioritization. *CoRR*, abs/2011.05188, 2020. URL `https://arxiv.org/abs/2011.05188`.

[64] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[65] Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

[66] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[67] E. Pianta, C. Girardi, and R. Zanoli. The TextPro tool suite. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association, 2008. URL `http://www.lrec-conf.org/proceedings/lrec2008/summaries/645.html`.

[68] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile. Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In R. Bernardi, R. Navigli, and G. Semeraro, editors, *Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019*, volume 2481 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019. URL `http://ceur-ws.org/Vol-2481/paper57.pdf`.

[69] D. Pruthi, M. Gupta, B. Dhingra, G. Neubig, and Z. C. Lipton. Learning to deceive with attention-based explanations. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Procee-*

*dings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4782–4793. Association for Computational Linguistics, 2020. URL `https://www.aclweb.org/anthology/2020.acl-main.432/`.

[70] L. Putelli, A. Gerevini, A. Lavelli, and I. Serina. The impact of self-interaction attention on the extraction of drug-drug interactions. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*, 2019.

[71] L. Putelli, A. E. Gerevini, A. Lavelli, and I. Serina. Applying self-interaction attention for extracting drug-drug interactions. In *International Conference of the Italian Association for Artificial Intelligence*, pages 445–460. Springer, 2019.

[72] L. Putelli, A. Gerevini, A. Lavelli, M. Olivato, and I. Serina. Deep learning for classification of radiology reports with a hierarchical schema. In *Proceedings of 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, 2020.

[73] L. Putelli, A. E. Gerevini, A. Lavelli, R. Maroldi, and I. Serina. Attention-based explanation in a deep learning model for classifying radiology reports. In A. Tucker, P. H. Abreu, J. S. Cardoso, P. P. Rodrigues, and D. Riaño, editors, *Artificial Intelligence in Medicine - 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15-18, 2021, Proceedings*, volume 12721 of *Lecture Notes in Computer Science*. Springer, 2021.

[74] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages, Biology and Medicine (LBM 2013)*, pages 39–44, 2013.

[75] C. Quan, L. Hua, X. Sun, and W. Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed research international*, 2016, 2016.

[76] C. Raffel and D. P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems. *CoRR*, abs/1512.08756, 2015.

[77] A. Rajkomar, J. Dean, and I. Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[78] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First International Conference on Machine Learning*, pages 133–142, 1999.

[79] N. Reamaroon, M. W. Sjoding, K. Lin, T. J. Iwashyna, and K. Najarian. Accounting for label uncertainty in machine learning for detection of acute respiratory distress syndrome. *IEEE Journal of Biomedical and Health Informatics*, 23(1):407–415, 2019.

[80] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[81] X. Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014. URL `http://arxiv.org/abs/1411.2738`.

[82] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[83] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer, 2010.

[84] S. Serrano and N. A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

[85] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60, 2019. doi: 10.1186/s40537-019-0197-0. URL `https://doi.org/10.1186/s40537-019-0197-0`.

[86] X. Sun, L. Ma, X. Du, J. Feng, and K. Dong. Deep convolution neural networks for drug-drug interaction extraction. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1662–1668, 2018.

[87] I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1452. URL https://doi.org/10.18653/v1/p19-1452.

[88] T. M. Thiyagu, D. Manjula, and S. Shridhar. Named entity recognition in biomedical domain: A survey. *International Journal of Computer Applications*, 181(41):30–37, Feb 2019. ISSN 0975-8887. doi: 10.5120/ijca2019918469. URL http://www.ijcaonline.org/archives/volume181/number41/30336-2019918469.

[89] B. van Aken, B. Winter, A. Löser, and F. A. Gers. Visbert: Hidden-state visualizations for transformers. In A. E. F. Seghrouchni, G. Sukthankar, T. Liu, and M. van Steen, editors, *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 207–211. ACM / IW3C2, 2020. doi: 10.1145/3366424.3383542. URL https://doi.org/10.1145/3366424.3383542.

[90] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. ISBN 0-408-70929-4.

[91] S. Vashishth, S. Upadhyay, G. S. Tomar, and M. Faruqui. Attention interpretability across NLP tasks. *CoRR*, abs/1909.11218, 2019. URL http://arxiv.org/abs/1909.11218.

[92] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In

*Advances in neural information processing systems*, pages 5998–6008, 2017.

[93] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017. URL `http://arxiv.org/abs/1710.10903`.

[94] J. Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019. URL `https://arxiv.org/abs/1906.05714`.

[95] L. Wang, X. Ruan, P. Yang, and H. Liu. Comparison of three information sources for smoking information in electronic health records. *Cancer Informatics*, 15:237 – 242, 2016.

[96] M. Wasim, W. Mahmood, and U. G. Khan. A survey of datasets for biomedical question answering systems. *International Journal of Advanced Computer Science and Applications*, 8(7):484–488, 2017.

[97] G. Weiss and F. Provost. The effect of class distribution on classifier learning: An empirical study. Technical report, Department of Computer Science, Rutgers University, 2001.

[98] S. Wiegreffe and Y. Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, 2019.

[99] M. Wieling, J. Rawee, and G. van Noord. Reproducibility in computational linguistics: Are we willing to share? *Computational Linguistics*, 44(4):641–649, 2018.

[100] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

[101] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical attention networks for document classification. In *HLT-NAACL*, 2016.

[102] Z. Yi, S. Li, J. Yu, Y. Tan, Q. Wu, H. Yuan, and T. Wang. Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In *International Conference on Advanced Data Mining and Applications*, pages 554–566. Springer, 2017.

[103] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[104] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C14-1220`.

[105] Y. Zhang and A. Nie. Inducing grammar from long short-term memory networks by shapley decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 299–305, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-srw.40`.

[106] Y. Zhang, P. Qi, and C. D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1244. URL `https://www.aclweb.org/anthology/D18-1244`.

[107] Y. Zhang, W. Zheng, H. Lin, J. Wang, Z. Yang, and M. Dumontier. Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. *Bioinformatics*, 34(5):828–835, 2018.

[108] J. Zheng, F. Cai, T. Shao, and H. Chen. Self-interaction attention mechanism-based text representation for document classification. *Applied Sciences*, 8(4), 2018. ISSN 2076-3417. doi: 10.3390/app8040613. URL `http://www.mdpi.com/2076-3417/8/4/613`.

[109] W. Zheng, H. Lin, L. Luo, Z. Zhao, Z. Li, Z. Yijia, Z. Yang, and J. Wang. An attention-based effective neural model for drug-drug interactions extraction. *BMC Bioinformatics*, 18, 12 2017. doi: 10.1186/s12859-017-1855-x.

[110] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.