ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA
IN
INTELLIGENT SYSTEMS

Predizione della struttura di un argomento con *feature* di *stance classification*

Candidato:
Federico Ruggeri

Matricola 0000785737

Relatore:
Chia.mo Prof. Paolo Torroni

Correlatore:
Dott. Marco Lippi

Anno Accademico 2016–2017
Sessione III

Indice

Introduzione

Nell'ultimo decennio i settori di ricerca nell'ambito dell'elaborazione automatica del linguaggio naturale (**NLP**) hanno acquisito sempre maggiore interesse, in concomitanza con l'introduzione continua di strumenti di *machine learning* innovativi. A tale proposito, si è assistito alla rapida definizione di alcuni importanti rami distinti quali la *sentiment analysis*, *l'opinion mining* e la *argumentation mining*, atti ad indagare, all'interno di determinate dimensioni semantiche individuabili a partire dai documenti testuali presi in esame, parte delle molteplici sfaccettature proprie del processo razionale umano. In particolar modo, seppur sia comunque presente un'ambizione ideologica, l'interesse principale è riconducibile ad un piano prettamente applicativo, di natura, ad esempio, sociale o politica, dove l'introduzione di nuove e particolari informazioni può apportare un notevole miglioramento in molti processi di interesse reale e contribuire all'indagine di nuove tendenze e/o conflitti nei confronti di determinati riferimenti specifici del contesto applicativo. Ad esempio, nel caso della *stance classification*, uno dei principali settori emergenti all'interno della *sentiment analysis*, la definizione di informazioni di tipo descrittivo, relative ad un campione della popolazione nei confronti di uno specifico individuo, prodotto o istituzione, e ottenute a partire dall'elaborazione di semplici documenti testuali da parte di strumenti computazionali, può certamente contribuire positivamente a chiarire maggiormente la situazione di interesse o a colmare eventuali mancanze. Ancora, nell'ambito della *argumentation mining*, la capacità di poter individuare e generare in maniera automatica gli argomenti per contesti controversi presenta un grande valore dal punto di vista pratico, se si considerano domini di tipo politico, giuridico, finanziario e altri ancora dove il valore persuasivo assume un ruolo preponderante. Pertanto, il potenziale riscontrato nei presenti ambiti di ricerca ha attirato l'attenzione di aziende di notevole

portata, al punto da finanziare progetti ambiziosi volti a rispondere a necessità emergenti proprie di molteplici realtà di diversa natura; un importante esempio è rappresentato dall'azienda **IBM** con l'introduzione del progetto *Debater*, avente come obiettivo la raccolta di ingenti quantitativi di informazioni dal web da elaborare successivamente ricorrendo a processi che simulano quello razionale umano. A partire da queste premesse, il presente elaborato ha l'intento di analizzare due specifici ambiti di ricerca tra quelli precedentemente introdotti, quali la *sentiment analysis* e l'*argumentation mining*. Nello specifico, si pone come obiettivo la definizione di un modello atto a individuare punti di affinità tra due specifiche attività di questi ultimi e di recente interesse, denominate *stance classification* e *argument structure prediction*, volte ad indagare rispettivamente il posizionamento di un dato individuo nei confronti di un particolare campo di interesse e i legami relazionali tra molteplici espressioni argomentative o tra i singoli componenti individuati per ciascuna. In particolare, la sperimentazione congiunta di diverse tecniche di elaborazione testuale può potenzialmente apportare un notevole miglioramento reciproco, in quanto generalmente i differenti ambiti e obiettivi di ciascun settore di ricerca specifico presentano innumerevoli somiglianze e *modi operandi* comuni, fattore che rispecchia perfettamente la natura collettiva delle presenti branche nell'ottica del singolo e generale processo razionale umano. Dal punto di vista strutturale il presente elaborato è organizzato come segue: i primi due capitoli introducono rispettivamente i concetti di *stance classification* e *argumentation mining*, soffermandosi, in particolar modo, sulle tecniche e le metodologie impiegate ed evidenziando al tempo stesso le differenze con altri settori di ricerca relativamente simili. Successivamente, il terzo capitolo delinea inizialmente gli obiettivi generali seguiti dall'elaborato proposto e di cui il presente descrive e analizza solamente il primo di essi. Nello specifico, il tema affrontato riguarda la definizione di uno strumento di classificazione atto a raggiungere lo stesso livello di performance proprio dello stato dell'arte nell'ambito della *stance classification*. In seguito, il quarto capitolo si interessa principalmente dell'individuazione di un legame tra quest'ultimo settore di ricerca e l'*argument structure prediction*, indagando inizialmente la possibilità di usufruire delle informazioni specifiche della *stance classification* per migliorare l'attività di classificazione. Successivamente, vengono messe a confronto alcune tecniche specifiche

attualmente impiegate per l'*argument structure prediction* con le principali adottate nell'ambito della *stance classification*. Infine, nel quinto e ultimo capitolo viene proposto un nuovo corpus per la sperimentazione degli stessi approcci impiegati nel capitolo precedente, con particolare attenzione a proporre degli opportuni test di confronto e di verifica.

Capitolo 1

La Stance Classification (SC)

Con l'avvento del web 2.0, hanno acquisito sempre più familiarità strumenti quali *social media*, forum di dibattito on-line e altri ancora atti a rendere visibili i pensieri e le opinioni degli utenti. Nel presente contesto, ambiti di ricerca aventi come obiettivo l'elaborazione computazionale di tali dati sono diventati sempre più oggetto di interesse all'interno di settori quali l'intelligenza artificiale, l'elaborazione del linguaggio naturale (**NLP**) e il *text mining*[55][163]. Tra le diverse possibili sfaccettature riconducibili ai suddetti settori, si è inserita recentemente ed è tutt'ora ambito di esplorazione, la *stance classification*. Quest'ultima si pone come obiettivo l'individuazione del posizionamento, i.e. *stance*, di un dato frammento di informazione, quale ad esempio un documento testuale, riconducibile al suo autore, nei confronti di uno specifico elemento di riferimento, identificato con il termine *target*. In questo capitolo, viene introdotto ed esplorato con attenzione il termine *stance classification*, descrivendo in particolar modo lo stato dell'arte attuale e cercando al tempo stesso di evidenziarne le differenze con altri settori di ricerca affini. Come verrà mostrato successivamente all'interno del presente elaborato, la *stance classification* non rappresenta solamente un nuovo interesse all'interno della comunità scientifica, bensì porta con sè innumerevoli applicazioni che possono portare beneficio ad altri campi di ricerca propri del **NLP** e del *text mining*.

1.1 Definizione

Come già accennato precedentemente, la *stance classification* viene generalmente descritta come il procedimento che consente di determinare in maniera automatica se la posizione (*stance*) assunta dall'autore di una determinata fonte di informazione, quale ad esempio un documento testuale, risulta essere a favore, in opposizione o neutrale nei confronti di un preciso concetto, argomento, individuo, istituzione, organizzazione, prodotto, politica e altro ancora. Pertanto, si identificano generalmente tre possibili valori attribuibili alla *stance*[143][144][159][145][60]: `pro/favor/for`, `con/against/` e `neutral/observing`. Tuttavia, la prevalenza degli studi effettuati in tale ambito, definisce la *stance classification* come un problema di risoluzione binaria, escludendo di conseguenza la posizione di neutralità nei confronti di un dato *target*. Nonostante tale definizione abbia validità pressoché generale, recentemente sono stati effettuati nuovi studi volti a raffinare il valore semantico attribuibile alla *stance*, associando ad esempio gradi di intensità a ciascuno dei valori possibili, come ad esempio `strongly for` e `strongly against`[140] o focalizzandosi su un nuovo set di possibili etichette tra cui scegliere durante la classificazione[138][169][129][36], dettato dallo specifico caso di studio. Quanto osservato, è riportato in maniera schematica nella tabella 1.1. Questo fattore denota la natura mutevole e in continua evoluzione dell'ambito di ricerca preso in esame, soggetto a nuove sperimentazioni e studi volti ad esplorare con cura le applicazioni potenziali.

1.2 Ambiti e Applicazioni

L'analisi dei costrutti linguistici e dei termini per mezzo dei quali gli individui esprimono i propri giudizi, opinioni e ideali, viene riassunta in unico termine denominato *stance*. Lo sfruttamento di tali informazioni sintattiche e semantiche, individuabili all'interno delle azioni di comunicazione di ciascun interlocutore, assume particolare interesse per studi in ambito sociale. Pertanto, la possibilità di individuare in maniera automatica la *stance* nei confronti di un determinato riferimento di interesse apre la strada verso innumerevoli applicazioni. In questa sezione, vengono descritti gli ambiti di interesse sui quali i ricercatori hanno focalizzato maggiormente la loro attenzione, sottolineandone le relative possibili applica-

Valori attribuibili alla stance	Lavori
{For/Favor, Against, Observing/Neither}	[46], [68], [58], [8], [40], [7]
{For/Favor, Against}	[60], [93], [41], [160], [45], [59], [156], [3], [23], [144], [143], [128]
{Strongly For, For, Strongly Against, Against}	[140]
{Pro, Con/Anti, None/Neutral/Other}	[153], [37], [99], [145]
{Pro, Con}	[14], [146], [159], [125]
{Contrariety, Source of Knowledge, Prediction, Necessity, Uncertainty, Hypotheticality}	[138]
{Pro-independence (PI), anti-independence (AI)}	[169]
{Implicit, Explicit}	[129]
{Ideological, Non-ideological}	[36]
{Support, Oppose}	[106], [13], [151]
{Support, Oppose, Proposes a new idea}	[76]

Tabella 1.1: Tipologie di set di valori sperimentati per la *stance classification*. In particolare, per ognuno di essi sono associati tutti i lavori che ne fanno uso. Per motivi di spazio, sono riportati solamente i riferimenti alla bibliografia.

zioni e in particolar modo le difficoltà riscontrate. La ricerca nell'ambito della *stance classification* si concentra principalmente sulle diverse tipologie di documenti testuali ottenibili dalle innumerevoli fonti disponibili oggigiorno, focalizzandosi in particolar modo sugli strumenti on-line, facilmente reperibili e usufruibili da un ingente quantitativo di utenti. Come conseguenza, uno degli aspetti cruciali è rappresentato per l'appunto dallo scenario di interesse, il quale può spaziare dai più comuni *social media*, quale Twitter, ai siti o forum di dibattito on-line, fino a considerare contesti più formali quali quello legislativo o educativo. Procedendo con ordine, strumenti quali *social media*, *microblog* e forum di discussione on-line socio-politici pubblici interessano quotidianamente un ingente quantitativo di utenti. Tra i più importanti vi è il precedentemente citato Twitter, che si è affermato come una delle principali risorse per lo scambio di informazioni e opinioni. Solo recentemente la ricerca ha rivolto il proprio interesse verso lo sviluppo di strumenti e modelli computazionali atti ad analizzare il contenuto dei dati esposto da strumenti quale Twitter. Tale aspetto acquista notevole importanza se si considera il ruolo sempre più incisivo che stanno assumendo i *social media*. In particolar modo, essi, in quanto ricchi di flussi di dati inerenti ad opinioni, pareri e ragionamenti, contribuiscono a rendere di maggior interesse lo studio della *stance classification*, poiché informazioni aggiuntive di tipo statistico, quale la distribuzione dell'opinione della popolazione rispetto a un determinato campo di interesse, possono favorire la formulazione e la definizione di soluzioni atte a rispondere adeguatamente alla situazione specifica. Un esempio di facile intuizione è dato dalle elezioni presidenziali, all'interno delle quali avere la possibilità di effettuare studi relativi al posizionamento della popolazione o dei personaggi politici direttamente interessati può consentire di identificare eventuali tendenze, conflitti e/o problematiche. Tuttavia, l'analisi di tali dati rappresenta tutt'oggi un ostacolo difficile da superare. Una simile problematica trova fondamento nelle caratteristiche linguistiche e sintattiche dei testi pubblicati dagli utenti, i quali sono ricchi di termini informali, *emoji*, *hashtag*, errori grammaticali e di scrittura, termini dialettali e gergali, parole onomatopeiche, caratteri ripetuti e termini personalizzati. Chiaramente tale abbondanza di informazioni, estraibile da strumenti quali *social media* per l'appunto, può avere diverse finalità, tra le quali si colloca la *stance classification*. Altri esempi di notevole interesse riguardano la costruzione di modelli relativi

alle preferenze dell'utente[128] e lo studio della localizzazione geografica dell'utenza[70][127]. Tuttavia, la ricerca nell'ambito della *stance classification* ha ritenuto preferibile soffermarsi, inizialmente, su contesti di minore complessità rispetto ai *social media* e ai *microblog*, quali dibattiti congressuali[151][167][23][13][12], discussioni interne a aziende[3][106], siti di dibattito online[164][142][144][5][159][60][146][20], articoli di tipo giornalistico[46], saggi brevi redatti da studenti[45] e commenti pubblici relativi ai regolamenti dell'agenzia delle entrate[76], caratterizzati nella maggior parte da limitazioni sintattiche e linguistiche in accordo con una specifica struttura e impostazione. Infatti, l'analisi di documenti testuali tradizionali differisce da quella relativa agli utenti di *social media* e *microblog*, in quanto nel primo caso la *stance* è spesso espressa in maniera esplicita. Viceversa, l'analisi dei profili degli utenti iscritti ai vari social media necessita di dati aggiuntivi relativi alla cronologia delle azioni associata a ciascun account, quali ad esempio collegamenti con altri utenti e interessi. Tra i vari scenari precedentemente introdotti, lo studio inerente alla *stance classification* in ambiti quali i siti di dibattito online comporta innumerevoli implicazioni importanti, in quanto spesso vengono trattati argomenti di natura politica o giornalistica. Formalmente si parla di *debate stance classification*[61]. Pertanto, la capacità di poter ottenere informazioni pertinenti e di valore, quale la *stance* nei confronti di un dato riferimento, può essere di ausilio per ulteriori ricerche, organizzazioni governative e compagnie. Ad esempio, la possibilità di predire la *stance* di un determinato gruppo di utenti, consente di poter individuare gruppi sociali e politici, migliorare la conoscenza nei confronti dei singoli utenti nel rispetto delle loro convinzioni e ideali[169] e infine può agevolare la definizione di sistemi di supporto, quali raccomandazione e filtraggio migliori[50][1][5][125][60]. Nell'attuale contesto di interesse, emerge un'ulteriore distinzione di granularità maggiore rispetto a quanto osservato precedentemente nell'ottica dei differenti scenari di studio. Più precisamente, si distingue tra situazioni di dibattito tradizionali, quali discussioni interne a compagnie e dibattiti politici, e siti o forum di dibattito on-line. Nel primo caso, le informazioni di interesse seguono una specifica struttura e impostazione in accordo con un'ottica formale. Viceversa, i dibattiti on-line abbondano di locuzioni linguistiche quali sarcasmo, insulti, domande, risposte, emozioni e insinuazioni[8]. Come conseguenza, tale distinzione complica notevolmente l'applicazione di studi associati

alla *stance classification* in quest'ultimo contesto, rappresentando ancora oggi una sfida aperta. In aggiunta ai diversi ambiti di ricerca descritti fino ad ora, recentemente sta acquistando sempre più interesse un ulteriore scenario di applicazione per la *stance classification*[14], legato al *argumentation mining*, concetto che si pone come obiettivo l'estrazione degli argomenti all'interno del testo e che verrà esplorato ampiamente nel capitolo 2. In particolar modo, si parla di *claim stance classification*[14], dove con il termine *claim* si intende un piccolo frammento di testo, spesso coincidente con una singola frase, rappresentante una conclusione, un ragionamento o un giudizio nei confronti di un dato argomento (*topic*) di riferimento. In tale ottica, a differenza del contesto di Twitter proposto da Mohammad et al.[98], il caso di studio preso in esame presenta la sostanziale differenza che il *target* della *stance* non è noto a priori, può risultare arbitrariamente complesso ed essere legato a qualunque dominio di interesse senza alcun tipo di restrizione. Pertanto non risulta possibile usufruire delle informazioni specifiche legate ad esso per migliorare la classificazione ed è quindi necessario ricorrere a tecniche quali la *sentiment analysis*. Un ulteriore esempio legato al *argumentation mining* è riportato da Rajendran et al.[129], in cui si pone come primo obiettivo di un più grande progetto l'applicazione della *stance classification* per distinguere tra opinioni esplicite e implicite all'interno di documenti testuali quali le recensioni di utenti su TripAdvisor[158], al fine di poter costruire degli argomenti validi e completi. Più precisamente, nel caso di opinioni implicite può essere necessario introdurre dati aggiuntivi, quali premesse per uno specifico ragionamento (*premise* o *evidence*) al fine di poter delineare un vero e proprio argomento. In caso contrario, si ottiene come risultato un argomento incompleto, denominato entimema. Infine, un'altra finalità che può avere risvolti significativi nell'ambito della *stance classification*, interessa direttamente scenari critici in cui un'analisi in tempo reale o soggetta a restrizioni temporali ben definite è di fondamentale importanza. Un esempio è dato dall'identificazione di atti di violenza[28] e dal rilevamento di possibili situazioni di disastro. Concludendo, possiamo riassumere il discorso fatto constatando il fatto che la possibilità di individuare in maniera automatica la *stance* nei confronti di un determinato riferimento di interesse, apre la strada verso innumerevoli applicazioni tra le quali, ad esempio, l'ottenimento di informazioni descrittive associabili ai dati presi in oggetto e la capacità di sintetizzare

i testi analizzati in modo tale da poter rendere possibile un confronto in tempo reale verso il riferimento di interesse preso in esame, i.e. il *target*.

1.3 Stance classification e Sentiment Analysis

Come osservato precedentemente all'inizio del capitolo, la *stance classification* è uno dei principali settori emergenti all'interno della *sentiment analysis*. Tuttavia, è importante precisare le differenze sostanziali che sussistono tra i due concetti. Generalmente la *sentiment classification* viene formulata come il processo per mezzo del quale si cerca di determinare la polarità di un documento testuale, quale essa sia positiva, negativa o neutrale, oppure l'opinione dell'interlocutore e il *target* dell'opinione a partire dal testo, ovvero l'entità verso la quale l'opinione è espressa. Viceversa, per quanto riguarda la *stance classification*, l'analisi è rivolta verso un determinato *target* di riferimento, nei confronti del quale si può definire una posizione favorevole, contraria o nessuna delle due. In tale senso, la *stance classification*, etichettata formalmente con il termine *target-specific stance detection*[37], si avvicina al concetto di *target-dependent sentiment classification*[69], con la differenza sostanziale che il *target* della *stance* può non essere menzionato esplicitamente all'interno del documento testuale oppure può non coincidere con il target dell'opinione[97]. In altre parole, la differenza principale tra *stance classification* e la più tradizionale *sentiment classification* è che l'identificazione della *stance* è strettamente legata sia alle espressioni soggettive individuate all'interno del documento, che al *target* di riferimento, il quale può non essere menzionato in maniera esplicita all'interno del testo. Ciò indica che al di là del contenuto testuale, proprio del documento preso in analisi, le informazioni associate al *target* sono importanti per la *stance detection*. In particolare, la presenza o meno del *target* di interesse può portare il modello computazionale in errore quando deve predire la *stance*. Un chiaro esempio è riportato da Liu e Zhang[85], dove la *stance* è contraria al *target* di riferimento, ovvero l'aborto, ma quest'ultimo concetto non compare all'interno del testo e di conseguenza deve essere inferito:

> *"We remind ourselves that love means to be willing to give until it hurts."*

Un secondo esempio atto a sottolineare le differenze tra i due concetti presi in esame, ovvero *stance classification* e *sentiment analysis*, concerne direttamente i costrutti linguistici utilizzati per esprimere la posizione dell'autore di un documento informativo o la propria opinione[45]. Più precisamente, come osservato nei lavori di Hunston e Thompson[152] e di Martin e White[126], gli scrittori e gli interlocutori assumono una posizione nei confronti di determinate proposizioni mediante l'utilizzo di indicatori probatori quali verbi e avverbi modali, i quali modificano l'intera frase. Viceversa, l'inserimento di aggettivi, come 'grande' o 'terribile', interessa esclusivamente il gruppo nominale associato. Ciò ha portato i ricercatori a distinguere tra *opinion-bearing language* e *stance taking language*, basandosi su informazioni quale la classe semantica del *target* dell'opinione o la *stance*. Esempi di tale distinzione sono forniti da Faulkner[45] e che riportiamo di seguito come ausilio per comprendere chiaramente quanto osservato. In particolare, la frase (1) evidenzia in grassetto un esempio di *opinion-bearing language*, mentre le frasi (2) e (3) rappresentano il concetto di *stance taking language*.

> *(4) "Snake Eyes" is the most **aggravating** kind of [movie]: the kind that shows so much potential and then becomes **unbelievably disappointing**. (opinion=positive)*
>
> *(5) This **indicates** that [our prisons are higher institutions for criminals].(stance=for)*
>
> *(6) So we **can infer** that [the statement is very true]. (stance=for)*

Riassumendo, l'*opinion mining* considera le entità o gli individui come *target*, mentre nel caso della *stance* tale ruolo è assunto dalle proposizioni di interesse.

1.4 Tecniche e metodologie di classificazione

Una volta introdotti ed esplorati i molteplici ambiti di interesse relativi alla *stance classification*, risulta di fondamentale importanza ripercorrerli nuovamente dal punto di vista tecnico della classificazione, al fine di marcare in maniera più significativa le differenze evidenziate precedentemente e di introdurre ulteriori distinzioni nell'ottica presa in esame, quali metodologie, paradigmi di apprendimento e le caratteristiche testuali e di contesto estratte e usate come input per la classificazione. A tal fine, nel cercare di raggiungere l'obiettivo imposto, mantenere lo stesso ordine di presentazione dei contesti di ricerca, introdotto nella sezione 1.2 e che hanno e continuano tutt'oggi ad interessare la *stance classification*, è di ausilio ad una maggiore comprensione e alla comparazione su più livelli dei concetti principali.

1.4.1 *Feature* di interesse

Per quanto riguarda il contesto dei *social media* e *microblog*, quale Twitter, per motivi legati direttamente alla natura dei dati esposti da tali strumenti, un simile ambito comporta non poche difficoltà nell'ottica della *stance classification*. Più precisamente, i *tweet* presentano nuove sfide che precludono l'utilizzo delle caratteristiche del testo (*feature*) più canoniche[35]. Una simile considerazione, trova spiegazione nel fatto che i *tweet* hanno lunghezza limitata, i.e. 140 caratteri al massimo, fattore che alle volte porta l'autore del documento a produrre testi poco strutturati e coerenti. Inoltre, tutto ciò è aggravato ulteriormente dalla grande quantità di termini informali e dalla potenziale presenza di errori grammaticali nel testo. Pertanto, nell'ottica di tale ragionamento, prevalgono classificatori basati su *feature* relativamente semplici, quali *bag of words* (**BoW**), *word n-gram* e *character n-gram*, in aggiunta a modelli complessi quali quelli neuronali[7][156], in grado di estrarre in maniera automatica le caratteristiche di interesse dal testo[98]. Tuttavia, si distinguono anche ulteriori approcci che si discostano da quelli più canonici e frequentemente utilizzati, come ad esempio l'identificazione di dipendenze relazionali e sintattiche[93], grazie a strumenti quali il *Linguistic Inquiry Word Count*[120] (**LIWC**), e l'utilizzo di set di regole basate sulla sintassi e sulla struttura discorsiva dei *tweet* per distinguere dagli altri

quelli contenenti pensieri e ideologie nei confronti di un dato *target* di riferimento[36]. In aggiunta, cercando di usufruire delle particolarità testuali proprie dei dati esposti da Twitter, sono state sperimentate anche *feature* ortografiche, *hashtag*, il *sentiment* inferito all'interno del testo ricorrendo all'ausilio di opportuni lexicon[100][96][101][165][65] e strumenti di analisi[66], e i *word* e *character skip n-gram*, per le quali si assume che contengano importanti informazioni da un punto di vista sociale e quindi in grado di catturare correttamente la dimensione semantica del testo[58]. In particolar modo, nonostante i punti di distacco con la *stance classification*, nell'ambito di Twitter, l'aggiunta del *sentiment* al set di *feature* per la classificazione, ha mostrato chiari segni di miglioramento delle performance di quest'ultima, anche se è importante tenere a mente che basarsi esclusivamente su di esso non risulta comunque sufficiente per affrontare con successo scenari complessi come quelli dei *social media*[99]. Infine, oltre alle metodologie più tradizionali, focalizzate esclusivamente sulle proprietà testuali dei documenti esposti da Twitter o altri *social media*, sono state sperimentate nuove tecniche volte a sfruttare informazioni di contesto proprie di tali strumenti, quali la rete di collegamenti dell'utente, la cronologia delle azioni e i propri interessi. Ad esempio, Rajadesingan e Liu[128], hanno determinato la *stance* degli utenti basandosi sull'ipotesi che se diversi autori condividono *tweet* concordi relativi ad un particolare argomento (*topic*) controverso, allora con molta probabilità essi assumono la stessa posizione all'interno del dibattito. Quanto osservato nell'ambito dei *social media* e *microblog*, può essere in gran parte applicabile anche in contesti di più facile analisi, quali ad esempio i forum e i siti di dibattito. In particolare, si considerano per la maggior parte dei casi dibattiti bilaterali legati prevalentemente ad argomenti ideologici controversi, quali i diritti per gli omosessuali e l'aborto. In tali scenari, le informazioni aggiuntive, quali le etichette per la *stance classification* sono spesso fornite dagli stessi autori dei *post*, il che consente di confrontare i risultati della classificazione con i relativi valori reali, al fine di valutarne le performance. Analogamente a quanto osservato per Twitter, sono stati sperimentati classificatori basati su *feature* quali **BoW**, *n-gram* e simboli di punteggiatura, con l'aggiunta di dipendenze sintattiche e relazionali[160], grazie a strumenti quale il **LIWC**, e informazioni relative alla struttura dialogica dei *post* presi in analisi, quali accordo e disaccordo, al fine di sfruttare a proprio vantaggio le nozioni aggiuntive

di contesto proprie dei dibattiti[5][159]. In aggiunta, sono state oggetto di studio anche semplici *baseline* basate su *feature*, quali verbi modali e *sentiment*, grazie all'ausilio di *lexicon*[165] e strumenti di analisi, atti ad agevolare l'individuazione di argomenti all'interno del testo[144]. Infine, si è arrivati a definire classificatori basati su *feature* più complesse, legate a specifici costrutti volti a catturare la dimensione semantica del testo[60][61], quali i *frame* estratti mediante gli strumenti *FrameNet*[10] e *SEMAFOR*[33]. Per quanto riguarda gli altri contesti di interesse oltre ai semplici siti di dibattito, sono state sperimentate anche set di *feature* specifiche in base al particolare caso di studio considerato. Ad esempio, Faulkner[45] ha analizzato il problema di individuare la *stance* a livello di documento all'interno di saggi brevi redatti da studenti, sulla base di informazioni legate alla polarità espressa dalle parole, al concetto definito dalla proposizione a cui la *stance* fa riferimento e alla relazione a livello di linguaggio tra le richieste delineate dal saggio di domanda e da quello di risposta. Un ulteriore esempio è dato da Sobhani et al.[140], i quali hanno estratto gli argomenti all'interno di commenti relativi alle notizie su siti di comunicazione on-line per individuare la *stance*. Infine, Bar-Haim et al.[14], per via delle forti restrizioni date dal loro dominio di ricerca, vertono la propria attenzione prevalentemente sull'utilizzo di informazioni proprie della *sentiment analysis*, piuttosto che affidarsi all'estrazione di *feature* specifiche per il *topic* o del contesto specifico. In particolar modo, si concentrano sull'analisi semantica del *topic* e del *claim*, includendo il processo di individuazione dei loro rispettivi *target* e l'identificazione del legame di contrasto tra questi ultimi.

1.4.2 Classificazione collettiva e sequenziale

Oltre alle semplici metodologie di ricerca, concentrate esclusivamente sulla sperimentazione di differenti set di *feature* volte a modellare i documenti testuali di interesse, recentemente la *stance classification* è stata affrontata considerando gruppi collettivi di documenti testuali per ottenere ulteriori miglioramenti nella determinazione della *stance*[151][167][23][60][159][146]. Ad esempio, nell'ambito dei *social media*, sono state esplorate tecniche di classificazione, quale l'utilizzo di strumenti come il *hinge-loss Markow random field* (**HL-MRFs**), su collezioni di *stance* estratte dai documenti forniti da Twitter[41]. D'altro canto, informazioni quali citazioni[23] o

confutazioni tra documenti testuali (*rebuttal link*)[161] sono state utilizzate come dati aggiuntivi per modellare i legami di accordo e disaccordo tra *post* e per inferire le *stance* associate ad essi. Sulla base di tale metodologia, si inseriscono numerosi esperimenti, distinti dai particolari algoritmi impiegati nel considerare i legami tra *post*. Un esempio è dato da Murakami e Raymond[106], i quali hanno sperimentato l'algoritmo *maximum cut* per aggregare tra loro le *stance* associate a molteplici *post*, al fine di inferire la *stance* del loro autore nel rispetto del *target* di riferimento. Ancora, Sridhar et al.[146][145] propongono modelli di classificazione collettiva, uno dei quali sperimenta l'uso della *Probabilistic Soft Logic* (**PSL**), basati sul principio che considerare contemporaneamente gruppi di *post*, insieme alle informazioni dei loro autori, consente di classificare correttamente la *stance* di un dato *post* preso in esame. In maggior dettaglio, si è dimostrato come la modellazione congiunta di *stance*, insieme ad informazioni relative al disaccordo tra *post*, comporti un miglioramento della classificazione, rispetto ai modelli che operano esclusivamente sul contenuto dei *post* stessi e sul contesto del *post* padre a cui fanno riferimento[159]. Più precisamente, tale miglioramento si basa sulla semplice osservazione che il disaccordo tra autori è un chiaro segnale che le loro *stance* saranno differenti. Il concetto è maggiormente rafforzato soprattutto se si presta attenzione allo scenario preso in esame. Oltre alla classificazione collettiva, è possibile sfruttare la struttura dei siti e forum di dibattito, con attenzione particolare al modo in cui sono raggruppati i *post* all'interno di una discussione. In maggior dettaglio, dato che gli argomenti e le relative opposizioni sono posti in sequenza, dove ogni *post* segue cronologicamente quello a cui si riferisce e risponde, Hasan e Ng[61] hanno proposto un modello di *stance classification* all'interno dei forum di dibattito impostato come un processo di etichettatura di una sequenza di documenti (*sequence labeling*), ricorrendo ad un metodo di inferenza globale per classificare correttamente i *post*.

1.4.3 Importanza della *background-knowledge*

Un altro filone nell'ambito della *stance classification*, basato sempre sul principio che informazioni aggiuntive quali quelle di contesto possano favorire la classificazione e in generale molte attività di **NLP**, usufruisce di regole e nozioni legate agli autori dei documenti testuali propri del

dominio di interesse nel tentativo di colmare le mancanze che un'analisi prettamente testuale presenta. Formalmente, tale metodologia viene riferita con il termine *background-knowledge*. Ad esempio, Hasan e Ng[59] si pongono l'obiettivo di voler catturare gli effetti delle ideologie sulla *stance*. Nel tentativo di perseguire il loro fine, ricorrono alla definizione di vincoli extra linguistici tra *post*, detti *Ideology Constraints* (**IC**). Quest'ultimi interessano più domini (*cross-domain*) e sono applicabili su tutti gli argomenti in cui uno specifico autore di interesse partecipa. Più precisamente, gli **IC** sono definiti in base al principio per cui la *stance* di un dato autore, riguardo ad uno specifico argomento, è in parte determinata dalle sue ideologie e che quindi può sussistere una correlazione tra le *stance* legate a lui su argomenti differenti. Dal punto di vista implementativo, tali regole sono convertite in vincoli lineari tramite *Integer Linear Programming* (**ILP**), limitando le probabilità relative ai suddetti **IC** attraverso l'introduzione di opportune soglie, determinate sulla base dei dati di sviluppo. Tuttavia, una simile metodologia presenta dei difetti sostanziali, come ad esempio non è possibile catturare la correlazione tra il disaccordo e l'ideologia tra autori e non sono nemmeno in grado di estrapolare l'uso dell'ideologia per predire la *stance* su domini non appartenenti al set di apprendimento[8]. Infine, un ulteriore esempio è riportato da Toledo-Ronen et al.[153], i quali introducono una nuova risorsa da cui attingere, atta a supportare diverse attività legate alla *argumentation mining* e alle tecnologie di dibattito. Tale risorsa, detta *Expert Stance Graph*, è costruita a partire da Wikipedia ed è volta a fornire informazioni di contesto sulla *stance* degli esperti rispetto ad argomenti controversi. Più precisamente, l'importanza di un simile progetto risiede nella possibilità di incrementare l'ammontare di informazioni disponibile, ottenuto prevalentemente da analisi testuali e di contesto. In particolar modo, avere a disposizione dati generali legati agli autori dei documenti consente di poter formulare un quadro completo del dominio di interesse, consentendo pertanto di comprendere aspetti semantici impliciti e di migliorare conseguentemente la classificazione. Per comprendere meglio quest'ultimo concetto, riportiamo lo stesso esempio riportato da Toledo-Ronen et al.[153] in cui la determinazione della *stance* risulta inizialmente difficile, ma l'utilizzo di informazioni proprie dell'autore consente di risolvere facilmente il problema.

Dawkins sums up his argument and states, "The temptation

> *(to attribute the appearance of a design to actual design itself) is a false one, because the designer hypothesis immediately raises the larger problem of who designed the designer. The whole problem we started out with was the problem of explaining statistical improbability. It is obviously no solution to postulate something even more improbable." (Dawkins, 2006, p. 158)*

In particolare modo, se si considera la pagina Wikipedia di Richard Dawkins, si evince da diversi dati in maniera molto esplicita come l'autore sia un noto ateista:

- Categorie: Dawkins appartiene alle seguenti categorie di Wikipedia: *Antitheists*, *Atheism activists*, *Atheist feminists and Critics of religions*.
- Testo: L'articolo contiene affermazioni quali *"Dawkins is a noted atheist"* e *"Dawkins is an outspoken atheist"*.
- Infobox: Dawkins è noto per la sua posizione di critica nei confronti della religione (*"criticism of religion"*).

Tuttavia, nonostante le differenti tecniche e approcci alla classificazione, molto spesso le *baseline* basate su *feature* semplici quali *n-gram* sono note per essere difficili da battere nei vari scenari legati alla *stance classification*, quali quelli descritti[144][60][98]. Pertanto, risulta ben chiaro come anche l'operazione di individuare *feature* atte a catturare i diversi aspetti sintattici e semantici all'interno del testo sia tanto complessa quanto lo scenario di interesse.

1.4.4 Importanza del *target* di classificazione

Un ulteriore fattore di grande importanza e distinzione all'interno di questo settore di ricerca, è dato dall'utilizzo o meno delle informazioni legate al *target* di riferimento. Un altro esempio è fornito da Ebrahimi et al.[40], i quali seguono un approccio di tipo probabilistico per la determinazione della *stance* nei *tweet*, modellando congiuntamente la stessa *stance*, la *stance* del *target* di classificazione e infine il *sentiment* del *tweet* in analisi. In particolar modo, la maggior parte delle precedenti metodologie

ha focalizzato la loro attenzione esclusivamente sull'estrazione di *feature* legate al testo preso in esame, ignorando per l'appunto il *target* di riferimento della *stance*. Come conseguenza, spesso i classificatori proposti presentano dati spuri, specialmente nel caso in cui il documento esprima un giudizio nei confronti di un altro *target*, diverso da quello oggetto della classificazione. In tale ottica, si aprono nuove strade legate alla *stance classification*, volte considerare in primo piano il *target* della classificazione. Una soluzione diretta a quanto osservato è proposta da Du et al.[37], i quali introducono un modello neuronale in grado di cogliere anche le informazioni date dal *target* di riferimento. Inoltre, sempre sulla scia di tale ragionamento, si considerano scenari in cui vengono meno alcune importanti ipotesi di base legate alla *stance classification*, come ad esempio l'indipendenza tra i *target* della classificazione. Più precisamente, i modelli attuali relativi alla *stance classification* si basano fortemente sulla precedente ipotesi. Tuttavia, in molti scenari, esiste una dipendenza naturale che lega tra loro i *target* di interesse. Un esempio di facile intuizione è dato dalla determinazione della *stance* nei confronti di due o più rappresentati politici nell'ambito di un'elezione, oppure nel rispetto di più marche di uno stesso prodotto. In tale senso, si ripone la propria attenzione sul problema di determinare la *stance* su più *target*, denominato formalmente come *multi-target stance detection*[68].

1.4.5 Il paradigma di apprendimento

Infine, anche il paradigma di apprendimento proprio della classificazione rappresenta un punto di diversità e distinzione all'interno dell'ambito di studio della *stance classification*. In particolar modo, il paradigma di apprendimento supervisionato è largamente utilizzato nella prevalenza dei lavori di ricerca effettuati, secondo il quale un grande ammontare di dati viene etichettato e arricchito da annotazioni, in modo tale da poter essere poi successivamente utilizzato come input per la fase di apprendimento del classificatore. Tuttavia, in quanto processo lungo e costoso, in alcuni scenari quali quello dei *social media* e *microblog*, si possono sfruttare informazioni specifiche del testo, come ad esempio gli *hashtag* nel caso di Twitter, o dati aggiuntivi di contorno per poter semplificare e di conseguenza velocizzare il processo di costruzione del set di apprendimento. In tali situazioni si parla generalmente di apprendimento debolmente super-

visionato (*weakly supervised*) o di *distant supervision*, il quale consente comunque al classificatore di raggiungere performance soddisfacenti in accordo con le difficoltà del contesto preso in esame. In altre parole, l'aggiunta di ulteriori dati, ottenuti mediante tecniche di selezione supervisionate come quelle indicate, e anche l'utilizzo di tecniche quale i *word embedding*, contribuisce a migliorare la *stance classification*[99].

1.5 Risultati scientifici

Terminate le digressioni generali su aspetti quali ambiti, applicazioni e metodologie di classificazione, vogliamo riassumere quanto osservato in maniera schematica, focalizzando la propria attenzione su aspetti tecnici quali *feature* utilizzate, algoritmi di classificazione e metriche di giudizio. In particolar modo, viste le innumerevoli varianti nell'ambito della *stance classification*, interessanti diversi livelli applicativi, risulta necessario mantenere tale suddivisione al fine di poter comparare correttamente informazioni quali le performance di classificazione, ottenute in accordo con le metriche di riferimento scelte. Inoltre, come strumento d'ausilio, riportato nella tabella 1.2, elenchiamo i corpora che hanno maggiormente interessato la ricerca, in modo tale da poter descrivere accuratamente le tipologie e la struttura dei dati presi in oggetto nei vari ambiti esplorati. In particolar modo, a ciascun corpus viene associato l'elenco dei lavori di ricerca che ha usufruito dei suoi dati, così da creare una mappa di interesse (figura 1.1) per la *stance classification* atta a valutare gli ambiti di maggior prevalenza. Sulla base dei precedenti strumenti di riferimento, è possibile osservare nella tabella 1.3 come informazioni generali quali *feature* linguistiche, lessicali e quantitative siano frequentemente utilizzate e sperimentate. Inoltre, non mancano tecniche di ausilio quale la *sentiment analysis*, in quanto, come osservato nella sezione 1.4, assume un ruolo di merito nell'ambito della *stance classification*, contribuendo positivamente nella fase di classificazione. Per quanto riguarda gli algoritmi e le tecniche a livello implementativo, si ricorre spesso a classificatori quali *Support Vector Machine* (**SVM**), *Naive Bayes* e *Logistic Regression*. Rimanendo in tale ottica, nonostante abbiano avuto una voce in capitolo solamente nel contesto dei *social media* e *microblog*, le reti neurali rappresentano comunque uno strumento di notevole interesse e sul quale verte

una particolare attenzione. Infine, in termini quantificativi delle prestazioni dei classificatori, metriche quali *accuracy* e *f1-score* rappresentano i principali strumenti di giudizio, in quanto consentono nella prevalenza dei casi di riassumere correttamente il comportamento dei classificatori impiegati. Concludendo, tutte le precedenti osservazioni, giocano un ruolo importante e di interesse nel presente elaborato, in quanto si collocano come punto di partenza per la prima di fase del progetto delineato come tesi.

Tabella 1.2: Elenco dei corpora su i quali sono stati effettuati dei lavori di ricerca nell'ambito della *stance classification*. In particolar modo, per ognuno di essi sono riportate le informazioni relative al contenuto e i lavori di ricerca in tale ambito che lo hanno interessato. Tra i vari data-set riportati, si può osservare dal (*) come il corpus *ArguAna TripAdvisor Corpus* sia l'unico a presentare anche dati non soggetti ad alcuna annotazione.

Corpus	Dimensione	Lavori
Emergent	300 rumoured claims, 2,595 articoli di giornale, 8.65 articoli di media per claim (min: 1, max: 50)	[46]
Internet Argument Corpus (IAC) v1	11,800 discussioni/threads, 3,317 autori, 390,704 posts, ∼73,000,000 parole.	[146], [8], [145]
Internet Argument Corpus (IAC) v2	414K posts (4forums), 65k posts (ConvinceMe), 3k posts (CreateDebate)	[160], [159],
ArguAna TripAdvisor Corpus*	2100 recensioni di hotel annotate, 200K recensioni di hotel non annotate	[129]
WC-ACL-2016 (IBM)	4603 categorie o liste di Wikipedia, 132 concetti	[153]
CreateDebate custom dataset	4902 post, 4902 metadati relativi a ciascun post	[60], [59], [145], [61]

Continua nella pagina seguente

Tabella 1.2 – *Continuazione della pagina precedente.*

Corpus	Dimensione	Lavori
The SemEval-2016 Stance dataset	2,914 tweet di training etichettate, 249 tweet di test	[99], [98], [93], [41], [37], [58], [40], [156], [7]
Multi-Target Stance dataset	4,455 tweets annotati manualmente	[68]
Brexit Blog Corpus (BBC)	1,682 frasi, 35,492 parole, 169,762 caratteri	[138]
Twitter-Ideology dataset	1,442,468 tweets di 645 membri ufficiali del congresso U.S.	[8]
International Corpus of Learner English (ICLE)	1319 saggi (*essay*)	[45]
Polititweets	34273 tweet di 7 politici francesi appartenenti a 6 gruppi politici	[36]
CNN corpus	1063 commenti di utenti	[140]
e-METI Idea Box corpus	936 proposte con commenti annessi	[106]
Environmental Protection Agency (EPA)	119 documenti, 240 frasi contenenti claim	[76]
Usenet corpus	13642 post con argomento l'aborto, 12029 post con argomento il controllo delle armi, 10285 post sull'immigrazione	[3]
ConVote corpus	3857 interventi organizzati in 53 dibattiti congressuali	[23], [151]
Ideological Debate corpus	2232 post sul porto d'armi, diritti per gli omosessuali, aborto e creazionismo	[144]
Twitter Debate corpus	543,404 tweet di 116,033 utenti contenenti 246,454 retweet	[128]

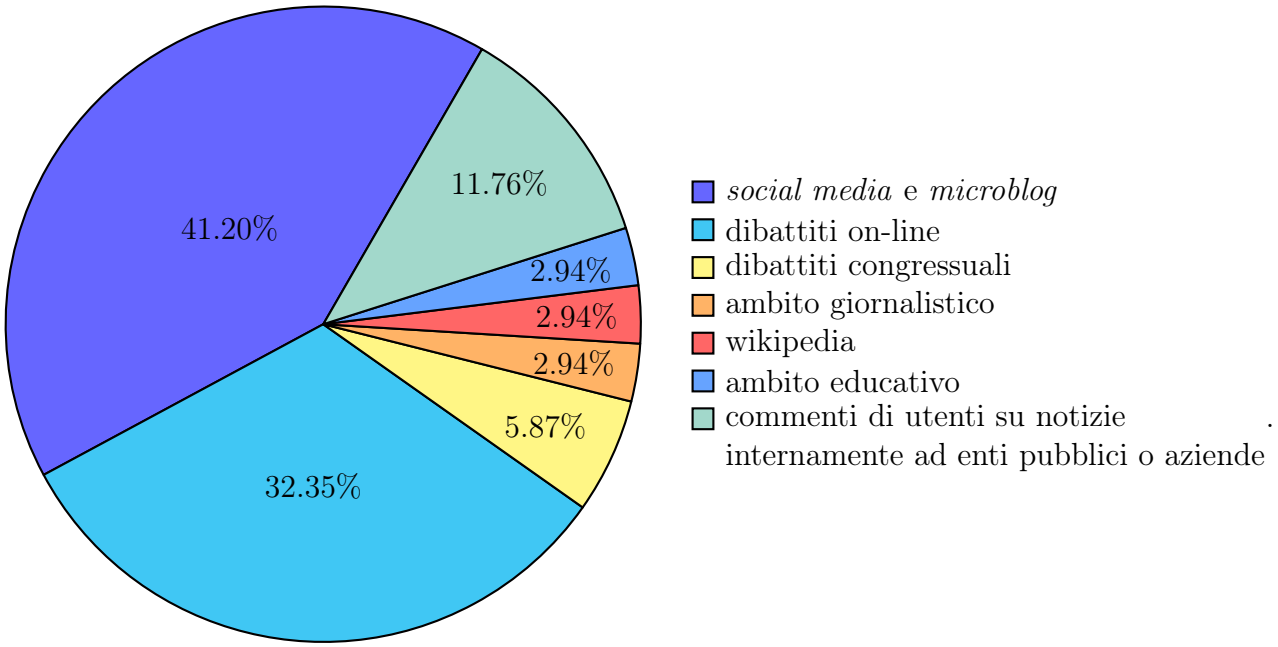


Figura 1.1: Distribuzione dei lavori di ricerca nell'ambito della *stance classification* ottenuta considerato come criterio di suddivisione gli scenari applicativi introdotti nella sezione 1.2.

Tabella 1.3: Elenco delle principali *feature*, tecniche e metriche impiegate nella *stance classification*, raggruppate in base agli ambiti di applicazione.

Ambito	Feature	Tecniche	Metriche
Social media e microblog	**Feature linguistiche**: Part-Of-Speech tag (POS), Bag of Words (BoW), word/character n-gram, word/character skip n-gram. **Regole linguistiche**. **Feature quantitative**: hashtag, parole, caratteri, frasi. emoji, punteggiatura, parole allungate, onomatopee. slang **Sentiment**: uso di lexicon esterni: NRC Emotion Lexicon, Hu and Liu Lexicon, MPQA Subjectivity Lexicon, NRC Emoticon, NRC Hashtag Lexicon, analizzatori: VADER, feature estratte mediante lo strumento Linguistic Inquiry Word Count (LIWC)[120] **Feature di contesto**: presenza o meno del target di interesse del tweet, presenza o meno di emoji, hashtag, parole allungate e simboli di punteggiatura. **Word Embedding**: similarità coseno **Feature lessicali**: caratteri, parole, simboli di punteggiatura, POS tag	**Classificatori**: SVM, BayesNet, NaiveBayes Multinomial, J48, HL-MRF, SMO, Random Forest (RF), Simple Logistics, Logistic Regression, AdaBoost, MLP, Alberi decisionali, Meta-classificatori, ZeroR, ReLP framework, RNN: basate su LSTM, specifiche quale TAN CNN: a livello di parole e di caratteri **Tecniche per feature**: Distant supervision, Word embedding, Filtraggio n-gram mediante Pointwise Mutual Information (PMI). **Tecniche classificazione**: Filtraggio feature non rilevanti mediante: Chi-Square, Info Gain e CFS	F1-score medio, F1-score macro, F1-score micro, Accuracy, Precision, Recall

Continua nella pagina seguente

Tabella 1.3 – *Continuazione della pagina precedente.*

Ambito	Feature	Tecniche	Metriche
Dibattiti on-line	**Feature linguistiche**: n-gram, BoW, verbi modali, regole sintattiche **Feature di contesto**: feature estratte dal post padre. Feature quantitative: parole, frasi, caratteri, documento, parole lunghe (6+ caratteri), pronomi, punteggiatura, discourse cue, vincoli tra autori (IC) **Sentiment**: categorie lessicali LIWC, dipendenze LIWC, negazioni, uso di lexicon quale MPQA[165] **Dipendenze sintattiche**: POS generalized, Opinion generalized **Frame semantici**: estratti mediante FrameNet[11] e SEMAFOR[33] **Informazioni relazionali**: autore, post	**Classificatori**: Naive Bayes con add-one smoothing, SVM, HMMs con Laplacian smoothing, CRFs, SVM (L2-regularized), PSL, JRIP, Maximum Cut, Minimum Cut, HL-MRF, Alberi decisionali, Logistic Regression, Maximum Entropy (MaxEnt) **Tecniche per feature**: Distant supervision **Tecniche classificazione**: Post-processing mediante l'uso di vincoli tra autori	Accuracy, Precision, Recall, F1-score, AUROC, AUC-PR
Dibattiti congressuali	**Feature linguistiche**: BoW **Feature di contesto**: vincoli su autori, conteggio citazioni, presenza o meno di citazioni, finestre di contesto	**Classificatori**: SVM, Minimum Cut, MRF, Iteravice classifier	Accuracy

Continua nella pagina seguente

Tabella 1.3 – *Continuazione della pagina precedente.*

Ambito	Feature	Tecniche	Metriche
Ambito giornalistico	**Feature linguistiche**: BoW, feature booleana se la frase termina con '?', distanze sintattiche (RootDist), negazioni tra coppie di parole triple subject-verb-object (SVO) **Word embedding**: similarità coseno	**Classificatori**: Logistic Regression (L1-regularized) **Tecniche per feature**: uso del Paraphrase Database (PPDB)[117] e dell'algoritmo di Kuhn-Munkres[74][105]	Accuracy, Precision, Recall
Ambito educativo	**Sentiment**: uso di lexicon esterni quale MPQA[165] **Feature lessicali**: POS tag **Informazioni relazionali**: metrica di similarità basata su Wikipedia (WLM)	**Classificatori**: Naive Bayes, SVM	Accuracy, Precision, Recall, F1-score
Wikipedia	**Feature sintattiche**. **Feature posizionali**. **Feature di contesto**: feature booleana se la frase è un titolo di Wikipedia. **Informazioni relazionali**: relazione di dipendenza tra claim, somiglianza morfologica tra i target dei rispettivi claim, somiglianza tra i percorsi ottenuti mediante WordNet[79]. **Word Embedding**: similarità coseno.	**Classificatori**: Logistic Regression (L2-regularized) **Tecniche per feature**: Modello per la predizione della stance di un claim C rispetto ad un topic T che tiene conto del sentiment del claim e del topic e della relazione di contrasto tra i loro rispettivi target Xc e Xt.	Accuracy rispetto ai tassi di copertura. (coverage)

Continua nella pagina seguente

Tabella 1.3 – *Continuazione della pagina precedente.*

Ambito	Feature	Tecniche	Metriche
commenti di utenti su notizie internamente ad enti pubblici o aziende	**Feature linguistiche**: BoW, n-gram, predicato principale **Feature lessicali**: POS tag POS bi-gram, coppie aggettivo-nome **Feature di contesto**: coefficiente di reazione **Sentiment**: parole e frasi positive e negative tramite lo strumento General Inquirer, utilizzo di opportuni analizzatori: SentiWordNet[9] **Frame semantici**: Estratti mediante FrameNet[11]	**Classificatori**: SVM, Maximum Cut, BoosTexter, Logistic Regression, Multinomial Naive Bayes	Accuracy, Precision, Recall, F1-score

Capitolo 2

L'Argumentation Mining (AM)

Insieme alla *stance classification*, recentemente, ha acquisito una posizione di notevole interesse nell'ambito della ricerca scientifica la cosiddetta *argumentation mining*, in settori quali l'analisi e l'elaborazione del linguaggio naturale (**NLP**). In particolare, la motivazione principale dietro ad una simile crescita d'attenzione risiede nel suo potenziale espressivo e nei suoi fini applicativi. Nello specifico, l'*argumentation mining* si esprime sotto forma di vie innovative di interpretazione ed elaborazione delle informazioni, con particolare riferimento a strumenti quali i *social media*. In aggiunta, un'ulteriore spiegazione trova le sue fondamenta negli sviluppi in aree di conoscenza affini quale il *machine learning*, le scienze sociali ed economiche, l'elaborazione di nuove politiche e le tecnologie di informazione. Pertanto, visto anche il netto coinvolgimento degli argomenti trattati dal settore di ricerca in questione all'interno del presente elaborato, risulta appropriato analizzare nel dettaglio il concetto di *argumentation mining*, soffermandosi inizialmente sui principi fondanti per poi passare successivamente alla descrizione degli ambiti di applicazione e ai vari modelli di argomentazione presenti in letteratura, con particolare riferimento alle tecniche e ai corpora esistenti. A questo proposito, è opportuno sottolineare che la struttura del presente capitolo si basa prevalentemente sul lavoro di Lippi e Torroni[83], in quanto quest'ultimo riassume in maniera chiara e dettagliata lo stato dell'arte dell'*argumentation mining*.

2.1 Introduzione al concetto

Come si può facilmente intuire dal nome, l'*argumentation mining* verte il proprio interesse sulla dimensione cognitiva ed espressiva propria di documenti contenenti ragionamenti e opinioni, atti a concretizzare le capacità razionali e di giudizio degli esseri umani. Pertanto, occorre dapprima soffermarsi sul significato stesso di argomentazione, individuandone le origini e gli ambiti di influenza. Procedendo con ordine, l'argomentazione, ovvero in sintesi il processo di costruzione degli argomenti, ha radici antiche in scienze quali la dialettica e la filosofia, dato che queste ultime sono incentrate sull'analisi e sullo studio dei processi che danno origine ai pensieri e alle affermazioni, con particolare attenzione alle modalità di risoluzione dei conflitti tra opinioni contrastanti[16]. Pertanto, nel corso dei secoli, l'argomentazione ha influenzato diverse aree di conoscenza, rappresentando di conseguenza un campo di ricerca interessante molteplici discipline, quale la filosofia, le scienze umane, la retorica, la psicologia ma anche l'informatica. In particolare, il concetto di argomentazione ha assunto un ruolo sempre più preponderante e di ruolo centrale nell'ambito dell'intelligenza artificiale[16], per via della natura degli obiettivi e della visione di fondo di quest'ultima, atta a volere coniugare la dimensione computazionale con i modelli cognitivi propri dell'essere umano. Nello specifico, lo studio dell'argomentazione nel campo dell'intelligenza artificiale, con particolare riferimento ad obiettivi come la rappresentazione della conoscenza e la ricerca di sistemi multi-agente, ha dato vita ad una nuova disciplina denominata *computational argumentation*[121][139][38]. D'altro canto, il concetto di argomentazione ha suscitato altrettanta attenzione in ambiti come le scienze cognitive, dove recenti studi sembrano indicare che le funzioni regolanti il ragionamento umano siano riconducibili ad un modello di tipos argomentativo[89]. Ancora, nell'ambito delle scienze sociali computazionali, i principi fondamentali che regolano i modelli di simulazione basati su agenti, recentemente introdotti, fanno riferimento in maniera esplicita alle teorie relative al concetto di argomentazione[86][48]. Nel presente contesto, un'importante fonte di informazioni nella prevalenza delle discipline interessate è rappresentata dal web, con particolare accezione per i *social media*. Più precisamente, risorse quali articoli di giornale online, recensioni di prodotti, *blog* e altro ancora, forniscono un flusso di dati eterogeneo e soggetto a continue

mutazioni, dove gli argomenti espressi dagli utenti possono essere individuati, isolati e infine analizzati. Tuttavia, l'attenzione verte anche su contesti specifici come il contesto legale, le recensioni online e i siti di dibattito[113][133][25]. Infine, accanto ai vari esempi applicativi riportati, si inseriscono anche progetti di grandi aziende, come ***IBM*** *Research* che ha finanziato un progetto di computazione cognitiva milionario, atto a sottolineare che l'interesse per il settore di ricerca in questione non è riconducibile solamente ad un semplice fattore di sfida in termini di successo scientifico, bensì rappresenta in un'ottica generale uno strumento applicativo dal grande potenziale. Pertanto, riassumendo, la disponibilità di informazioni di tale genere, congiuntamente ai rapidi e continui progressi nell'ambito di settori come il **NLP** e il *machine learning*, hanno definito le fondamenta per l'introduzione di una nuova area di ricerca denominata *argumentation mining* o *argument mining* (**AM**). Quest'ultima duplice nomenclatura è indice del significativo fattore di esplorazione concettuale ancora presente, proprio di questo nuovo settore di ricerca.

2.1.1 Definizione

Poiché l'*argumentation mining* è un dominio di ricerca giovane, sia le sue definizioni che gli approcci e i *target* di studio variano ampiamente in base al contesto di interesse. Ad esempio, alcune ricerche mirano ad estrarre gli argomenti da documenti generici non strutturati, attività costituente un passo fondamentale nell'ottica di applicazioni di pratico utilizzo[79]. Viceversa, altre partono già da un set di argomenti e focalizzano la propria attenzione su aspetti quale l'identificazione di relazioni di attacco o di supporto tra argomenti[26][21]. In ogni caso, generalmente, l'obiettivo principale dell'*argumentation mining* consiste nell'estrarre in maniera automatica gli argomenti a partire da documenti testuali generici e non strutturati, in modo tale da fornire dati organizzati per i modelli computazionali in grado di catturare, per mezzo di particolari rappresentazioni, concetti quali il ragionamento e gli argomenti, espressi in forma testuale all'interno dei testi presi in esame. Un esempio concreto di quanto detto è osservabile nella figura 2.1, dove si assiste all'estrazione automatica di argomenti a partire da semplici documenti testuali mediante l'impiego di un opportuno sistema di *argumentation mining*. Più precisamente, dapprima si procede con l'individuazione delle frasi contenenti potenziali

argomenti o parti di essi, i.e. frasi argomentative: nell'esempio sono evidenziati con i termini *claim* ed *evidence* (figura 2.1(a)), i quali verranno descritti e analizzati in dettaglio nelle sezione successive di questo capitolo. Proseguendo, si predicono gli eventuali collegamenti tra gli argomenti estratti (figura 2.1(b)) e infine si stimano i legami tra i singoli componenti all'interno di ciascun argomento (figura 2.1(c)), in modo tale da ottenere come risultato finale una struttura relazionale esplorabile facilmente, i.e. un grafo.

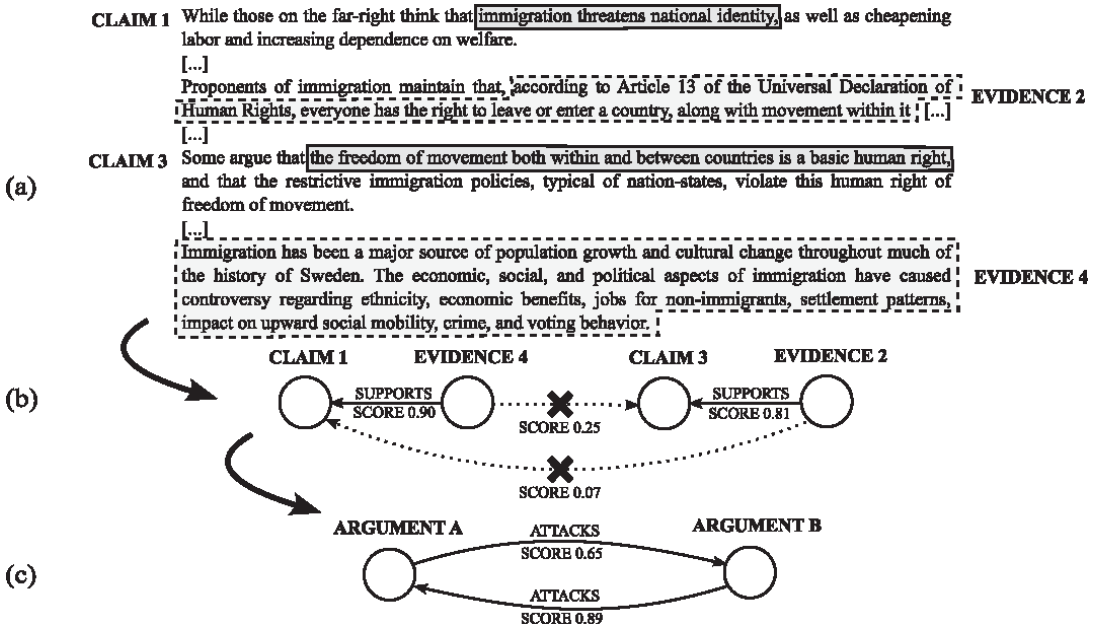


Figura 2.1: Esempio di estrazione di argomenti a partire da documenti testuali. La presente immagine è cortesia del lavoro di Lippi e Torroni[83].

2.1.2 Potenziale espressivo e ambiti applicativi

Una volta compresi i principi fondanti dietro la recentemente introdotta disciplina dell'*argumentation mining*, prima di soffermarsi nel dettaglio sulla sua struttura interna e sui punti di ricerca aperti, è opportuno presentare un'idea dell'importanza delle tematiche da essa affrontate in ambiti di applicazione reale, in modo tale da poter evidenziare nel dettaglio il potenziale applicativo del presente settore di ricerca. Più precisamente,

lo sviluppo di potenti sistemi di *argumentation mining* apre la strada verso nuove e innovative applicazioni in molteplici discipline, quali le scienze sociali e umanistiche, così come l'ingegneria, l'informatica e l'intelligenza artificiale. Come esempi principali dei possibili miglioramenti, che l'introduzione di strumenti di *argumentation mining* potrebbe apportare a diversi ambiti di studio e di ricerca in scenari reali, consideriamo l'apprendimento di politiche o la formulazione di decisioni[92], dove l'utilizzo di argomenti estratti automaticamente consentirebbe il miglioramento dei modelli e delle scelte strategiche di ausilio. Ancora, l'*argumentation mining* potrebbe rappresentare un bene aggiuntivo all'interno di processi di ingegnerizzazione aventi come obiettivo la valutazione automatica di soluzioni di design alternative[15]. Proseguendo, l'analisi dei mercati e il *profiling* dei clienti potrebbero ottenere significativi benefici dall'analisi degli argomenti fornita dagli utenti del web. Sempre nel presente scenario di interesse, un'ulteriore prospettiva di notevole coinvolgimento è data anche dall'analisi dei comportamenti e delle interazioni sociali, così come la dialettica e la retorica degli argomenti di interesse. Pertanto, l'*argumentation mining* potrebbe sbloccare nuove modalità innovative di organizzazione, supporto e visione dei dibatti online, mediante, ad esempio, il *clustering* dei post degli utenti e proponendo nuovi criteri di filtraggio basati sulla forza delle argomentazioni e sulla mutua posizione di accordo dei partiti coinvolti e i loro *post*. Quanto evidenziato, si propone in qualità di una nuova innovativa soluzione alle più adottate tecniche all'interno della comunità del *computational argumentation*, basate principalmente sul fattore collaborativo degli utenti[94][49]. Inoltre, ambiti di ricerca quale l'*opinion diffusion*[56] potrebbero certamente trarre beneficio dallo sviluppo di sistemi di *argumentation mining*, con l'accortezza di integrare il loro modello architetturale con strumenti proveniente da attività quale l'analisi dei *social network*[135], al fine di poter sfruttare informazioni quale la struttura della rete, i pattern di comunicazione, le informazioni relative agli utenti e le loro connessioni. Sempre in tale contesto, un altro recente elemento di novità nell'ambito dell'intelligenza artificiale, che potrebbe essere sfruttato dall'*argumentation mining*, è rappresentato dai sistemi di raccomandazione, dove sono stati effettuati diversi tentativi per cercare di combinare gli strumenti propri della *sentiment analysis* con sistemi di filtraggio collaborativi[166][71]. Infine, L'*argumentation mining* potrebbe anche fornire un miglioramen-

to sostanziale allo sviluppo di strumenti atti ad interpretare gli argomenti presenti nel web e agire in qualità di tecnologia di supporto per la *Argument Web vision*[19], la quale consente di definire una struttura di collegamento tra argomenti, in accordo con il formato **URI**, atta ad abilitare la navigazione tra molteplici risorse multimediali e siti web. Per quanto riguarda, invece, l'ambito prettamente economico e di business, considerando una prospettiva leggermente diversa, Hogenboom et al.[63], propongono un'analisi teorica delle potenziali applicazioni delle tecniche di argomentazione nell'ambito della *sentiment analysis* per documenti proprio del genere considerato. Un altro dominio di grande interesse, dove i sistemi di *argumentation mining* potrebbero essere di notevole ausilio, è dato dalla cosiddetta *intelligence analysis*, con particolare attenzione al contesto della comunicazione uomo-macchina e ai sistemi multi-agente[81]. Più precisamente, la capacità di comprensione della dimensione narrativa e la facoltà di poter formulare discorsi, come ad esempio lo *storytelling*, da parte di un computer rappresentano alcuni dei principali contesti di sfida in questo settore di ricerca, con particolare riferimento alla costruzione di servizi a scopo educativo. Infine, la definizione di macchine in grado di raggiungere un livello di comprensione e di ragionamento paragonabile a quello umano rappresenta da sempre una delle più grandi sfide dell'intelligenza artificiale, la quale, purtroppo, deve tener conto di fattori quali i costi di costruzione e mantenimento di conoscenza strutturata in domini aperti a partire da dati non organizzati. Tuttavia, come primo passo in quest'ultima direzione, è stato osservato come metodi di inferenza statistica nei sistemi *DeepQA*, come ad esempio ***IBM*** *Watson*, abbiamo già raggiunto livelli di complessità tali da poter competere e vincere contro degli esseri umani al gioco *Jeopardy*[44]. Pertanto, riassumendo, l'*argumentation mining* potrebbe quindi essere il prossimo settore di ricerca in grado contribuire in maniera significativa allo sviluppo scientifico in questa direzione. In particolare, potrebbe essere la chiave per una nuova generazione di sistemi di intelligenza artificiale, capaci di poter combinare *framework* statistici e di inferenza logica: ovvero in grado di estrarre argomenti a partire da testi non strutturati e scritti da esseri umani, per poter poi successivamente ragionare su di essi e produrre infine, secondo un procedimento logico, nuovi argomenti e quindi nuova conoscenza.

2.2 Modelli argomentativi

Visto il notevole fattore di influenza rappresentato dall'*argumentation mining* nell'ambito di molteplici discipline di vario genere, come conseguenza non dovrebbe stupire il fatto che la letteratura in tale settore di ricerca sia ricca di modelli di definizione dell'argomento. Tra i più importanti vi sono IBIS[75] e quelli introdotti da Freeman[47] e da Toulmin [154]. Quest'ultimo, in particolare, ha delineato una suddivisione della struttura logica del processo argomentativo e razionale dell'essere umano in sei diverse categorie.

- ***Claim***: affermazione soggettiva e spesso ambito di discussione.
- ***Datum***: concetto o affermazione inconfutabile, il quale forma la base per l'elaborazione e la definizione del *claim*.
- ***Warrant***: regola di inferenza atta a collegare il *datum* con il *claim*.
- ***Qualifier***: elemento d'ausilio avente la funzione di rappresentare il livello di confidenza nei confronti del *claim*.
- ***Rebuttal***: elemento che definisce le condizioni per delineare il *claim*.
- ***Backing***: elemento atto a giustificare il *warrant*.

Tuttavia, nonostante la grande influenza esercitata, dal punto di vista pratico e applicativo il modello di Toulmin è ancora soggetto a valutazioni e discussioni riguardanti una sua possibile rappresentazione in ambiti politici, giuridici e legali, umanistici, filosofici e propri del web[108][57].

2.2.1 Categorie di modello

Oltre al punto di scelta arbitrario, rappresentato dall'introduzione di un modello atto a delineare la struttura del concetto di argomento, è importante sottolineare la presenza di ulteriori distinzioni di maggiore granularità, aventi come punti di riflessione la discriminazione delle varie tematiche di interesse generale di ciascun modello di definizione o l'introduzione di uno o più criteri di argomentazione. Tra le principali distinzioni, un'importante suddivisione dei modelli di definizione dell'argomento prevede l'introduzione di tre principali categorie[17].

- ***Rethorical***: l'enfasi è incentrata sulle capacità persuasive dell'interlocutore di attirare l'attenzione di un pubblico di ascoltatori.
- ***Dialogical***: descrivono il modo in cui gli argomenti sono connessi tra di loro mediante strutture dialogiche per l'appunto.
- ***Monological***: l'attenzione verte sulla struttura dell'argomento in sè, includendo, ad esempio, i legami tra i differenti componenti di uno specifico argomento preso in considerazione.

Inoltre, un'altra ben nota classificazione nell'ambito della *computational argumentation* è data dalla dicotomia tra argomentazione astratta e strutturata. Procedendo con ordine, la prima considera ogni argomento come un'entità atomica, sprovvista, pertanto, di alcuna struttura interna[38]. In particolare, dal punto di vista del criterio di suddivisione introdotto da Bentahar, tale descrizione rappresenta quindi un modello dialogico, atto a focalizzare la propria attenzione sulle relazioni che sussistono tra i diversi argomenti (figura 2.1(c)), come ad esempio le relazioni di attacco o supporto, e sull'analisi di questi ultimi in accordo a specifiche semantiche. Viceversa, l'argomentazione strutturata propone una suddivisione interna dell'argomento in componenti. Ciò risulta di fondamentale importanza se si considera, ad esempio, come obiettivo l'estrazione delle parti costituenti gli argomenti a partire dall'analisi del linguaggio naturale (figura 2.1(a) e 2.1(b)). Come conseguenza, l'*argumentation mining* segue prevalentemente il modello di argomentazione strutturata. Infine, anche in questo caso è possibile ricollegarsi a quanto osservato da Bentahar: il modello proposto concorda principalmente con la definizione di categoria *monological*, ma può essere tuttavia ricondotto anche a quella *dialogical*[18]. Pertanto, visto il notevole interesse riscontrato nell'ambito della argomentazione strutturata, risulta molto difficile definire in maniera formale il concetto di argomento, nell'ottica dei componenti che lo costituiscono, tale da essere riconosciuto in maniera universale. Tra le tante, una definizione intuitiva di argomento è stata proposta da Walton[162], descrivendo quest'ultimo come un insieme di affermazioni costituito da tre parti: (1) una o più premesse, (2) la conclusione e (3) l'inferenza, spesso identificata come l'argomento stesso, atta a legare i primi due tra loro. In particolare, nella letteratura, le conclusioni sono spesso riferite con il termine *claim*, le premesse sono invece denomina-

te *evidence*, *reason* o *datum* nel caso del modello di Toulmin e infine il collegamento tra i due prende il nome di *warrant*.

2.3 Struttura dei sistemi di *argumentation mining*

La necessità primaria di partire dall'elaborazione di testi non strutturati per poter successivamente estrarre nozioni e informazioni atte a catturare la dimensione argomentativa dell'espressione umana, ha richiesto la definizione di sistemi di *argumentation mining*, atti a gestire singolarmente e in sequenza tutte le problematiche di contorno che condizionano il raggiungimento degli obiettivi propri di questo settore di ricerca. In particolar modo, l'architettura proposta è rappresentata da una *pipeline* (figura 2.2), all'interno della quale è possibile evidenziare tre fasi principali: *argumentative sentence detection*, *argument component boundary detection* e infine *argument structure prediction*, denominate rispettivamente in accordo con l'ordine definito dal presente modello in questione. Tuttavia, prima di poter soffermarsi nel dettaglio sulle presenti fasi di elaborazione del testo, è opportuno ragionare sui molteplici fattori che le interessano, al fine di evidenziare con cura gli aspetti di difficoltà ed eventuali punti aperti di studio. In particolare, le problematiche oggetto di interesse dell'*argumentation mining* possono essere riassunte secondo cinque dimensioni ortogonali:

- **Granularità dell'input**: indica il livello di dettaglio all'interno del quale gli argomenti e i loro componenti sono ricercati. In particolar modo, la prevalenza degli studi focalizza la propria attenzione sulle singole proposizioni, anche se alcuni autori suggeriscono una ricerca dei confini sintattici dei componenti ad un livello di dettaglio maggiore, come ad esempio l'*argumentative zoning*[150], dove si considerano porzioni di testo paragonabili a paragrafi.

- **Tipologia dell'input**: definisce le caratteristiche dei dati, come ad esempio l'ambito, il quale può essere di tipo giuridico, relativo a discussioni online, giornalistico, educativo, i.e. saggi brevi, e altro ancora, ciascuno distinto da peculiarità specifiche. Tale fattore è particolarmente importante in determinati scenari, dove risulta

necessario tenere conto di informazioni specifiche legate al contesto. Ad esempio Budzynska et al.[22] mostrano come l'*argumentation mining* nell'ambito dei dialoghi non possa essere studiato a livello sperimentale in maniera soddisfacente solamente mediante l'ausilio di modelli estranei al contesto in questione, i.e. *dialogue-agnostic model*.

- **Modello di definizione dell'argomento**: ogni sistema di *argumentation mining* fa riferimento ad uno specifico modello di definizione del concetto di argomento. Fino ad ora, la prevalenza dei sistemi implementati fa riferimento al modello di Walton, ovvero quello basato sui termini *claim* e *premise*.

- **Granularità del *target* di interesse**: il *target* del processo di ricerca varia anche in termini di granularità. In particolare, alcuni lavori considerano come riferimento i componenti dell'argomento, come ad esempio il *claim*, mentre in altri ancora il *target* coincide con l'intero argomento.

- **Obiettivo dell'analisi**: i principali obiettivi sono rappresentati dall'individuazione degli argomenti o dei loro componenti[162][79][131], la loro classificazione, la predizione di legami e relazioni tra argomenti o i loro componenti. Successivamente, sono presenti anche attività secondarie come l'attribuzione della paternità di un argomento, i.e. *attribution*, e il rilevamento di assunzioni e di componenti impliciti, detti entimemi, ricorrendo all'ausilio di supposizioni legate al buon senso, i.e. *completion*.

Sulla base della seguente suddivisione, si identificano le principali tematiche di interesse dell'*argumentation mining*, atte a seguire uno specifico modello di approccio, descrivente nel dettaglio alcuni o tutti i punti di interpretazione arbitraria precedentemente elencati.

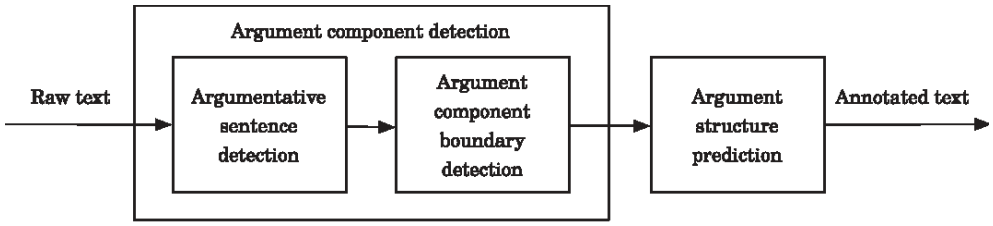


Figura 2.2: Architettura di un sistema di *argumentation mining*. La presente immagine è cortesia del lavoro di Lippi e Torroni[83].

2.3.1 Argument Component Detection

L'individuazione all'interno di documenti testuali degli argomento e/o dei loro componenti, in accordo con la specifica granularità del *target* selezionata, rappresenta uno dei principali ambiti di ricerca all'interno di un sistema di *argumentation mining* (figura 2.1(a)). In particolar modo, nella prevalenza delle soluzioni implementative proposte, il suddetto problema è spesso suddiviso in due sotto fasi, ovvero l'estrazione di frasi contenenti potenziali argomenti, i.e. frasi argomentative, e l'individuazione dei limiti sintattici dei rispettivi componenti. Formalmente si fa riferimento alle problematiche in questione con i termini *argumentative sentence detection* e *argument component boundary detection* rispettivamente. Tuttavia, è importante sottolineare che non tutti i sistemi adottano necessariamente il modello di suddivisione proposto. In particolar modo, Stab e Gurevych[148] e Eckle-Kohler et al.[42] assumono che i confini sintattici dei componenti siano già stati precedentemente individuati, così da ridurre i loro obiettivi alla singola attività di classificazione.

Argumentative Sentence Detection

Tipicamente, il primo passo nell'individuazione delle componenti di ciascun argomento consiste nel selezionare ed estrarre le frasi, riportate all'interno del documento testuale preso in esame, contenenti un argomento o anche solamente una parte di esso. Pertanto, il presente processo consente di definire queste ultime evidenziate come frasi argomentative. In particolare, il seguente problema può essere facilmente formulato come un'attività di classificazione, gestibile, in principio, da qualsiasi tipologia

di classificatore nell'ambito del *machine learning*. Tuttavia, nonostante la semplice formulazione, si è osservata la sperimentazione di molteplici soluzioni, condizionate dal particolare modello di definizione dell'argomento scelto e dall'obiettivo finale del sistema di *argumentation mining* preso in esame. In particolare, dal punto di vista della classificazione si distinguono tre diversi scenari:

- Un classificatore binario è allenato a distinguere le frasi argomentative dalle altre, lasciando la fase di identificazione dei singoli componenti, quali il *claim* e le *premise*, ad una fase successiva.
- Un classificatore multi-classe è allenato a discriminare tutte le componenti che costituiscono un argomento, definite dal particolare modello di interesse, così da poter individuare all'interno di una singola frase più di un singolo componente. Un'alternativa altrettanto valida consiste nel definire un classificatore in grado di poter assegnare più etichette ad una stessa entità, i.e. *multi-label*, così da poter attribuire una frase a più componenti.
- Si considera un insieme di classificatori binari, uno per ogni componente in accordo con il modello di definizione dell'argomento selezionato, in modo tale da poter attribuire separatamente più classi ad una stessa frase. Alternativamente si può considerare lo stesso modello *multi-label* descritto al punto precedente.

Argument Component Boundary Detection

L'obiettivo di questa seconda fase prevede la determinazione dei confini sintattici di ciascun componente di un argomento[148], procedimento noto anche come *argumentative discourse unit*[119] o *argument unit*[42]. Ad esempio, se si considera la figura 2.1, la porzione di testo etichettata come `EVIDENCE 4` spazia tra più proposizioni, mentre le altre evidenziate sono delimitate da una singola frase ciascuna. Più precisamente, il problema di segmentazione in considerazione ha come necessità la determinazione dei punti di inizio e di fine di ciascun componente per ogni frase predetta come argomentativa, dato che non è garantito che l'intera proposizione corrisponda esattamente ad un singolo componente di un argomento[57]. Dal punto di vista della granularità dell'input, occorre considerare tre possibili casi, non necessariamente mutuamente esclusivi:

- L'intera frase o una parte di essa coincide con un componente di un argomento.
- Due o più componenti possono essere presenti all'interno della stessa frase.
- Un componente di un argomento spazia tra più frasi.

A tale proposito, la maggior parte dei metodi esistenti assume solamente una delle tre possibilità elencate[113][79]. Pertanto, risulta ben evidente come il problema dell'individuazione dei confini sintattici dipenda fortemente dal modello di definizione dell'argomento scelto. A tal proposito, alcuni esempi sono riportati da Habernal et al.[57], dove, in accordo con il modello introdotto da Walton, in media un *claim* spazia su 1.1 frasi, mentre una *premise* su 2.2 frasi. Ancora, il corpus di IBM[4][79], considera i *claim* come porzioni di testo relativamente brevi, contenute sempre all'interno di una singola frase, mentre le *premise* possono spaziare su più paragrafi. D'altro canto, alcuni lavori ignorano completamente il problema in questione, come Mochales Palau e Moens[113], i quali associano i componenti degli argomenti a frammenti di frasi, i.e. *clause*, precedentemente estratti. Ancora, Stab e Gurevych[148] e Eckle-Kohler et al.[42] considerano frasi già segmentate in modo tale da poter focalizzare la propria attenzione sulla loro classificazione in accordo con quattro classi predefinite: *premise*, *claim*, *major claim* e *none*.

2.3.2 Argument Structure Prediction

La fase certamente più complessa di un sistema di *argumentation mining* si pone come obiettivo la predizione dei legami tra gli argomenti o i loro componenti, in accordo con la particolare granularità del target selezionata. Più precisamente, il problema in questione rappresenta certamente una vera e propria sfida, in quanto richiede la comprensione della natura delle relazioni tra gli argomenti individuati all'interno del testo, necessitando pertanto di rappresentazioni di alto livello e di una conoscenza dei processi cognitivi. Formalmente, il problema pone principalmente l'accento sull'azione di predizione rispetto a quella di individuazione, dato che il *target* non coincide con una semplice porzione di testo, bensì è rappresentato da una connessione tra più frammenti testuali, appartenenti al

documento di interesse. Il risultato finale ottenuto al termine di quest'ultima fase è dato da un grafo di connessioni tra gli argomenti individuati o i loro componenti. Più precisamente, i collegamenti in questione possono rappresentare diverse tipologie di legami, quali solitamente l'implicazione logica, il supporto o il conflitto. Nell'ambito dei *social media* e dei contenuti web, un simile grafo può rappresentare uno strumento di notevole interesse e ausilio per una moltitudine di applicazioni. Tra i principali si evidenziano l'analisi delle dinamiche all'interno di dibattiti online, la diffusione delle influenze sui *social network*, la valutazione della approvazione o meno di differenti argomenti e il ruolo di particolari utenti al loro interno, il rilevamento di comportamenti anomali e infine lo studio delle strategie di retorica più influenti ed efficaci. Pertanto, l'impatto in ambiti, quali le scienze sociali, la linguistica computazionale, l'argomentazione formale e l'analisi discorsiva, sarebbe notevolmente marcato e degno di nota. Ancor di più rispetto alle altre fasi dei sistemi di *argumentation mining*, la *argument structure prediction* è fortemente influenzata dallo specifico modello di definizione dell'argomento e dalla granularità del *target* di interesse. In particolare, considerando un modello semplice quale quello di Walton, il problema della predizione delle connessioni tra conclusione, i.e. *claim*, e le premesse che lo supportano, i.e. *premise*, è formulato in maniera diretta mediante un grafo bipartito, dove i nodi sono partizionati in due categorie (figura 2.1(b) e 2.1(c)). D'altro canto, modelli di gran lunga più complessi comportano un'attività di predizione dei legami più sofisticata. Un esempio di facile intuizione è dato se si considera il modello di Toulmin, dove i componenti di un argomento, ovvero *warrant*, *rebuttal*, *backing*, *qualifier* e altri ancora, devono essere tutti individuati. In aggiunta, occorre considerare come spesso gli argomenti siano presenti solamente in parte all'interno dei documenti testuali. Ad esempio, rimanendo sempre nell'ambito del modello di Toulmin, anche il *claim* può essere alle volte implicito[108]. Come conseguenza, tali argomenti non possono essere individuati nelle prime due fasi di un sistema di *argumentation mining* e devono pertanto essere inferiti a partire dal contesto. Tuttavia, una simile necessità richiede la definizione di opportuni modelli, per i quali solamente poco sviluppo è stato effettuato in merito fino ad ora[129].

2.4 Tecniche e *feature*

Riprendendo la suddivisione effettuata nella sezione 2.3, è possibile riepilogare ed evidenziare le principali tecniche e *feature* relative alla fase di classificazione, sperimentate al fine di risolvere le molteplici problematiche di interesse della *argumentation mining*.

2.4.1 Argumentative Sentence Detection

I sistemi esistenti hanno usato fino ad ora una grande varietà di classici algoritmi propri del *machine learning*, includendo **SVM**[113][115][148][42], *Logistic Regression*[79][131], *Naive Bayes*[113][20][115][42], *Maximum Entropy*[113], *Decision Tree* e infine *Random Forest*[148][42]. In particolare, tutti i classificatori elencati sono stati allenati secondo un paradigma di apprendimento supervisionato, facendo quindi sempre riferimento ad una collezione di esempi opportunamente etichettati, i.e. un corpus. Sebbene molti lavori di ricerca nella letteratura abbiano tentato di comparare alcuni di questi approcci di classificazione, non vi è ancora chiara evidenza di quale tra essi sia da preferire. Infatti, nella quasi totalità dei sistemi di *argumentation mining* esistenti, la gran parte delle energie è stato focalizzata sullo sviluppo di *feature* altamente informative e sofisticate, atte a migliorare in maniera significativa le performance di classificazione, piuttosto che costruire modelli e algoritmi specifici per il problema considerato. Pertanto, riassumendo, la prevalenza delle soluzioni fino ad ora proposte si affida a classificatori semplici e di largo utilizzo. Riprendendo il discorso sulla scelta delle *feature*, anche da questo punto di vista, la prevalenza delle soluzioni proposte tende a fare riferimento alle più classiche *feature* per la rappresentazione del testo. Tra tutte, spiccano le cosiddette *bag-of-words* (**BoW**). In aggiunta, il modello può essere ulteriormente esteso considerando gli *n-gram*, quali solitamente *bi-gram* e *tri-gram*. Tuttavia, nonostante la sua popolarità, il modello basato su **BoW** presenta due importanti limitazioni:

- L'ordine delle parole nella frase è ignorato o considerato solamente in parte nel caso di *bi-gram* e *tri-gram* o *n-gram* di più alto grado.
- La similarità semantica tra i termini non è presa in considerazione.

Pertanto, per superare le limitazioni sottolineate risulta necessario ricorrere all'impiego di *feature* più avanzate, basandosi su strumenti di ausilio quali ontologie, tesauri e database lessicali come *WordNet*[79]. In aggiunta, un'altra categoria di *feature* adottate si basa su informazioni di tipo grammaticale, mediante strumenti statistici come analizzatori strutturali e di dipendenze sintattiche ed estrattori delle cosiddette *part-of-speech*[87] (**POS**). Altre *feature* di frequente utilizzo includono informazioni relative alla punteggiatura, ai tempi verbali[113][148] e ai connettivi[42]. D'altro canto, è possibile delineare *feature* ancora più complesse mediante l'ausilio di strumenti di predizione esterni, come i punteggi di soggettività, i.e. *subjectivity scores*, strumenti di identificazione del *sentiment* o sistemi di riconoscimento delle entità, i.e. *entity recognition*[79][131]. Infine, vi è comunque spazio per l'utilizzo di *feature* specifiche per il determinato contesto di applicazione. Ad esempio, nell'ambito dei dibattiti online, Biran e Rambow[20] utilizzano *feature* estratte manualmente, grazie all'ausilio dello strumento ***RST*** *Treebank*[29], per un classificatore *Naive Bayes* avente come obiettivo l'individuazione delle premesse, i.e. *justification*, supportanti un dato *claim*. Un altro aspetto cruciale nella scelta dei classificatori atti ad individuare le frasi argomentative a partire dal testo, riguarda l'utilizzo o meno di informazioni di contesto. In particolare, molti approcci si basano fortemente sui dettagli specifici dello scenario di interesse. Ad esempio, nel lavoro di Palau e Moens[113] sui documenti legali, è possibile usufruire di informazioni aggiuntive quali parole chiave o specifici descrittori sintattici per favorire il rilevamento di pattern strutturali ricorrenti. Ancora, il sistema sviluppato da ***IBM*** *Research* presso Haifa, come parte del progetto *Debater*, è stato progettato per poter operare sull'ipotesi di conoscere a priori il *topic* di interesse, in modo tale da potersi basare su tali informazioni per agevolare l'individuazione dei componenti di ciascun argomento[79][131]. Formalmente si fa riferimento a quanto descritto con i termini *context-dependent claim detection* (**CDCD**) e *context-dependent evidence detection* (**CDED**). Tuttavia, sebbene in molti casi l'utilizzo di informazioni di contesto si sia dimostrato uno strumento estremamente potente per l'implementazione di *feature* accurate, è altrettanto vero che il loro utilizzo limiti in qualche modo la capacità di generalizzazione dei sistemi di *argumentation mining*. Come conseguenza, l'impiego di *feature* specifiche del contesto tendono a portare il sistema

in una situazione di *overfit*. Pertanto, uno degli obiettivi principali dal punto di vista implementativo nell'ambito dei sistemi di *argumentation mining* riguarda la possibilità di poter portare a termine con successo esperimenti che spaziano su corpora, generi e scenari applicativi di diversa natura. A tal fine, nel tentativo di poter risolvere le problematiche evidenziate precedentemente, Lippi e Torroni[82] hanno proposto un modello basato su **SVM** per poter individuare *claim* indipendentemente dal contesto, i.e. *context-independent claim detection* (**CICD**). Più precisamente, la soluzione proposta sfrutta *kernel* strutturati su alberi di analisi sintattica, i.e. *constituency parse tree*, con particolare attenzione per il modello proposto da Moschitti[102], per misurare il coefficiente di similarità tra proposizioni. Infatti, molto spesso, tali *costituency parse tree* sono in grado di catturare la struttura retorica della frase senza ricorrere all'ausilio di informazioni di contesto, fattore che nella maggior parte dei casi è un forte indicatore della presenza di una frase. Tuttavia, sebbene la definizione di sistemi di validità generale rappresenti una vera e propria sfida nell'ambito dell'*argumentation mining*, in alcuni casi, come nel contesto dei *social media*, risulta necessario fare affidamento ad informazioni di contesto. In particolar modo, la natura intrinseca dei dati forniti da strumenti quali *social network* e *microblog*, caratterizzati da dimensioni limitate, ricchi di espressioni gergali, parole onomatopee, errori grammaticali e altro ancora, porta alla necessità di dover definire delle metodologie di *machine learning* e delle rappresentazioni ben specifiche. In aggiunta, anche all'interno dello stesso ambito, gli approcci sviluppati ad esempio per i *microblog* possono differire con molta probabilità dalle tecniche dedicate ad altri generi, quali i forum, le recensioni di prodotti, i *blog* e i siti di comunicazione.

2.4.2 Argument Component Boundary Detection

Per quanto riguarda la *argument component boundary detection*, approcci di frequente utilizzo nell'ambito della segmentazione potrebbero essere sfruttati mediante l'impiego di classificatori relazionali e caratterizzati da output strutturati, atti a formalizzare con facilità l'obiettivo come un problema di *sequence labeling*[109], ovvero dove ad ogni parola, all'interno del testo preso in esame, viene associata una particolare classe. Inoltre, metodi quali *Conditional Random Field*, *Hidden Markov Models*

e ad altri simili, sono stati applicati con successo ad una varietà di problemi di questo genere, includendo, ad esempio, il riconoscimento delle cosiddette *named entity* nei *tweet*[132] o l'estrazione di informazioni dai dati propri dei *social media*[67]. Il vantaggio principale di usare queste tipologie di metodi risiede nella possibilità di applicare la classificazione di tipo collettiva su un insieme di esempi, dove ciascuna istanza non è classificata in maniera indipendente dalle altre e dove l'ordine sequenziale è preso in considerazione. Ad esempio, i lavori di Goudas et al.[53], Sardianos et al.[134] e Park et al.[116] rappresentano un primo tentativo in tale direzione, sfruttando metodi quale i *Conditional Random Fields* per segmentare i confini di ciascun componente di un argomento. In particolare, Sardianos et al.[134] impiega anche le reti neurali ricorrenti (**RNN**) per costruire le rappresentazioni dei dati di interesse.

2.4.3 Argument Structure Prediction

Gli attuali approcci per la argument structure prediction si basano su ipotesi molto semplificative. Ad esempio, nel corpus di Aharoni et al.[4], una delle assunzioni principali riguarda il fatto che ogni *evidence*, i.e. *premise*, sia sempre associata ad un dato *claim* e il legame tra questi ultimi sia esclusivamente di tipo supporto. Come conseguenza, è possibile utilizzare le informazioni del *claim* per poter predire correttamente la *evidence*. Un ulteriore esempio è fornito da Palau e Moens[113], i quali hanno proposto una soluzione al problema ricorrendo ad un'analisi basata su grammatiche libere da contesto, definite manualmente e in accordo con i tipici pattern retorici e strutturali propri dell'ambito giuridico, al fine di poter predire le relazioni tra i componenti di ciascun argomento. Pertanto, visto l'approccio specifico per il contesto di interesse, è improbabile che il modello ottenga gli stessi risultati in altri scenari applicativi. Ancora, Stab e Gurevych[148] propongono un metodo prettamente classico, ricorrendo all'uso di un classificatore **SVM** per predire i legami tra componenti in accordo con il modello di Walton[162], i.e. il modello *claim/premise*, basandosi sul lavoro di Freeman[47]. In particolare, ad esempio, la classe *major claim* è propria del contesto di interesse, individuata spesso ricorrendo all'uso di *feature* dedicate, come la posizione della frase all'interno del saggio. Come conseguenza, il lavoro descritto da Stab e Gurevych impiega l'uso di *feature* specifiche, le quali sfruttano

informazioni proprie del contesto applicativo, i.e. *background-knowledge*. Tuttavia, quest'ultimo lavoro risulta comunque un approccio di notevole interesse, dato che oltre alla fase di classificazione delle frasi, verte principalmente sul problema della *argument structure prediction*. Allo stesso modo, Biran e Rambow[20] applicano la medesima tecnica per l'individuazione delle premesse e per la determinazione dei legami tra queste ultime e i *claim* di interesse. Infine, Cabrio e Villata[24] adottano l'implicazione logica a livello testuale, i.e. *textual entailment*, con l'obiettivo di inferire relazioni di attacco o supporto tra due dati argomenti.

2.5 Elementi di distinzione e di affinità

In base alle conoscenze fino ad ora descritte nell'ambito dell'*argumentation mining*, si è potuto osservare come molte delle problematiche oggetto di interesse e delle metodologie associate presentino somiglianze e sottili differenze con altre branche proprie del **NLP**, quali la *sentiment analysis* e l'*opinion mining*. Dal punto di vista delle affinità, l'*argumentation mining* si avvicina molto a diverse attività di **NLP** quali, ad esempio, la discriminazione di frasi contenenti informazioni non affidabili, i.e. *hedge cue detection*[155], la predizione della soggettività, i.e. *subjectivity prediction*[43], l'identificazione di quesiti e interrogativi, i.e. *question classification*[168] e infine la *sentiment analysis*[114]. Ad esempio, come riportato in maniera riassuntiva dalla tabella 2.1, l'individuazione di frasi argomentative coincide fondamentalmente con l'attività di classificazione delle proposizioni[72]. Ancora, il problema della determinazione dei confini sintattici dei componenti di un argomento può essere modellato come un problema di etichettatura sequenziale, i.e. *sequence labeling problem*, e pertanto si collega ad attività quale il riconoscimento di entità, i.e. *entity recognition*[107], e altre applicazioni relative alla segmentazione del testo[30]. Infine, la *argument structure prediction* è affine ad attività di predizione dei legami e delle relazioni[52], ma presenta anche varie analogie in campi quale la classificazione delle relazioni, i.e. *relation classification*, l'analisi dei discorsi, i.e. *discourse analysis*[80], la valutazione della similarità tra testi a livello semantico[2] e infine le differenti applicazioni nell'ambito dell'implicazione logica a livello testuale, i.e. *textual entailment*[111]. Per quanto riguarda l'*opinion mining*, tut-

tavia, sebbene condivida con l'*argumentation mining* alcune analogie, come evidenziato da Habernal et al.[57], mentre l'obiettivo della prima consiste nel comprendere il pensiero delle persone nei confronti di una determinata entità, la seconda verte la propria attenzione sui motivi che si celano dietro tali ragionamenti, il che implica un'analisi delle cause e delle motivazioni piuttosto che basarsi solamente su informazioni quali opinioni e il *sentiment*. In particolare, l'elemento di distinzione di un argomento risiede nella sua struttura interna, la quale nell'ottica delle sue premesse è rappresentata dalla conclusione e dal processo di inferenza che collega questi ultimi tra di loro. Pertanto, la grande ambizione dell'*argumentation mining* è data nel cercare di effettuare un passo in avanti rispetto all'*opinion mining* e alla *sentiment analysis*, ovvero focalizzarsi sull'analisi di quei processi cognitivi che portano gli esseri umani ad accettare o rifiutare in maniera razionale una opinione, un argomento o una teoria.

AM	ML-NLP
Argumentative sentence detection	Sentence classification Hedge cue detection Sentiment analysis Question classification Subjectivity prediction
Argument component boundary detection	Sequence labeling Named entity recognition Text segmentation
Argument structure prediction	Link prediction Discourse relation classification Semantic textual similarity

Tabella 2.1: Corrispondenze tra *argumentation mining* (**AM**) e le molteplici attività riguardanti il *machine learning* e l'elaborazione del linguaggio naturale (**ML-NLP**). L'immagine è cortesia del lavoro di Lippi e Torroni[83].

2.6 Corpora

Qualsiasi approccio nell'ambito dell'*argumentation mining*, per mezzo di tecniche quale il *machine learning* e l'intelligenza artificiale, richiede con chiara evidenza una collezione di documenti annotati, i.e. corpus, per essere usata come set di apprendimento per qualsiasi tipologia di classificatore. Tuttavia, generalmente la costruzione di corpora annotati è un'attività complessa e che richiede molto tempo, in quanto coinvolge risorse quali squadre di esperti atte a definire annotazioni omogenee e consistenti. In particolar modo, quanto osservato è vero per il dominio dell'*argumentation mining*, dove attività quale l'identificazione dei componenti di un argomento, la determinazione dei loro confini sintattici e la modellazione dei legami che li collegano tra loro, rappresentano un campo di sfida anche per gli stessi esseri umani[112][57]. In aggiunta, sono stati costruiti differenti data-set per molteplici obiettivi ben specifici e pertanto risulta difficile che questi ultimi siano adatti per diverse tipologie di approcci o scenari di interesse. Un esempio atto a dimostrare quanto appena osservato è dato quando si considera la *argument structure prediction*. Più precisamente, si è assistito alla creazione di molteplici corpora, caratterizzati da annotazioni e aventi come unico obiettivo l'analisi delle relazioni tra argomenti o i loro componenti in accordo con la specifica granularità del *target* selezionata, come ad esempio l'individuazione di legami di tipo supporto o di attacco tra argomenti, la quale rappresenta una particolare variante della *argumentation mining*. Generalmente, i corpora in questione, quale ad esempio *AraucariaDB*, presentano contenuti di tipo argomentativo, aspetto che li rende poco adatti per altre attività generiche di *argumentation mining*. A causa dello specifico obiettivo per cui sono stati costruiti, tali corpora mancano molto spesso di parti non argomentative, le quali rappresentano un elemento necessario per la classificazione dato che assumono il ruolo di esempi negativi per la fase di apprendimento dei classificatori. In particolare modo, i presenti data-set sono stati definiti sull'ipotesi che la fase di *sentence detection* sia già stata effettuata e che quindi siano disponibili solamente le proposizioni ritenute argomentative. Ancora, il corpus ***NoDE*** *benchmark database*[27] segue la stessa direzione, ma si distingue in base alla granularità del *target* differente. In particolare, la presente collezione di dati contiene argomenti ottenuti da una grande varietà di

fonti, includendo *Debatepedia* e *ProCon*. Tuttavia anch'esso presenta lo stesso difetto di non includere esempi di tipo non argomentativo. Ancora, il lavoro di Boltuzic e Snajder[21] è pressoché simile: essi considerano un piccolo corpus di documenti relativi a commenti di utenti nell'ambito di due argomenti controversi e sviluppano un sistema di classificazione delle relazioni tra ciascun commento e l'argomento associato, etichettandole in cinque differenti classi: `strong attack`, `attack`, `strong support`, `support` e `none`. Analogamente, il corpus tedesco presentato da Kirschner et al.[73] è una collezione di pubblicazioni scientifiche annotazione con strutture argomentative: `supports`, `attacks`, `details` e `sequence`. Un ulteriore esempio della forte limitazione e specificità dei lavori di ricerca riguardanti l'*argumentation mining* è dato dalla netta suddivisione degli ambiti applicativi. In particolar modo, oltre a concentrare la propria attenzione su una o determinate fasi dell'architettura relativa ad un sistema di *argumentation mining*, quale la precedentemente citata *argument structure prediction*, si può osservare come i dati appartenenti ai diversi corpora definiti negli anni, appartengano generalmente ad un unico dominio applicativo. Più precisamente, tenendo anche a mente il discorso fatto nell'ambito dei fattori di influenza della *argumentation mining*, si delineano i seguenti principali scenari di interesse.

- **Dominio giuridico e legale**: il diritto è stato uno dei primi domini di applicazione dell'*argumentation mining* e uno di quelli di maggior successo, grazie ai lavori di Mochales Palau e Moens[113] sulla corte europea dei diritti dell'uomo (**ECHR**)[112] e sui data-set di *AraucariaDB* per l'estrazione di *claim* e delle rispettive *premise* di supporto a partire da una collezione di documenti legali strutturati. Ancora, recentemente è stato presentato il cosiddetto Vaccine/Injury Project (**V/IP**)[6], il cui scopo consiste nell'estrazione di argomenti a partire da un insieme di decisioni giudiziarie riguardanti le regolazioni sui vaccini. In particolare, i presenti lavori di ricerca rappresentano fino ad oggi uno dei pochi esempi di sistemi atti ad implementare una completa pipeline per l'*argumentation mining*, sebbene sia fortemente specializzata per un particolare genere.

- **Biologia e medicina**: Lo sviluppo di data-set annotati a partire da testi di biologia e medicina è indice di una nuova tendenza che sta acquisendo sempre maggiore attenzione. Infatti, quest'ultima

potrebbe rappresentare un passo estremamente importante nella costruzione di ontologie e basi conoscenza descriventi i collegamenti tra, ad esempio, i sintomi o i geni e le malattie associate, o anche in un ruolo d'ausilio nella prescrizione di medicine specifiche. In particolare, Houngbo e Mercer[64] hanno proposto un sistema di classificazione delle proposizioni presenti all'interno di testi di biomedicina attraverso la distinzione di quattro differenti categorie: `introduction`, `method`, `results` e `discussion`. Infine, Green[54] offre una descrizione di tipo qualitativo del processo di creazione di un corpus all'interno di questo ambito.

- **Scienze umane**: i saggi brevi di tipo retorico, filosofico e di raccomandazione costituiscono un altro campo di ricerca dell'*argumentation mining* di notevole interesse. Ad esempio, Lawrence et al.[77] vertono la loro attenzione sullo studio riguardante l'integrazione di annotazioni manuali e generate automaticamente all'interno di una collezione di trattati filosofici del diciannovesimo secolo. Ancora, un data-set molto specifico ma ben documentato è stato presentato da Stab e Gurevych[147] come una collezione di 90 saggi di raccomandazione. In particolare, gli argomenti sono di diversa natura e le annotazioni all'interno delle proposizioni riguardano le *premise*, i *claim* e un *major claim* per ciascun saggio breve. Tuttavia, per via della natura dei dati e delle linee guida di annotazione, solamente una piccola minoranza delle frasi all'interno del corpus è di tipo non argomentativo. Pertanto, essendo stato progettato in maniera specifica per l'analisi dei saggi di raccomandazione, quest'ultimo corpus potrebbe non essere il più appropriato come set di apprendimento per la classificazione nel caso in cui l'obiettivo di quest'ultima concerni la generalizzazione nei confronti di altri generi.

- **Contenuti web**: I social media e il web semantico offrono una moltitudine di documenti di diverso genere attraverso una varietà di fonti di informazione differenti. Al momento, le tecniche utilizzate per estrarre le informazioni di interesse da tali sorgenti di dati sono prevalentemente basate su analisi statistiche e di connessioni, così come avviene in ambiti quale l'*opinion mining*[114] e l'analisi dei *social network*[39]. Viceversa, un sistema di *argumentation*

mining potrebbe garantire l'analisi qualitativa di una moltitudine di commenti pubblicati sui *social media* e sui siti di comunicazione specializzati, fornendo quindi degli strumenti innovativi per i ricercatori in contesti sociali e politici e infine creando nuovi scenari per il marketing e il business. Sebbene vi siano stati solamente pochi tentativi in quest'ambito, con molta probabilità per via della eterogeneità dei contenuti e delle diversità gergali, tuttavia questa nuova tendenza apre la strada ad una varietà di nuovi e interessanti domini applicativi. In tale ottica, l'*argumentation mining* potrebbe diventare la chiave per lo sviluppo di tecnologia atta a creare nuova conoscenza a partire da una vastità di dati disorganizzati e non strutturati. Infatti, il più grande data-set di *argumentation mining* disponibile al momento è stato sviluppato da ***IBM** Research*[4][131], a partire da pagine di Wikipedia. In particolare, l'obiettivo di tale corpus consiste nell'estrarre *claim* e *premise*, denominate *evidence*, dipendenti dal contesto preso in esame, rilevanti per uno specifico *topic* di interesse. Inoltre, vi sono anche altri corpora basati sul contenuto generato da utenti. Ad esempio, il lavoro descritto da Goudas et al.[53] tenta di proporre una soluzione a diverse fasi della pipeline propria dell'*argumentation mining*, includendo la *sentence classification* e la *boundaries detection*, mentre Sardianos et al.[134] si concentrano solamente su quest'ultima fase. Ancora, il data-set in lingua giapponese introdotto da Reisert et al.[130] considera le *premise* estratte all'interno di *microblog*, mentre una collezione di micro-testi è stata definita da Peldszus[118]. Proseguendo, il corpus presentato da Eckle-Kohler et al.[42] consiste in notizie in lingua tedesca annotate secondo il modello di Walton[162]. Un altro corpus ben noto è stato sviluppato da Haberna et al. [2014] al fine di modellare gli argomenti in accordo con una variante del modello di Toulmin[154]. Più precisamente, quest'ultimo data-set include 990 commenti in lingua inglese relativi ad articoli o *post* all'interno di forum, 524 dei quali sono etichettati come argomentativi. Infine, nel contesto dei dibattiti online, Biran e Rambow[20] hanno annotato un corpus di 309 discussioni su *blog*, ottenute da risorse quale il *LiveJournal*, etichettando i *claim*, le *premise*, denominate *justification*, e i legami tra questi ultimi.

Infine, vi sono alcune aspetti propri dell'*argumentation mining*, che

fino ad ora sono stati considerati in maniera marginale, ma che potrebbero apportare un notevole contributo all'esplorazione e allo sviluppo del presente settore di ricerca.

2.6.1 I *Big Data*

Molte discipline sono al giorno d'oggi attratte dalla cosiddetta sfida dei *big data*, la quale consiste nel sfruttare l'immenso ammontare di informazioni e conoscenza disponibile sul web, per attività di diverso genere e scopo. Dal punto di vista dell'*argumentation mining*, ciò rappresenta una grande opportunità, in quanto una grande moltitudine di fonti di informazione, quali *social media* o il *semantic web*, può fornire dati di tipo argomentativo con diverse caratteristiche. Tuttavia, quando si ha a che fare con collezioni di dati di tali dimensioni, è necessario tenere conto di problematiche quale la scalabilità dei sistemi di *argumentation mining*, a differenza di quanto osservato fino ad ora dove la maggior parte degli studi di ricerca sono stati sperimentati su corpora di piccola dimensione. In aggiunta, il web potrebbe essere d'ausilio per risolvere un altra problematica chiave all'interno dell'*argumentation mining*, ovvero la limitata disponibilità di corpora opportunamente annotati. Infatti, un'interessante linea di ricerca potrebbe basarsi sullo sfruttamento di tecniche quale il *crowdsourcing* per annotare grandi quantitativi di dati. Ad esempio, in contesti quale la *image classification*[110] e la *object detection*[34], la straordinaria potenza computazionale del *crowdsourcing* ha permesso la costruzione di data-set annotati di larga scala. In particolar modo, uno dei più grandi corpus esistenti nell'ambito della *sentiment analysis* è stato costruito per mezzo di tale meccanismo di *crowsourcing*[141]. Tuttora, il *crowdsourcing* è in fase di studio e analisi per cercare di comprendere a pieno il suo vero potenziale, valutando in particolare possibili problemi di coerenza di annotazione dei dati. In generale, il potenziale rappresentato da tale meccanismo è certamente enorme, motivato dal numero crescente di piattaforme atte a supportarlo, come ad esempio *Amazon Mechanical Turk*. Nell'ambito dell'*argumentation mining*, una sfida aggiuntiva è rappresentata dalla qualità delle domande presentate agli utenti partecipanti, la quale può avere un impatto molto negativo in merito al raggiungimento di un quorum riguardo a determinati dati e che quindi rappresenta il motivo per cui i primi tentativi in questa direzione abbiano

avuto scarsi risultati[57]. In quest'ottica, mentre attività come l'*image tagging* sono di facile formulazione, altre, quale l'individuazione dei confini sintattici dei componenti di ciascun argomento, rappresentano sicuramente delle vere e proprie sfide. In particolar modo, la difficoltà risiede nel definire delle fasi di annotazione opportune e sensate per il contesto di studio di interesse, in modo tale da poter ottenere come risultato un accordo tra annotatori non esperti.

2.6.2 Gestione di dati non supervisionati

Quando si ha a che fare con data-set di grandi dimensioni, un problema cruciale consiste nell'adottare algoritmi di *machine learning* veloci e al tempo stesso efficienti. Per via delle difficoltà e dei costi riscontrati nel creare corpora ricchi di dati, tutti opportunamente etichettati, una alternativa al *crowdsourcing* è data dall'impiego di tecnologie di *machine learning* in grado di gestire i dati non strutturati in maniera non o semi supervisionata. In particolare, le tecniche proprie del *deep learning*, che hanno ottenuto risultati senza precedenti in molti ambiti dell'intelligenza artificiale, inclusa l'elaborazione del linguaggio naturale (**NLP**), rappresentano certamente una delle principali scelte a livello implementativo come soluzione delle presenti problematiche. Pertanto, l'abilità di gestire grandi quantitativi di dati non supervisionati rappresenta sicuramente uno degli aspetti principali delle cosiddette *deep network*, e una tra le motivazioni chiave dietro il loro successo. L'apprendimento non supervisionato è infatti adottato in molte architetture nell'ambito del *deep learning* avente come ruolo principale la definizione di una fase di *pre-training* il cui obiettivo consiste nell'estrarre insiemi gerarchici di *feature* direttamente dai dati. Recentemente, la prevalenza delle ricerche nell'ambito del *deep learning* si è concentrata su attività legate al linguaggio, producendo una grande varietà di sistemi sofisticati dedicati a specifiche applicazioni[78]. Molti di questi sistemi sono capaci di catturare informazioni a livello semantico avanzate a partire da testi non strutturati, e all'interno di questo contesto, il Web rappresenta una miniera di valore inestimabile per ottenere informazioni eterogenee e di molteplici sfaccettature. In tale direzione, un approccio di grande successo è dato dai cosiddetti *word embedding*, i.e. *word vector*, i quali consistono in spazi di *feature* appresi in maniera automatica in grado di codificare somiglianze

tra termini ricche e di alto livello[91]. Esempi di architetture nell'ambito del *deep learning*, capaci di gestire sequenze di input di lunghezza arbitraria e in grado di raggiungere lo stato dell'arte in ambiti specifici del linguaggio, quale la *sentiment analysis*, sono le cosiddette *Recurrent Neural Tensor Networks*[141], le *Tree-Structured Long Short-Term Memory Networks*[149] e le versioni specifiche per il linguaggio delle reti convoluzionali[72]. Perciò, tali architetture possono essere adottate per definire e risolvere alcune delle principali fasi dell'*argumentation mining*, come la *sentence classification*. Ad esempio, Sardianos et al.[134] fornisce un primo approccio all'utilizzo dei *word vector* proprio in tale ambito di interesse.

2.6.3 Dati strutturati e di tipo relazionale

Un altro importante limite della maggior parte degli approcci esistenti nell'ambito dell'*argumentation mining* è rappresentato dal modo in cui sono gestiti i dati strutturati e relazionali[95]. Nell'ultimo decennio, il *machine learning* ha attraversato la cosiddetta rivoluzione relazionale[51], al fine di estendere le proprie metodologie e algoritmi per gestire flussi di dati dal grande contenuto informativo, come alberi, grafi o sequenze. Approcci atti a fornire output strutturati, come i *Conditional Random Fields* o le *Structured Support Vector Machines* (**SSVM**), ad esempio, sono in grado di attuare tecniche quale la classificazione collettiva, ovvero le predizioni su un nuovo esempio mai visto prima possono essere prodotte collettivamente, prendendo in considerazione la struttura intrinseca dei dati, come le informazioni di tipo sequenziale o i legami in termini di connessioni. Un esempio di facile intuizione è dato dai paragrafi all'interno di un documento testuale, dalle proposizioni consecutive in un dialogo e dai collegamenti all'interno di un *social network*. Come approccio in accordo con tale direzione, i *Conditional Random Fields* sono stati applicati da Goudas et al.[53] e Sardianos et al.[134] nel tentativo di risolvere il problema della *argument component segmentation*. In aggiunta, per quanto riguarda il problema della predizione di legami tra *premise* e *claim*, o tra differenti argomenti, può essere facilmente modellato come un'attività di predizione dei legami all'interno di un grafo, dove i nodi rappresentano gli argomenti o i loro componenti. Più precisamente, nel presente scenario, un contributo significativo può essere

con molta probabilità dato dal cosiddetto *statistical relational learning*, un settore di ricerca il cui obiettivo consiste nel combinare la logica del primo ordine con tecniche di *machine learning* di tipo statistico, ovvero ricorrendo all'impiego di rappresentazioni simboliche e sub-simboliche. In particolare, mentre il *machine learning* di tipo statistico e i modelli grafici possono gestire incertezze a livello di dati, il potere espressivo della logica del primo ordine, invece, può essere sfruttato per modellare la *background-knowledge* di una dato dominio di interesse e per descrivere le relazioni tra le varie istanze dei dati. Tale tipologia di approccio è stata applicata con successo in una varietà di attività aventi molteplici affinità con l'*argumentation mining*. Ad esempio, la scoperta di legami in reti di ambito sociale e biologico rappresenta fondamentalmente un problema di predizione delle relazioni[52] e quindi assomiglia al problema di *argument structure prediction* quando si considera il grafo degli argomenti e i rispettivi collegamenti. Ancora, l'estrazione di informazioni e l'individuazione di entità all'interno di corpora testuali[32][122] condividono analogie con la *argumentative sentence classification*. Infine, le cosiddette *sequence tagging* e *sentence parsing*[123] possono offrire interessanti prospettive per la modellazione di dati testuali strutturati.

Capitolo 3

La *Stance Classification* come strumento per l'*Argumentation Mining*

Come osservato nei capitoli precedenti, alcune delle principali tematiche di ricerca attuali sono rappresentate dalla *stance classification* e dalla emergente *argumentation mining*. A tale proposito, il presente elaborato vuole inserirsi in entrambi i contesti, con la speranza di poter delineare un punto di contatto atto a svelare nuove possibilità applicative. Più precisamente, l'ambito di ricerca verte la propria attenzione sulla definizione di un *framework*, in grado di rispondere a molteplici questioni, quale, ad esempio, la sperimentazione di metodologie proprie della *stance classification* in attività ben specifiche della *argumentation mining*, o l'analisi nei medesimi ambiti delle informazioni qualitative espresse dalla *stance*, al fine di valutarne l'efficacia in qualità di *feature* aggiuntiva. Si delineano, quindi, i seguenti obiettivi.

- Effettuare un'analisi delle principali tecniche e metodologie di classificazione nell'ambito della *stance classification*. In particolar modo, solamente per mezzo di una piena comprensione dei vari approcci in merito, è possibile successivamente selezionare i più promettenti tra loro, al fine di poter sperimentarli anche in altri ambiti, quale, ad esempio, l'*argumentation mining*.

- Nell'ottica di un particolare campo dell'*argumentation mining*, i.e. l'*argument structure prediction*, si ricerca l'individuazione di un possibile parallelismo tra quest'ultimo e la *stance classification*. Più precisamente, mediante l'osservazione dei processi di elaborazione del testo, specifici di ciascuno dei due settori di ricerca, si vuole vedere se gli input e gli output della *stance classification* possano essere mappati in accordo con il particolare problema selezionato dell'*argument structure prediction*.
- Come naturale conseguenza dell'obiettivo descritto precedentemente, risulta di particolare interesse valutare le tecniche e le *feature* specifiche per la *stance classification* in un diverso ambito di studio.
- Infine, mediante un'attenta analisi dei risultati ottenuti, occorre discutere l'eventuale possibilità di sviluppo di tecniche ad hoc per il presente scenario di interesse.

Pertanto, tenendo conto della caratteristica sequenziale delle linee guida precedentemente elencate, occorre necessariamente affrontare un problema per volta, al termine di ciascuno dei quali sarà di notevole importanza effettuare delle opportune valutazioni in merito al suo corretto soddisfacimento e alle reali possibilità di completamento dei successivi. Procedendo con ordine, il presente capitolo si propone di risolvere il primo obiettivo descritto, con particolare attenzione a motivare con cura tutte le scelte implementative effettuate così da agevolarne la comprensione e da sottolineare i punti di difficoltà riscontrati. In particolare, il capitolo in questione è strutturato come segue. Inizialmente, il primo tema affrontato è rappresentato dalla delucidazione del processo di analisi delle *feature* impiegate nell'ambito della *stance classification*, selezionando in particolar modo quelle ritenute maggiormente adatte per il caso di studio in oggetto. Più precisamente, pedissequamente alla fase di selezione delle principali *feature*, si inserisce un'ulteriore fase di scrematura di queste ultime, individuandone, nello specifico, la miglior combinazione e di ciascuna la configurazione dei parametri più performante. Successivamente, si prosegue con la descrizione delle metodologie impiegate al fine di poter definire un classificatore consono per l'attività di *stance classification*, in merito allo specifico data-set di apprendimento e con particolare attenzione al raggiungimento di performance paragonabili con lo stato dell'arte del presente settore di ricerca. In seguito, sulla

base dei risultati di apprendimento del classificatore, si procede con la fase di predizione della *stance* sui dati contenuti nei corpora di interesse relativi alla *argument structure prediction*, con particolare attenzione a definire i *target* di riferimento per la *stance classification* per mezzo di un'analisi preliminare. Infine, si procede con un'ultimo esame dei dati costruiti, al fine di poter verificare la validità delle annotazioni aggiunte ed effettuare considerazioni sulle singole coppie in termini di coerenza della *stance* attribuita per lo stesso *target* di riferimento.

3.1 Obiettivo

Come primo passo nella direzione del traguardo imposto, occorre definire in dettaglio il contesto applicativo di interesse. In particolare, tra le molteplici fasi di elaborazione del testo appartenenti ad un sistema di *argumentation mining*, desta notevole interesse l'*argument structure prediction*, in quanto, come evidenziato ampiamente nella sezione 2.4.3, quest'ultima rappresenta un vero e proprio elemento di sfida, dove al momento è stato possibile ottenere dei risultati soddisfacenti solamente in scenari soggetti a forti ipotesi semplificative. Inoltre, sulla base di quest'ultimo fattore di notevole importanza, risulta preferibile selezionare dei corpora caratterizzati da un significativo quantitativo di dati e non soggetti a scenari fin troppo specifici, in modo tale da consentire la sperimentazione di tecniche e metodologie che non rientrano necessariamente all'interno dei contesti specifici propri dell'*argumentation mining*. Pertanto, come data-set di riferimento nell'ottica dei molteplici obiettivi elencati, sono stati selezionati quelli introdotti da ***IBM** Research*, ovvero **CE-ACL-14** e **CE-EMNLP-15**, in quanto rappresentano i corpora di più grandi dimensioni per l'*argument structure prediction* e non riguardano un ambito applicativo altamente specifico dato che sono stati formulati considerando informazioni tratte da Wikipedia, riguardanti argomenti, i.e. *topic*, di diversa natura. Una volta definiti il contesto ed i dati di riferimento, è possibile passare all'introduzione del primo obiettivo proposto. Più precisamente, vista la volontà di voler sperimentare le metodologie proprie della *stance classification* nell'ambito della *argument structure prediction*, è evidente come l'analisi effettuata nel primo settore di ricerca debba selezionare tecniche, *feature* e dati il più possi-

bile conformi al secondo contesto di interesse. In altre parole, poiché i corpora di riferimento sono quelli introdotti da ***IBM*** *Research* e quindi contenenti elementi quali *claim* ed *evidence*, i.e. *premise*, occorre valutare con attenzione quali scenari applicativi propri della *stance classification*, con le relative metodologie e *feature*, presentino elementi di similarità dal punto di vista della tipologia dei dati contenuti nei presenti data-set di interesse. In secondo luogo, come obiettivo secondario, ma non in termini di importanza, si ricerca la definizione di un opportuno classificatore, in grado di presentare risultati comparabili con lo stato dell'arte della *stance classification*, al fine di limitare il più possibile eventuali errori di classificazione dovuti a motivi prettamente implementativi e non per via del particolare contesto di studio. Più precisamente, si delinea una serie di sotto-obiettivi, descriventi ciascuno i singoli passi da compiere per poter infine soddisfare il primo degli obiettivi generali introdotti all'inizio del capitolo.

- **Ricerca di *feature* rilevanti**: è opportuno considerare la natura dei dati di interesse relativi all'ambito della *argument structure prediction*, in modo tale da poter selezionare le *feature*, le tecniche e infine i data-set disponibili per l'attività di *stance classification*.

- **Ottimizzazione**: costruire un classificatore performante e al tempo stesso efficiente, utilizzabile, pertanto, su data-set di diversa struttura e in grado di ottenere risultati soddisfacenti con riferimento alle metriche di classificazione usate.

- **Costruzione di nuovi dati aggiuntivi**: nello specifico ci si riferisce alla determinazione della *stance* per ciascuna coppia di *target* di interesse, così da arricchire ulteriormente i data-set di riferimento per l'*argument structure prediction*.

- **Analisi dei dati**: studio sia quantitativo che qualitativo nell'ambito delle coppie *evidence-claim* estratte e dell'omogeneità tra *stance* di *evidence* e *claim*, formanti una coppia, nei confronti dello stesso *target* all'interno del medesimo scenario di studio.

3.2 Analisi degli strumenti applicativi propri della *stance classification*

Come è stato descritto ampiamente all'interno della sezione 1.2, la *stance classification* è stata impiegata in molteplici scenari applicativi, interessando aree di notevole importanza quali i *social media*, i dibattiti online, i siti di comunicazione e infine contesti di tipo educativo o giornalistico. Inoltre, ognuna delle precedenti tematiche di interesse è caratterizzata da fattori ben specifici, volti a catturare le dinamiche particolari che contribuiscono a distinguerle in maniera ben chiara. Pertanto, tenendo in considerazione il contesto di origine dei dati presenti nei due corpora oggetto di studio, i.e. **CE-ACL-14** e **CE-EMNLP-15**, è possibile formulare alcune importanti considerazioni riguardanti la tipologia dei dati di ciascun corpus. Dapprima, per via della presenza di elementi di distacco prettamente legati a fattori linguistici e di forma, riteniamo di poter escludere ambiti quale i *social media* e i *microblog*, in quanto caratterizzati da elementi lontani dalla realtà dei dati descritti all'interno dei corpora di interesse, quali testi di lunghezza relativamente breve, forme dialettali e gergali, costrutti impliciti e infine errori grammaticali. Ad esempio, gli articoli forniti dallo strumento Wikipedia sono documenti di dimensione di gran lunga superiore ai semplici *tweet*. Pertanto, l'attenzione verte prevalentemente su corpora relativi ad ambiti quali siti o forum di dibattito e documenti di diverso genere, come educativo o giornalistico. In secondo luogo, poiché i data-set introdotti da ***IBM** Research* contengono informazioni relative prevalentemente ai componenti dei molteplici argomenti individuati all'interno degli articoli oggetto di interesse, i.e. *claim* ed *evidence*, è opportuno valutare quali dei rimanenti corpora in merito alla *stance classification* fornisca informazioni affini a tali concetti. Tuttavia, per quanto concerne il processo di selezione delle principali metodologie e strumenti nell'ambito della *stance classification*, non è opportuno scartare a priori tutte quelle evidenziate all'interno dei contesti precedentemente esclusi, in quanto si rischierebbe di escludere strumenti in grado di apportare notevoli miglioramenti in termini di classificazione. In particolare, come evidenziato dalla tabella 3.1, seppur la prevalenza dei lavori di ricerca ritenuti potenzialmente adatti, in accordo con il criterio di selezione presentato, appartenga ai domini applicativi precedentemente considerati, si può notare la presenza di un elemen-

to, ovvero *Stance Classification of tweets using Skip Char Ngrams*[58], facente riferimento al corpus introdotto da Mohammad[99], nell'ambito del *social network* Twitter.

Titolo	Sommario
Emergent: a novel data-set for stance classification	coppie titolo di articolo - claim etichettate con la rispettiva stance
Stance Classification of tweets using Skip Char Ngrams	coppie tweet - topic etichettate con la rispettiva stance e sentiment
Stance Classification of Ideological Debates: Data, Models, Features, and Constraints	coppie post - topic etichettate con la rispettiva stance e con informazioni aggiuntive relative al post precedente all'interno della stessa discussione
Collective Stance Classification of Posts in Online Debate	coppie post - topic etichettate con la rispettiva stance, autore e riferimento al post precedente all'interno della stessa discussione
Stance Classification using Dialogic Properties of Persuasion	coppie post - topic etichettate con la rispettiva stance e con informazioni aggiuntive relative al post precedente all'interno della stessa discussione
That's your evidence?: Classifying Stance in Onine Political Debate	coppie post - topic etichettate con la rispettiva stance e con informazioni aggiuntive relative al post precedente all'interno della stessa discussione
Stance Classification of Context-Dependent Claims	coppie claim - topic etichettate con la rispettiva stance

Tabella 3.1: Principali lavori di ricerca nell'ambito della *stance classification*, ritenuti per determinati aspetti potenzialmente affini nel rispetto dell'obiettivo di interesse.

Più precisamente, i precedenti lavori sono stati ritenuti appropriati in termini di *feature* utilizzate, tecniche di classificazione e di classi relative alla *stance classification*. In particolare, per quanto concerne le

feature, è stata effettuata una selezione, sulla base di vari criteri quali: frequenza di utilizzo all'interno dei vari lavori analizzati, efficacia nella classificazione e infine la peculiarità dello specifico data-set di interesse preso in analisi (ad esempio Emergent in questo caso). Per quanto riguarda invece l'individuazione di corpora adatti allo scopo indicato, è stato scelto come data-set per la fase di *stance classification* quello introdotto da Ferreira e Vlachos[46], indicato con il termine Emergent[1]. Le motivazioni che hanno portato alla scelta del presente corpus fanno sia riferimento al discorso sulla tipologia dei dati effettuato precedentemente che ad un ulteriore fattore legato fortemente a quest'ultimo aspetto. Tuttavia, prima di poter introdurre la presente osservazione, risulta appropriato presentare nel dettaglio il data-set Emergent, dato che quest'ultimo fa riferimento ad una particolare tipologia di dato propria del corpus in questione. Più precisamente, Emergent è stato costruito a partire da molteplici articoli di ambito giornalistico, opportunamente selezionati e annotati da differenti giornalisti in accordo con un set predefinito di affermazioni controverse, i.e. *rumoured claim*. In particolare, il presente corpus considera 2,595 articoli di giornale, ciascuno inerente ad uno dei molteplici *rumoured claim*, definiti opportunamente a priori. In maggior dettaglio, la distribuzione dei dati per ciascun *rumoured claim* risulta sufficientemente bilanciata, dove per ognuno di essi vi è almeno un articolo di giornale associato e al più cinquanta (8.65 articoli di media per ciascun *rumoured claim*). In aggiunta, dal punto di vista della sua organizzazione, il data-set riporta le seguenti informazioni: `entryID`, `claimHeadline`, `articleHeadline`, `articleHeadline stance`, `claimID` e infine `articleID`. Nello specifico, è opportuno sottolineare che il valore attribuile al campo `articleHeadline stance` appartiene al seguente spazio di scelta: `for`, `against` e infine `observing`, definendo di conseguenza uno scenario di classificazione a più classi e non binario, anche se non è esclusa a livello di algoritmo la possibilità di gestire il presente scenario come una composizione di differenti problemi di natura binaria. Pertanto, come si può facilmente intuire, la seconda motivazione dietro alla scelta di un simile data-set riguarda espressamente il *target* di riferimento della classificazione, ovvero le cosiddette *rumou-*

[1]Tecnicamente anche il corpus introdotto da Bar-Haim et al.[14] sarebbe opportuno, tuttavia quest'ultimo non era disponibile al momento dello sviluppo del presente elaborato.

red claim, le quali, seppur non in linea con il modello argomentativo di riferimento per la costruzione dei data-set introdotti da ***IBM*** *Research*, risultano comunque molto affini al concetto di *claim* considerato da Levy et al.[79] e da Rinott et al.[131]. Infine, un'ultimo fattore di interesse all'interno della presente analisi degli strumenti propri della *stance classification* riguarda direttamente le tecniche e gli algoritmi di classificazione. In particolare, visti i molteplici approcci descritti ampiamente all'interno della sezione 1.4, risulta opportuno valutare attentamente quale di essi sia potenzialmente adottabile in accordo con le precedenti scelte effettuate in merito alle *feature* e ai corpora di studio. Tuttavia, se si tiene in considerazione la scelta fatta nell'ambito della selezione del corpus, il problema decisionale in quest'ultimo ambito risulta di facile risoluzione. Infatti, non è possibile ricorrere a strategie quale la *collective stance classification*, dato che la natura dei dati descritti da Emergent non consente di definire in maniera evidente una rete di relazioni atte a permettere l'attuazione di un'attività di *stance classification*. Discorso analogo per la *sequence labeling* poiché non è presente una struttura gerarchica dei dati tale da consentire questo particolare approccio di studio. Pertanto, si ricorre alla metodologia di classificazione più canonica e frequentemente utilizzata, caratterizzata, in particolare, dall'uso di algoritmi e tecniche relativamente semplici, quali **SVM** o *Logistic Regression*, e basata infine sul paradigma di apprendimento supervisionato, avendo cura di considerare anche le informazioni proprie di ciascun *target* di classificazione. Più precisamente, la possibilità di rispecchiare le metodologie e le tecniche impiegate da Ferreira e Vlachos[46], consente al tempo stesso di poter instaurare un elemento di comparazione tra i loro risultati e quanto proposto all'interno del presente capitolo. Concludendo, è importante sottolineare come tutte le tematiche affrontate rappresentino dei punti di scelta arbitrari, di cui il presente elaborato propone solamente una delle molteplici varianti. Pertanto, già dalla prima problematica affrontata riguardante nello specifico la *stance classification* si apre la strada per differenti approcci potenzialmente equivalenti o meno, volti ad affrontare un nuovo contesto applicativo riguardante altri settori di ricerca affini quale l'*argumentation mining*.

3.2.1 Feature di classificazione

Riprendendo il discorso relativo alle *feature* di classificazione risulta opportuno soffermarsi in maggiore dettaglio su quest'ultima tematica, al fine di evidenziarne gli aspetti principali e la loro importanza dal punto di vista dell'elaborazione del testo. In particolare, al termine della presente fase di ricerca degli strumenti propri della *stance classification* maggiormente conformi all'obiettivo imposto, sono state selezionate le seguenti *feature* di classificazione.

- **Bag-of-Words (BoW)**: rappresentazione di tipo statistico del testo per mezzo della quale una frase `S` viene codificata in un vettore $v = v_1, \ldots, v_{|D|}$ di valori binari, dove `D` è il dizionario dei termini individuati all'interno di tutte le frasi, e $v_j = 1$ se la parola associata w_j, presente in `D`, appare all'interno di `S`. In particolare, il presente modello è stato studiato ampiamente, con particolare attenzione alle variante **TF-IDF**[136], la quale tiene conto della frequenza di accadimento di una parola all'interno di una frase, i.e. *term frequency* (**TF**), e della rarità del termine stesso all'interno del vocabolario, i.e. *inverse document frequency* (**IDF**).

- **N-gram**: viene generalmente definito come la sequenza di termini adiacenti di grado `n`, ovvero il numero di parole da considerare per formare uno specifico *n-gram*. In particolare, se si associa a ciascun *n-gram* la sua frequenza di accadimento all'interno dei documenti di interesse secondo il criterio **TF-IDF** si ottiene una generalizzazione del concetto di **BoW**.

- **Cosine similarity**: si intende la similarità coseno calcolata tra le rappresentazioni *word vector* di ciascun *target* di classificazione, effettuata ricorrendo ai dati già strutturati di Google, i.e. *Word2Vec*[90]. Più precisamente, dati due vettori `A` e `B`, la *cosine similarity* viene calcolata mediante la seguente equazione:

$$similarity = cos(\theta) = \frac{A \cdot B}{||A||_2||B||_2} \tag{3.1}$$

- **Subject-Verb-Object (SVO) triples**: triple {soggetto, verbo, oggetto} individuate da un opportuno strumento di analisi, i.e.

parser, estratte per tutte le proposizioni di ciascun *target* di classificazione. In particolare modo, viene effettuata una comparazione di tutte le coppie incrociate di elementi di due triple **SVO**, appartenenti rispettivamente ai due *target* di classificazione.

- **Basic**: comprende un vasto set di *feature* di tipo quantitativo quali: numero di caratteri, lunghezza media delle parole in termini di caratteri, numero di frasi, numero di parole, lunghezza media delle frasi in termini di parole, conteggio dei seguenti simboli {'?', '!', '"'}, conteggio delle coppie di parentesi, percentuale di parole con più di sei lettere, percentuale di forme pronominali, percentuale delle cosiddette *sentiment word*. In particolar modo, le ultime tre *feature* descritte rientrano nel set di *feature* descritto da Anand et al.[5].

- **Discourse Cues**[146]: frequenza del primo *uni-gram*, *bi-gram* e *tri-gram* di ciascun testo.

- **Repeated Punctuation**[146]: conteggio dei simboli di punteggiatura.

- **Linguistic Inquiry and Word Count (LIWC) *feature***[120]: conteggio delle categorie lessicali individuate, numero di parole per frase (**WPS**), numero di forme pronominali (**Pro**), numero di parole relative ad emozioni positive o negative. In particolar modo, ciascuna delle presenti *feature* è normalizzata in base alla frequenza di accadimento.

- **Dependency relation**[5]: comprende tre diverse tipologie di *feature*, aventi in comune una definizione di base descrivente queste ultime come triple `(rel, w1, w2)`, dove `rel` indica la relazione di dipendenza grammaticale tra i due termini `w1` e `w2`. Le seguenti varianti, denominate secondo il termine di *depency relation feature*, si distinguono in base alla natura dei termini `w1` e `w2`.

  - **Syntatic**: `w1` e `w2` sono rappresentati da semplici parole individuate all'interno del testo.
  - **POS generalized**: si sostituisce `w1` con la rispettiva etichetta *Part-of-Speech* (**POS**).

  - **Opinion generalized**: si seleziona, per mezzo di uno opportuno strumenti di analisi del *sentiment*, quale il *lexicon* **MPQA**[165], il sottoinsieme di *dependency relation feature* aventi almeno uno dei due termini caratterizzato da un *sentiment* non neutro. In particolare, per ognuna di esse si rimpiazza la cosiddetta *opinion word* con la sua rispettiva polarità, ossia positiva o negativa.

- **Frame Semantic Features**: utilizzo di strumenti, quali *FrameNet*[11] e *SEMAFOR*[33], per creare dei costrutti, i.e. *frame*, volti a catturare la dimensione semantica di ciascuna proposizione individuata all'interno del testo. In particolar modo, per ogni *frame* si definiscono tre diverse tipologie di *feature*.

  - **Frame-word interaction feature**: *feature* binaria composta da A) il nome del *frame* dalla quale è definita, B) coppia non ordinata di parole, rappresentanti due particolari elementi del *frame* in questione, i.e. *frame element*. Più precisamente, per ogni coppia di *frame element* si crea, a partire da ogni coppia non ordinata di parole, la cosiddetta *frame-word interaction feature*, composta da parole appartenenti rispettivamente ai due *frame element* considerati.
  - **Frame-pair feature**: *feature* binaria composta da una coppia di parole relative ai nomi dei due *frame*, nella quale il *target* del primo è presente all'interno di un *frame element* appartenente al secondo.
  - **Frame n-gram feature**: rappresenta la versione basata su *frame* della *feature* legata ad un *n-gram*. Più precisamente, dato un *uni-gram* o un *bi-gram* dove ogni parola è una *open class word*, si creano, a partire da quest'ultimo, tutte le possibili *frame n-gram feature*, rimpiazzando una o più parole con il nome del *frame* se esse rappresentano il suo *target*, oppure con il ruolo semantico del *frame* se la parola è presente all'interno di un particolare *frame element*.

- **Skip n-grams**: costruzione simile a quella degli *n-gram*, differente tuttavia per il criterio di selezione delle parole per formare gli

n-gram. In particolare, la selezione non è più basata sulla adiacenza tra parole, bensì si considerano anche elementi con distanza maggiore di 1 in termini di parole.

Tuttavia, solamente alcune delle *feature* elencate sono state poi implementate o utilizzate durante la classificazione per l'ottenimento dei risultati. I motivi variano dalla semplice impossibilità d'uso per questioni di utilizzo proprietario (**LIWC**), complessità implementativa (*Frame Semantic Features*) e infine lo scarso impatto una volta applicate sui corpora di interesse (*Discourse Cues*, *Opinion Generalized*). Più precisamente, le *feature* evidenziate in rosso sono quelle per cui non è stata definita un' implementazione o che non sono state utilizzate durante la classificazione. Infine, oltre alle *feature* selezionate all'interno dei lavori relativi alla *stance classification* elencati, sono stati sperimentati i cosiddetti *Syntactic n-gram*[137] (**sn-gram**), ossia degli *n-gram* costruiti sulla base delle relazioni sintattiche ottenute per mezzo di un opportuno *parser*, limitandoli tuttavia al caso di default dei *bi-gram*. Il motivo è dato dal fatto che gli esempi riportati nel lavoro di presentazione di tale *feature*, presentano il processo di individuazione delle dipendenze da parte dello strumento di analisi *Stanford Dependency Parser* che differisce da quella ottenuta sperimentalmente. Per questo motivo, seppur a livello logico l'estensione implementativa ai casi in cui il grado `n` sia maggiore di 2 appaia corretta, si è ritenuto preferibile limitarsi al caso base, in quanto il set di *sn-gram* individuati coincide per l'appunto con le coppie di dipendenza estratte dallo *Stanford Dependency Parser*. Tutti i dettagli implementativi, relativi al procedimento di estrazione di ciascuna *feature* in termini di codice, sono riportati in dettaglio all'interno della appendice A.

3.3 Procedimento di costruzione di un classificatore per la *stance classification*

Il passo successivo, una volta conclusa la fase di corretta implementazione di tutte le *feature* precedentemente introdotte, consiste nella definizione di un classificatore in grado di raggiungere un livello di prestazioni comparabile con quanto presentato da Ferreira e Vlachos[46] sul corpus

di Emergent. In particolar modo, i requisiti principali che regolano il presente processo implementativo vertono su fattori quali efficienza computazionale ed efficacia. Più precisamente, la capacità di adattamento ad un generico data-set di interesse deve coniugarsi con la facoltà di poter gestire molteplici ammontare di dati di dimensione arbitraria, nell'ottica anche delle *feature* di classificazione utilizzate. Infatti, all'interno del presente contesto di discussione occorre tener conto delle risorse richieste dalla fase di estrazione di ciascuna *feature*, in quanto potrebbero insorgere eventuali problemi di scalabilità. Ad esempio, se si considera il caso delle triple **SVO** o delle *dependency relation*, queste ultime necessitano della presenza di strumenti esterni, i.e. *parser*, i quali in alcuni casi possono richiedere tempi di computazione non trascurabili. Ancora, per quanto concerne il calcolo della *cosine similarity*, è necessario dapprima effettuare la codifica delle porzioni di testo prese in esame nella loro rispettiva rappresentazione tramite *word vector*, azione che richiede il caricamento in memoria di strutture dati di ausilio di dimensioni significative, comportando di conseguenza un notevole rallentamento del processo di estrazione delle *feature*. In secondo luogo, una volta superato il presente problema di ottimizzazione, occorre effettuare diversi test al fine di individuare la miglior configurazione possibile dei parametri propri del classificatore costruito, ossia quella per cui i valori delle metriche di giudizio, scelte per quest'ultimo come strumento di misurazione delle performance, sono i più elevati possibile. Pertanto, occorre procedere per passi, in modo tale da poter descrivere nel dettaglio tutti i punti di criticità individuati e affrontati all'interno del procedimento di costruzione del classificatore per la *stance classification*.

3.3.1 Ottimizzazione

Il primo aspetto affrontato riguarda il processo di ottimizzazione della fase di estrazione delle *feature* di interesse. Più precisamente, il risultato che si vuole ottenere al termine della suddetta fase è rappresentato da una matrice di valori associata a tutte le *feature* impiegate, ciascuna delle quali verte a catturare una particolare dimensione sintattica o semantica, espressa all'interno dei documenti testuali oggetto di interesse. In particolar modo, come è già stato anticipato precedentemente, ognuna delle *feature* selezionate presenta come elemento di distinzione strumenti

e risorse computazionali specifiche. Più precisamente, tutte le *feature*, ad eccezione degli *n-gram* e *skip n-gram*, richiedono un particolare processo di ottimizzazione, all'interno del quale i punti critici del calcolo dei dati associati alle *feature*, vengono memorizzati in maniera permanente, così che, una volta pre-calcolati e salvati tutti i dati, ogni fase della classificazione, che richieda il calcolo delle matrici dei valori associate a tali *feature*, può essere effettuata in un quantitativo di tempo notevolmente inferiore e senza richiedere nuovamente l'impiego degli strumenti d'ausilio specifici. Per fare un esempio, la *feature cosine similarity*, in quanto facente uso dei dati strutturati di Google, i.e. *Word2Vec*, richiede un grande quantitativo di memoria, i.e. **RAM**. D'altro canto, *feature* quali *dependency relation*, *sn-grams* e triple **SVO**, in quanto basate sul medesimo strumento di analisi, i.e. *parser*, richiedono, in particolare, un grande quantitativo di tempo. Pertanto, la definizione di un'opportuna architettura di supporto, atta a gestire la presente problematica, consente di ridurre drasticamente le tempistiche di calcolo associate alle *feature* di interesse, sia nell'ambito della loro semplice estrazione che in contesti specifici come, ad esempio, la calibrazione dei rispettivi parametri di configurazione. Tutti i dettagli implementativi in merito al processo di ottimizzazione dell'estrazione delle *feature* di interesse, soffermandosi in particolar modo sui miglioramenti in termini di risorse computazionali e tempistiche, sono evidenziati e riportati in dettaglio all'interno dell'appendice A.

3.3.2 Ricerca della miglior configurazione

Superato il problema relativo al calcolo della matrice dei valori associata a ciascuna *feature*, si pone il problema di individuarne la miglior combinazione in congiunzione ai loro rispettivi parametri di configurazione. A tal fine è necessario effettuare una ricerca esaustiva, i.e. *gridsearch*, provando di volta in volta ognuna delle possibili combinazioni. Tuttavia, occorre valutare dapprima l'applicabilità di tale ricerca. In primo luogo, gran parte delle *feature*, ad eccezione di *cosine similarity* e delle triple **SVO**, può fare riferimento a ciascuno dei due target della *stance classification*, il che raddoppia il numero di *feature* a disposizione durante la fase di classificazione. Più precisamente, delle 9 *feature* di partenza, 7 sono applicabili a ciascun *target*, ottenendo di conseguenza un totale di 16 *feature*. Ad

esempio, la *feature basic* può essere calcolata sia per `articleHeadline` che per `claimHeadline` nel caso in cui si consideri il *corpus* di Emergent. Purtroppo, considerare tutte le possibili combinazioni di 16 *feature* richiede un arco di tempo a livello computazionale estremamente ampio, anche nonostante le ottimizzazioni effettuate. Come conseguenza, si è deciso di optare per la ricerca di una soluzione sub-ottima, ma il più possibile valida per la classificazione. Pertanto, successivamente viene introdotto e descritto accuratamente il presente problema di ottimizzazione non ottimale, con particolare attenzione a confrontare i risultati ottenuti con quelli proposti da Ferreira e Vlachos[46], al fine di poterne garantire la validità. Innanzitutto, il primo problema da affrontare riguarda la determinazione dei parametri di configurazione delle funzioni di estrazione delle *feature* di interesse. In particolare, le uniche richiedenti un simile procedimento sono le seguenti: *n-gram* e *skip ngram*. Nello specifico, si ricerca la determinazione dei valori ottimali nell'ambito di parametri quali la distanza massima in termini di parole da considerare per la formazione degli *n-gram*, i.e. `skip_range`, l'intervallo dei valori associati a `n`, rappresentato da una tupla di grado 2, e infine il numero di *n-gram* da selezionare, ordinati in base alla frequenza di accadimento, i.e. `max_features`. A tale scopo, si individuano i seguenti scenari di studio, atti a rappresentare la strategia di ricerca sub-ottimale precedentemente introdotta.

- Calibrazione dei parametri di configurazione delle *feature n-gram* e *skip n-gram*, specifici per ciascuno dei due *target* della classificazione, considerando tutte e 16 le *feature*.

- Calibrazione dei parametri di configurazione delle *feature n-gram* e *skip n-grams*, specifici per ciascuno dei due target della classificazione, considerando singolarmente le *feature* associate a ciascun *target*. In altre parole, dapprima si calibrano i parametri considerando il primo *target* e quindi solamente le *feature* associate ad esso con l'eccezione di *cosine similarity* e delle triple **SVO**. Ragionamento analogo per il secondo *target*. Ad esempio, nel caso del data-set Emergent, il primo *target* di classificazione è rappresentato da `claimHeadline`, per cui si applicano tutte le *feature* esclusivamente per il presente *target*, ad eccezione delle due prece-

dentemente elencate. Infine, si ripete lo stesso procedimento per il secondo *target* di classificazione, i.e. `articleHeadline`.

Le tabelle 3.3 e 3.4 riportano i parametri ottenuti al termine di ciascuna fase di calibrazione, associata ad uno dei due casi di studio di interesse. In particolare, dal punto di vista tecnico, è opportuno sottolineare che tutte le ricerche esaustive sono state misurate mediante la tecnica della validazione incrociata, i.e. *cross validation*, caratterizzata dal numero di suddivisioni dei dati i.e. *split*, per le molteplici fasi di apprendimento interne pari a 3. In aggiunta, il criterio di separazione dell'insieme dei dati di apprendimento, i.e. *training set*, nelle diverse suddivisioni, i.e. *fold*, è dato dalla classe `StratifiedKFold` con parametro `shuffle=True`, il quale consente di ottenere *fold* ogni volta diverse ma sempre bilanciate in termini di distribuzione delle differenti classi. Per quanto riguarda le metriche, si considera solamente la *accuracy*, in quanto si mantiene come elemento di comparazione il lavoro proposto da Ferreira e Vlachos[46], all'interno del quale la prevalenza delle performance del classificatore è misurata per mezzo della presente metrica, ad eccezione dell'insieme dei dati di verifica, i.e. *test set*, per il quale si considerano anche i valori relativi a *precision* e *recall* di ciascuna classe. Dal punto di vista dello spazio dei parametri selezionabili per ciascuna *feature* all'interno della *gridsearch*, la tabella 3.2 riporta la lista dei valori di ciascun parametro di configurazione, utilizzata durante la fase di calibrazione. Infine, sono stati scelti come classificatori le seguenti classi: `LogisticRegression`, `SGDClassifier` e `LinearSVC`, sempre per motivazioni riconducibili al lavoro di Ferreira e Vlachos[46] dove la classe di riferimento è `LogisticRegression`. Pertanto, come riportato chiaramente dalle tabelle 3.3 e 3.4, la fase di calibrazione dei parametri viene interessata più volte in base al particolare classificatore di interesse. Più precisamente, in quest'ultimo caso, in quanto scelta come classificatore all'interno del progetto Emergent, è possibile effettuare un'analisi di confronto diretta con i risultati proposti da quest'ultimo al fine di valutare l'efficacia dell'utilizzo di un set di *feature* differente, quale quello preso in esame.

Parametro	Feature	Lista dei valori attribuibili
n-gram range	Ngrams	{(1, 1), (1, 2), (1, 3)}
	Skipgrams	{(2, 2), (2, 3)}
skip_range	Skipgrams	{(2, 2), (2, 3), (2, 4), (2, 5), (2, 6)}
max_features	Ngrams	{None, 10000, 20000}
	Skipgrams	{None, 10000, 20000}

Tabella 3.2: Spazio dei valori associato a ciascun parametro di configurazione, proprio di una *feature* specifica. In particolar modo, per quanto riguarda il parametro *max_features*, il valore *None* indica l'assenza di alcuna delimitazione nell'ambito di selezione dei migliori valori in termini di frequenza di accadimento. Pertanto, si considerano tutti i valori, a differenza degli altri casi in cui solamente i primi 10000 o 20000 sono selezionati.

SGDClassifier	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 2)		20000	0.7086
Skipgrams_first	(2, 3)	(2, 3)	None	
Ngrams_second	(1, 1)		10000	0.7144
Skipgrams_second	(2, 2)	(2, 2)	None	

LogisticRegression	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 3)		None	0.7032
Skipgrams_first	(2, 2)	(2, 2)	None	
Ngrams_second	(1, 1)		None	0.7167
Skipgrams_second	(2, 3)	(2, 4)	10000	

LinearSVC	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 3)		None	**0.7136**
Skipgrams_first	(2, 2)	(2, 6)	10000	
Ngrams_second	(1, 1)		None	**0.7283**
Skipgrams_second	(2, 3)	(2, 6)	10000	

Tabella 3.3: Risultati calibrazione *n-gram* e *skip n-gram* considerando tutte e 16 le *feature*. I suffissi *first* e *second* fanno riferimento ai *target* della classificazione. Nel caso di training su Emergent sono *claimHeadline* e *articleHeadline* rispettivamente.

SGDClassifier	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 2)		None	0.6285
Skipgrams_first	(2, 2)	(2, 2)	None	
Ngrams_second	(1, 1)		20000	0.7325
Skipgrams_second	(2, 2)	(2, 4)	None	

LogisticRegression	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 1)		None	**0.6323**
Skipgrams_first	(2, 3)	(2, 4)	10000	
Ngrams_second	(1, 1)		None	**0.7425**
Skipgrams_second	(2, 3)	(2, 6)	10000	

LinearSVC	n-gram range	skip k-range	max_features	accuracy
Ngrams_first	(1, 1)		None	0.6312
Skipgrams_first	(2, 3)	(2, 5)	10000	
Ngrams_second	(1, 1)		None	0.7368
Skipgrams_second	(2, 3)	(2, 5)	10000	

Tabella 3.4: Risultati calibrazione *n-gram* e *skip n-gram* considerando singolarmente le *feature* associate a ciascun *target*. I suffissi *first* e *second* fanno riferimento ai *target* della classificazione. Nel caso di training su Emergent sono *claimHeadline* e *articleHeadline* rispettivamente.

Successivamente, una volta ottenuti i migliori parametri di configurazione per ciascuno dei casi di studio di interesse, occorre determinare il sottoinsieme di *feature* in grado di catturare maggiormente le dimensioni sintattiche e semantiche dei documenti testuali presi in esame. In altre parole, è necessario introdurre una fase di scrematura delle *feature*, in accordo con la metrica specifica di riferimento utilizzata per misurare le performance dei molteplici classificatori considerati. A tal fine, si definisce il cosiddetto test di ablazione, i.e. *ablation test*, il quale consiste nell'allenare ciascun classificatore di interesse con un set di *feature* che di volta in volta esclude una particolare *feature*. Più precisamente, si effettua un numero di *cross validation* pari a quello delle *feature*, ciascuna delle quali non considera una determinata *feature* appartenente al set iniziale a disposizione. A tal fine, è possibile ottenere una stima delle performance del classificatore in assenza del contributo di ogni *feature* di

interesse, in modo tale da poter discriminare quelle potenzialmente più efficaci dalle rimanenti. Nello specifico, si effettua il test di ablazione considerando tutte e 16 le *feature*, con i parametri relativi a *n-gram* e *skip n-gram* individuati nei due casi migliori. In particolar modo, anche in questo scenario la metrica di riferimento è la *accuracy* e il numero di *split* nella *cross validation* è pari a 10. Per ogni singola fase di ciascun test di ablazione viene salvato il risultato della *cross validation*. Le tabelle 3.5 e 3.6 riportano i risultati ottenuti nei due rispettivi casi di studio precedentemente introdotti. Più precisamente, facciamo riferimento con il termine **Ablation All** per indicare il test di ablazione effettuato con i parametri associati a *n-gram* e *skip n-gram*, ottenuti considerando tutte e 16 le *feature*. Viceversa, per il secondo caso utilizziamo il termine **Ablation Divided**.

Feature	Accuracy
single_ngrams_second	0.713658682432
single_skipgrams_second	0.723283102758
cosine_similarity	0.725204740329
basic_second	0.72829354355
single_skipgrams_first	0.729061300697
repeated_punctuation_second	0.729064259321
POS_generalized_first	0.729445916082
syntactic_first	0.729445916082
sngrams_first	0.729445916082
basic_first	0.729447401083
repeated_punctuation_first	0.729447401083
single_ngrams_first	0.729448874705
svo_triples	0.730222560611
POS_generalized_second	0.733694551112
sngrams_second	0.734848466071
syntactic_second	0.736392879127

Tabella 3.5: Test di ablazione **Ablation All**. Il prefisso *single* indica il fatto che la *feature* in questione è specifica per uno solo dei due *target*.

Feature	Accuracy
single_ngrams_second	0.711328726481
single_skipgrams_second	0.71751531187
POS_generalized_second	0.723701908638
cosine_similarity	0.725604229281
basic_second	0.726759548925
sngrams_second	0.728306977766
svo_triples	0.729062843125
basic_first	0.729460801029
repeated_punctuation_second	0.729460801029
repeated_punctuation_first	0.729460801029
single_ngrams_first	0.730234486802
POS_generalized_first	0.730623580215
syntactic_second	0.731394307365
syntactic_first	0.731778922749
sngrams_first	0.731778922749
single_skipgrams_first	0.732159094509

Tabella 3.6: Test di ablazione **Ablation Divided**. Il prefisso *single* indica il fatto che la *feature* in questione è specifica per uno solo dei due *target*.

A questo punto, basandoci esclusivamente sui risultati dei test di ablazione, ordinando in ordine crescente rispetto alla metrica di riferimento, si considerano le prime 8 *feature*, ovvero quelle ritenute le più rilevanti all'interno del processo di classificazione. Successivamente, si effettua una *gridsearch* solamente sul sottoinsieme di *feature* indicato. La tabella 3.7 riporta il valore della metrica di riferimento e la miglior combinazione di *feature* ottenute.

Scenario	accuracy	combinazione feature
Ablation-All	**0.7595**	basic_second, single_ngrams_second, cosine_similarity
Ablation-Divided	0.7514	single_skipgrams_second, basic_second, basic_first, single_ngrams_second, cosine_similarity, svo_triples

Tabella 3.7: Risultati della ricerca esaustiva per ciascun set di 8 *feature* estratti dai relativi test di ablazione.

Una volta terminata quest'ultima fase, si procede infine con l'ultima calibrazione interessante i parametri di configurazione propri di ciascun classificatore utilizzato nei due scenari presi in esame. In particolare, come si può osservare dalla tabella 3.8, i migliori classificatori sono rappresentati dalle classi `LinearSVC` e `LogisticRegression`, dove quest'ultimo supera di poco il primo. Pertanto, oltre alla semplice ricerca della miglior combinazione di *feature*, parallelamente è stato portato avanti un'ulteriore processo di ottimizzazione, interessante nello specifico l'individuazione della migliore tipologia di classificatore. Così facendo, l'ultimo aspetto da considerare all'interno del presente capitolo riguarda la predizione della *stance* per ciascun elemento contenuto all'interno del data-set Emergent.

Scenario	classificatore	parametri	accuracy
Ablation-All	LinearSVC	C = 1.0, loss = squared_hinge	0.7595
Ablation-Divided	LogisticRegression	C = 10, penalty = L1	**0.7618**

Tabella 3.8: Risultati della calibrazione dei parametri associati a ciascun classificatore.

Infine, come ultimo passo volto a verificare la validità del processo di ottimizzazione precedentemente introdotto, è opportuno verificare se tale sottoinsieme di *feature* sia stato una scelta sensata o meno. In altre parole, occorre precisare se il test di ablazione può rappresentare un valido

strumento all'interno della presente ricerca della miglior configurazione di *feature*. Pertanto, come verifica, effettuiamo la stessa ricerca esaustiva considerando il secondo sottoinsieme di *feature* scartato in precedenza (tabella 3.9).

Scenario	accuracy	combinazione feature
Ablation-All	0.6770	single_ngrams_first, POS_generalized_second
Ablation-Divided	0.6473	POS_generalized_first, syntactic_second

Tabella 3.9: Risultati della ricerca esaustiva di verifica selezionando le rimanenti 8 *feature* per ciascun test di ablazione effettuato.

3.4 Predizione della *stance* dei dati appartenenti a CE-ACL-14 e CE-EMNLP-15

Prima di poter passare all'ultima fase di interesse relativa alla predizione della *stance* dei dati appartenenti ai data-set **CE-ACL-14** e **CE-EMNLP-15** è necessario verificare opportunamente se il classificatore precedentemente definito è comparabile con lo stato dell'arte della *stance classification*, con particolare attenzione ai risultati proposti da Ferreira e Vlachos[46]. A tal fine, si considera lo stesso *test set* introdotto all'interno del progetto Emergent, in modo tale da poter ottenere delle statistiche atte al confronto. Più precisamente, la presente collezione di dati è formata da 524 elementi di diverso genere rispetto a quelli appartenenti al *training set*, in quanto tale aspetto contribuisce alla verifica delle capacità di generalizzazione del classificatore preso in esame. Dal punto di vista delle tecniche di misurazione delle performance, si calcolano la *precision* e il *recall* per ciascuna classe relativa alla *stance* e la rispettiva matrice di confusione, al fine di poter valutare le capacità di discriminazione del classificatore di interesse (tabelle 3.10 e 3.11). In aggiunta, si tiene conto anche della *accuracy* generale, così come riportato da Ferreira e Vlachos[46]. A tal proposito, i risultati ottenuti dal classificatore si avvicinano di molto a quelli presentati nel progetto Emergent,

migliorando addirittura leggermente la *accuracy* globale, pari a 0.7519 rispetto a 0.7300 ottenuta da Ferreira e Vlachos[46].

	for	against	observing
Precision	0.8262	0.6410	0.7251
Recall	0.7926	0.8241	0.6631

Tabella 3.10: Performance del classificatore sul test set relativo al progetto Emergent. Per ogni classe sono riportate informazioni quali *precision* e *recall*.

Predicted	against	for	observing
True			
against	75	2	14
for	18	195	33
observing	24	39	124

Tabella 3.11: Matrice di confusione ottenuta dal classificatore sul test set di Emergent.

Sulla base dei risultati soddisfacenti ottenuti nell'ambito della *stance classification*, si può procedere con l'analisi dei data-set di interesse per la *argument structure prediction*, i.e. **CE-ACL-14** e **CE-EMNLP-15**, al fine di poter effettuare un'ulteriore verifica delle performance del classificatore e definire i corpora di riferimento per i prossimi obiettivi indicati all'inizio del capitolo e oggetto di interesse del presente elaborato. Procedendo con ordine, risulta necessario soffermarsi in maggior dettaglio sulla struttura dei data-set per la *argument structure prediction*. Nello specifico, i corpora introdotti rispettivamente da Aharoni et al.[4] e Rinott et al.[131] seguono la medesima impostazione, ovvero prendono come riferimento i seguenti campi principali: `topic`, `claim`, `evidence`, denominata `CDE` nel caso di **CE-ACL-14**, e infine `article`, relativo al documento di tipo giornalistico da cui sono state estratte le altre informazioni. In particolare, si parla principalmente di *context-dependent claim* (**CDC**): 'un'affermazione breve e generale che supporta o contesta in maniera esplicita il *topic* di riferimento'; e di *context-dependent evidence* (**CDE**), definita come 'una porzione di testo supportante un dato

claim all'interno del contesto definito dal particolare *topic* preso in esame'. Inoltre, dal punto di vista quantitativo, **CE-ACL-14** è costituito da 1,392 *claim* etichettati, relativi a 33 *topic* differenti, e 1,291 *evidence* annotate per 350 *claim* specifici associati a 12 *topic* di diversa natura. Viceversa, **CE-EMNLP-15** è formato da 2,294 *claim* e da 4,690 *evidence* entrambi opportunamente annotati, relativi a 58 *topic* distinti. In particolare, si distinguono tre diverse tipologie di *evidence*:

- ***Study***: descritte generalmente come dei risultati derivati da fenomeni quali analisi quantitative di determinati dati, verità matematiche, fatti basati sulla realtà o eventi storici. Ad esempio: *"Tropical deforestation is responsible for approximately 20% of world greenhouse gas emissions [REF]."*, dove [REF] indica un collegamento o riferimento ad un'altra fonte di testo.

- ***Expert***: riporta le testimonianze di persone, gruppi, comitati o organizzazioni aventi al loro interno dei membri o delle autorità esperte nell'ambito del *topic* di riferimento. Ad esempio: *"Dr. Gary Kleck, a criminologist at Florida State University, estimated that approximately 2.5 million people used their gun in self-defense or to prevent crime each year, often by merely displaying a weapon."*.

- ***Anedoctal***: relativa ad un'affermazione descrivente eventi specifici o particolari episodi, i.e. aneddoti. Ad esempio: *"The Orderly Departure Program from 1979 until 1994 helped to resettle refugees in the United States as well as other Western countries."*.

Pertanto, poiché sono presenti diversi potenziali *target* nell'ottica della *stance classification*, quali *topic* e *article* in particolare, risulta necessario valutare quali scenari tenere in considerazione per la fase di classificazione. D'altro canto, gli elementi di interesse per i quali si definisce il concetto di *stance* nei confronti di un dato riferimento sono ben noti e rappresentati da il *claim* e dalla *evidence*, i.e. *premise*. Come conseguenza di quanto appena osservato, si delineano diversi scenari applicativi nell'ambito della *stance classification*, denominati, ad esempio, *claim - topic* nel caso in cui si consideri come elemento di interesse il *claim* e come *target* il *topic*, oppure *claim - article* se si sostituisce quest'ultimo con l'articolo di tipo giornalistico. Discorso analogo per quanto concerne

la *evidence*. Nello specifico, per motivi legati prettamente alla dimensione semantica dei contenuti, si è deciso di considerare come unico *target* di riferimento il *topic* in quanto rappresentante un elemento ambito di discussione e in grado di consentire l'individuazione di posizionamenti ben marcati. Inoltre, la presente motivazione trova ulteriore giustificazione nell'ambito della *argument structure prediction* quando si considera il problema dell'assenza di esempi negativi atti a consentire l'attuazione di un'attività di classificazione. Pertanto, il presente elaborato tiene conto solamente dei seguenti scenari di interesse: *claim - topic* e *evidence - topic*. Per consentire una valutazione sotto più punti di vista dei risultati riportati dal classificatore scelto, si considerano elementi di tipo statistico relativi a fattori quali la distribuzione delle classi proprie della *stance classification* ed eventuali incoerenze in termini di valori associati alla *stance* all'interno di una stessa coppia *claim - evidence* (tabella 3.12). In particolare, quest'ultimo aspetto, sulla base delle ipotesi di costruzione dei data-set proposti da ***IBM*** *Research*, può rappresentare un'importante indicatore delle performance del classificatore, in quanto le caratteristiche dei corpora suggeriscono che le *stance* relative ad una coppia *claim - evidence* nei confronti di un determinato *topic* debbano coincidere.

Data-set	# for		# against		# observing		Coppie Incoerenti	Coppie fortemente incoerenti	Coppie Totali
CE-ACL-14	claim	evidence	claim	evidence	claim	evidence	753	235	1378
	766	751	221	278	391	349			
CE-EMNLP-15	claim	evidence	claim	evidence	claim	evidence	2669	549	5168
	2710	2975	553	672	1905	1521			

Tabella 3.12: Statistiche estratte dalla predizione della *stance* sui data-set di interesse nell'ambito degli scenari *claim-topic* ed *evidence-topic*. Le *stance* incoerenti rappresentano elementi per cui la *stance claim-topic* differisce da quella stimata per il caso *evidence-topic* nel rispetto di una stessa coppia *claim - evidence*. In altre parole, tale dato può essere interpretato come il numero di coppie *claim-evidence* aventi *stance* non identica. Infine, i campi *Coppie incoerenti* e *Coppie fortemente incoerenti* rappresentano il numero di coppie *claim - evidence* i cui valori delle due *stance* associate non sono uniformi e non contengono la classe *observing* rispettivamente.

Capitolo 4

Risultati sperimentali ottenuti sui nuovi corpora nell'ambito della argument structure prediction

Una volta terminata la fase relativa alla *stance classification* delle *claim* ed *evidence* nei confronti di un dato *topic*, si può iniziare a ragionare in termini di *argument structure prediction*. Più precisamente, nel presente contesto sperimentale si hanno a disposizione i due data-set introdotti da ***IBM** Research*, ovvero **CE-ACL-14** e **CE-EMNLP-15**, i cui dati, per la precisione i *claim* e le *evidence*, i.e. *premise*, sono stati annotati da un opportuno classificatore specifico della *stance classification*. Pertanto, il prossimo passo da compiere consiste nella definizione di ulteriori strumenti di classificazione in grado di proporre una soluzione soddisfacente al problema rappresentato dalla *argument structure prediction*. In particolare, nell'ottica degli obiettivi generali introdotti all'inizio del capitolo precedente e rappresentanti la visione di fondo del presente elaborato, si considerano come traguardi da raggiungere i due punti centrali elencati, ovvero la determinazione di un potenziale collegamento tra *stance classification* e *argument structure prediction* e la sperimentazione di strumenti specifici del primo settore di ricerca citato in quest'ultimo contesto di interesse. A differenza di quanto considerato per la *stance classification* all'interno del capitolo precedente, il presente scenario applicativo non

presenta la possibilità di poter definire dei riferimenti per la comparazione dei risultati ottenuti, in quanto non sussistono al momento ricerche relative all'ambito dell'*argument structure prediction* basate sugli stessi strumenti di interesse per il presente elaborato. Pertanto, risulta necessario descrivere accuratamente tutte le eventuali problematiche riscontrate durante il processo di implementazione insieme alle ipotesi iniziali di lavoro, così da poter suggerire potenziali soluzioni future o varianti di quanto proposto. Pertanto, introduciamo la struttura del presente capitolo come segue. Inizialmente, si procede con la definizione dei molteplici sotto-obiettivi specifici del contesto di applicazione di interesse, quali, ad esempio, l'analisi dei corpora a disposizione, con particolare attenzione per il problema dell'assenza di esempi negativi per classificazione, e lo studio di nuove *feature*, inerenti alla *argument structure prediction* ma sperimentate in altri contesti di diverso genere. In particolar modo, il primo aspetto citato prevede il soffermarsi su due principali punti di criticità, di cui il secondo è conseguenza logica del primo: la costruzione di coppie *evidence - claim* caratterizzate da un legame di tipo non supporto, i.e. *opposing evidence - claim*, e il problema del bilanciamento dei dati per la classificazione. Una volta delineate e affrontate in maniera opportuna le precedenti problematiche, si procede alla definizioni di molteplici classificatori distinti, atti ad individuare correttamente le relazioni di supporto e non all'interno dei corpora di interesse. Infine, il capitolo si conclude con la presentazione dei risultati di classificazione, con particolare riferimento alle differenti tecniche implementate, al fine di poter effettuare dapprima una semplice comparazione dei vari approcci, ma soprattutto per trarre importanti osservazioni in merito al caso di studio di interesse.

4.1 Obiettivo

Ricollegandosi al precedente discorso introduttivo, l'obiettivo del presente capitolo verte principalmente su due aspetti generali: (1) l'individuazione di un legame tra due diversi settori di ricerca nell'ambito del **NLP**, ovvero la *stance classification* e l'*argument structure prediction*, e (2) la sperimentazione di tecniche proprie della *stance classification* in un diverso contesto applicativo, ovvero l'*argument structure prediction*,

in modo tale da poter comparare i risultati ottenuti con alcune *baseline* di riferimento e altri approcci specifici di quest'ultimo campo di ricerca. Nello specifico, si considera l'attività di *argument structure prediction* come un problema di classificazione binaria, dove le classi associate alla relazione tra argomenti descrivono legami di tipo supporto e non. Ad esempio, all'interno del presente elaborato si fa riferimento ai seguenti valori: `link` e `no-link`. A tal fine, in modo tale da poter scomporre le presenti linee guida in sotto-obiettivi di più semplice gestione e risoluzione, si delineano i seguenti punti operativi, volti a descrivere passo per passo l'intero procedimento implementativo proposto.

- Costruzione di corpora atti a consentire un'attività di *argument structure prediction*. A tal fine, a partire dai data-set a disposizione, ovvero **CE-ACL-14** e **CE-EMNLP-15**, occorre valutare opportune strategie applicative aventi come obiettivo la definizioni di esempi positivi e negativi per la fase di classificazione. Nello specifico, si considerano due possibili opzioni: basare il criterio di selezione sull'informazione relativa alla *stance* oppure considerare la natura delle relazioni già definite all'interno dei corpora di interesse, ovvero legami di tipo supporto tra *evidence* e *claim*, per effettuare opportune considerazioni in merito.

- Definizione di opportuni classificatori per l'attività di *argument structure prediction*, ognuno distinto in termini di *feature* e algoritmo di classificazione. In particolar modo, oltre alla definizione di semplici *baseline* di riferimento, si sperimenta l'uso di metodologie già precedentemente testate in un altro scenario applicativo relativo sempre alla *argument structure prediction*, di tecniche specifiche per la *stance classification* e infine di approcci sub-simbolici quali le reti neurali ricorrenti.

- Valutazione delle informazioni relative alla *stance* in qualità di *feature* aggiuntiva per la *argument structure prediction*. Più precisamente, si vuole valutare se l'inserimento delle presenti informazioni può contribuire positivamente alla fase di classificazione.

Vediamo dunque di approfondire in dettaglio ciascuno degli obiettivi elencati nelle sezioni successive del presente capitolo.

4.2 Il problema dell'assenza di esempi negativi per la classificazione

La prima tematica da affrontare e di notevole importanza riguarda direttamente la natura dei data-set a disposizione, ovvero **CE-ACL-14** e **CE-EMNLP-15**. In particolar modo, al fine di poter intraprendere un'attività di *argument structure prediction* è necessario definire un classificatore atto a distinguere tra coppie *evidence - claim* di tipo supporto dalle altre. Tuttavia, riprendendo brevemente il discorso introduttivo sui corpora di interesse, questi ultimi fanno riferimento solamente a legami del primo tipo, i.e. *supporting evidence - claim*. A tal proposito, la presente problematica costituisce un ostacolo di notevole importanza e al tempo stesso di non facile risoluzione, il quale influenza in maniera significativa i risultati sperimentali proposti dal presente elaborato. Infatti, la non disponibilità di opportuni *corpora* di larga scala e non soggetti a forti limitazioni per l'*argument structure prediction*, introduce il bisogno di dover delineare un processo di costruzione o estrazione dei dati mancanti. Chiaramente, viste le limitate risorse a disposizione non è possibile considerare un processo di annotazione manuale e risulta quindi doveroso indicare soluzioni alternative. Pertanto, come necessaria conseguenza, è di importanza primaria delineare una strategia atta a poter definire l'insieme degli esempi negativi, ovvero coppie *opposing evidence - claim*, a partire dai dati a disposizione. Un primo approccio di facile intuizione basa il proprio ragionamento sull'utilizzo delle informazioni relative alla *stance classification*, introdotte nel capitolo precedente. Più precisamente, considerando come *target* di riferimento il *topic* o l'articolo di tipo giornalistico, definiti all'interno dei due corpora di interesse, si può usufruire delle indicazioni date dalla *stance* per individuare coppie *opposing evidence - claim*. Nello specifico, si ricercano eventuali coppie aventi *stance* fortemente contrastanti e nel caso della classificazione sul data-set Emergent, esse sono delineate dalle classi `for` e `against`. In particolare, dato il punto arbitrario di scelta nell'ambito della definizione del *target* di riferimento e l'ipotesi relazionale riguardante le coppie *evidence - claim* contenute all'interno dei corpora di interesse, si individuano i seguenti scenari:

- **Stance claim - article**: si considera l'articolo di giornale come *tar-*

get di riferimento per la *stance*. A tal proposito, la presente ipotesi ha come prerequisito la disponibilità delle informazioni relative alla *stance classification* per il presente *target* di classificazione. Successivamente, si considerano come coppie *opposing evidence - claim* tutte quelle aventi *stance* relative all'articolo contrastanti. In altre parole, si raggruppano i dati contenuti all'interno dei corpora di interesse in base all'articolo di tipo giornalistico da cui sono stati estratti. Infine, si considerano di volta in volta due coppie *evidence - claim* e si osservano i due valori della *stance* relativi ad entrambi i *claim*: se i valori sono fortemente contrastanti allora si costruiscono due coppie *opposing evidence - claim* incrociando i dati a disposizione.

- **Stance claim - topic**: ragionamento analogo al punto precedente con l'unica differenza che come *target* di riferimento si considera il *topic* e non più l'articolo.

- **Concatenated stances**: si considerano gli stessi elementi dello scenario *Stance claim - article* con l'aggiunta del valore della *stance* tra *topic* e l'articolo di tipo giornalistico, definendo nello specifico delle opportune espressioni booleane volte a descrivere il vincolo di contemporaneità. Anche in questo caso, la strategia di definizione delle coppie *opposing evidence - claim* si basa sul prerequisito di aver precedentemente estratto le presenti informazioni di interesse. In altre parole, le due *stance* relative ad una determinata coppia *evidence - claim* sono mappate nei loro rispettivi valori booleani: ad esempio, `for` diventa `True`, mentre `against` viene considerato come `False`. Successivamente, si valutano i risultati dell'applicazione del vincolo di contemporaneità, ovvero un *AND* logico, per due coppie *evidence - claim* prese in esame: se i valori delle espressioni booleane sono contrastanti allora si possono costruire le coppie *opposing evidence - claim* sempre incrociando i dati a disposizione.

4.2.1 Stance claim - article

Si parte dal presupposto che vi siano più coppie *evidence - claim* estratte da un singolo articolo. Inizialmente, vengono filtrati tutti gli articoli, selezionando solamente quelli contenenti almeno una coppia avente *stance*

positiva e un'altra con *stance* negativa. Esempi di *stance* positive possono essere: `agree`, `for`, `pro`. Viceversa, per *stance* negativa solitamente si intendono i seguenti valori: `disagree`, `against`, `con`. In particolare, nel caso del corpus Emergent i valori di riferimento sono `for` e `against`. Successivamente, per ognuno degli articoli che hanno superato con successo la fase di filtraggio, si considerano tutte le possibili coppie e si vanno a selezionare solamente quelle aventi *stance* non uguali. Infine, per queste ultime, si costruiscono le due coppie *evidence* - *claim*, considerando per ciascuno dei due *claim* la *evidence* appartenente all'altra coppia. Un esempio di quanto descritto è rappresentato dall'immagine sottostante (figura 4.1).

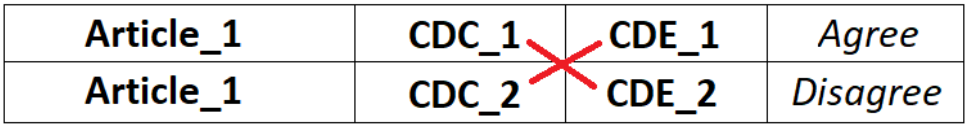


Figura 4.1: Estrazione delle coppie *opposing evidence* - *claim* nello scenario *Stance claim* - *article*.

4.2.2 Stance claim - topic

In maniera analoga a quanto descritto nella sezione 4.2.1, si considerano le coppie incrociate *evidence* - *claim*. La figura 4.2 riassume l'ultimo passo del procedimento.

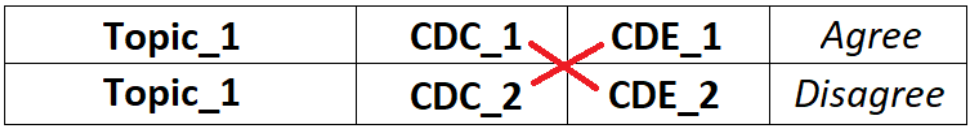


Figura 4.2: Estrazione delle coppie *opposing evidence* - *claim* nello scenario *Stance claim* - *topic*.

4.2.3 Concatenated stances

In quest'ultimo scenario si parte dal presupposto che la metodologia *Stance claim* - *article* scarti degli elementi che possano essere riutilizzati,

ovvero articoli aventi coppie *evidence* - *claim* appartenenti ad una stessa categoria di *stance*. A questo proposito, l'idea in questione consiste nell'individuare dapprima la *stance* di ciascuna coppia *topic* - *article* e per poi successivamente considerare queste ultime con le *stance* relative alle coppie *claim* - *article*. Più precisamente, per ogni *entry* nel *dataset* si valutano le rispettive due *stance* e si calcola il risultato della loro combinazione, i.e. *AND* logico. Successivamente, si effettua un raggruppamento in base al *topic* e si considerano coppie di elementi. Nel caso in cui le *stance* risultanti di ciascun elemento della coppia siano diverse tra di loro, si possono considerare le coppie *evidence* - *claim* incrociate come descritto nelle sezioni 4.2.1 e 4.2.2. La figura 4.3 descrive il procedimento sopra citato. Infine, la presente metodologia può dunque potenzialmente incrementare il numero di coppie *opposing evidence* - *claim* rispetto al singolo scenario *Stance claim* - *article*.

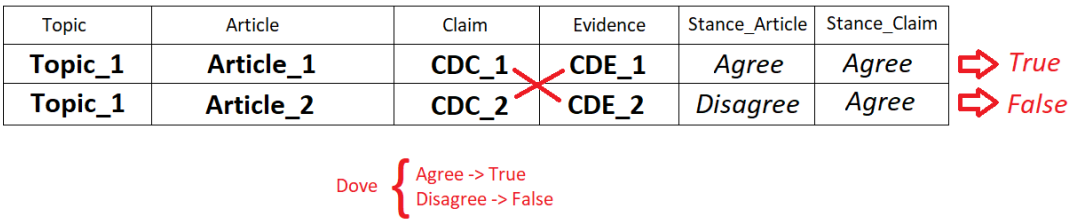


Figura 4.3: Estrazione delle coppie *opposing evidence* - *claim* nello scenario *Concatenated stances*.

4.2.4 Problematiche e una valida alternativa

Tuttavia, la presente strategia di estrazione di coppie *opposing evidence* - *claim*, atte a rappresentare gli esempi negativi per la fase di classificazione, si basa su forti assunzioni e presenta al tempo stesso alcune debolezze. Più precisamente, come si può facilmente intuire, il fatto di basarsi esclusivamente sul valore della *stance* significa delineare la strategia sulle caratteristiche e performance del classificatore scelto per l'attività di *stance classification*. In altre parole, occorre tener conto della natura delle informazioni proposte dal presente strumento in quanto queste ultime possono presentare valutazioni errate. Nonostante si possa limitare il più possibile il numero di predizioni non corrette del classificatore, mediante per l'appunto la definizione di tecniche, *feature* e metodologie atte

a garantire performance ottimali, occorre sempre tenere conto della potenziale presenza di valori spuri, i.e. *outlier*. Di conseguenza, basare la propria strategia di selezione su informazioni, i.e. la *stance*, la cui validità non è mai totalmente garantita può compromettere la qualità dei dati costruiti per la *argument structure prediction*. In secondo luogo, la problematica appena evidenziata è ulteriormente aggravata se si effettua una considerazione in merito alla natura dei dati contenuti all'interno dei corpora di interesse, con accezione particolare agli articoli di tipo giornalistico. Più precisamente, essendo documenti originari di Wikipedia, ovvero l'enciclopedia libera costruita per mezzo di collaborazione di molteplici utenti, questi ultimi sono pertanto neutrali per definizione. Come conseguenza, non è garantita la presenza di vere opposizioni tra coppie *evidence* - *claim* nell'ambito di uno stesso articolo o *topic*. Pertanto, le coppie *opposing evidence* - *claim* estratte potrebbero non avere la stessa validità delle rispettive controparti manualmente identificate e descritte all'interno dei corpora di interesse. Infine, come naturale conseguenza occorre anche considerare l'eventuale possibilità che i dati ottenuti, i.e. le coppie *opposing evidence* - *claim*, non siano in quantità sufficienti tali da poter garantire la definizione di un'attività di *argument structure prediction* corretta. Quindi, sulla base delle presenti molteplici problematiche e limitazioni, si può considerare una diversa strategia di selezione, di gran lunga più semplice dal punto di vista concettuale e basata su un'osservazione di facile intuizione. Più precisamente, considerando l'unica ipotesi certamente verificata relativa al legame relazionale tra le coppie *evidence* - *claim* costituenti i corpora di interesse, si può dedurre che non sia possibile stabilire a priori alcuna tipologia di legame di tipo supporto per qualsiasi coppia incrociata considerata a partire dai dati di partenza. Pertanto, queste ultime possono essere tutte considerate come potenziali esempi negativi per la fase di classificazione, nonostante le problematiche evidenziate nell'ambito della natura dei dati di interesse. Nello specifico, poiché il *topic* rappresenta un riferimento di maggiore importanza in termini semantici rispetto all'articolo di tipo giornalistico da cui sono state estratte le coppie *text*, si è deciso di condizionare il processo di costruzione delle coppie *opposing evidence* - *claim* in base al raggruppamento per *topic* dei dati forniti dai corpora di interesse. In altre parole, gli esempi negativi vengono valutati nell'ottica di un *topic* comune. Tuttavia, anche la presente strategia di selezione dei dati

presenta un principale problema: il fatto di basarsi su un simile criterio di estrazione degli esempi negativi comporta inevitabilmente alla definizione di corpora significativamente sbilanciati in favore di questi ultimi, aspetto che può sicuramente rappresentare un ulteriore ostacolo per l'attività di *argument structure prediction*, come verrà sottolineato più volte successivamente all'interno del presente capitolo.

4.3 La *stance* come strumento d'ausilio per la *argument structure prediction*

Al termine di questa fase preliminare di costruzione di nuove informazioni, utilizzabili per la *argument structure prediction*, vengono effettuate alcune ricerche sulla struttura dei dati, basate esclusivamente sui valori delle *stance* associate a ciascuna coppia. In particolar modo, l'obiettivo imposto verte essenzialmente sulla possibilità o meno di usufruire delle informazioni relative alla *stance* per contribuire positivamente alla fase di classificazione nell'ambito dell'*argument structure prediction*. In altre parole, si vuole valutare se la *stance* possa essere usata in qualità di *feature* aggiuntiva per il processo di discriminazione tra esempi positivi e negativi. A tal fine, si ricorre alla definizione di due importanti indicatori atti a consentire la valutazione dell'attuabilità dell'idea precedentemente introdotta. Nello specifico, si individua il numero di coppie incrociate aventi *stance* opposta e stesso *topic* e si costruiscono le matrici di confusione relative alla *stance* per poter valutare se si possa usufruire di tale informazione aggiuntiva durante la fase di classificazione. Poiché il data-set Emergent prevede l'assegnazione di una terza classe, ovvero `observing`, risulta necessario tenere conto di tale fattore all'interno delle statistiche calcolate per ciascun data-set di interesse, in quanto quest'ultima rappresenta fondamentalmente una *wild card*. Più precisamente, le informazioni raccolte sono soggette a diversi filtri di gestione relativi alla classe `observing`. Pertanto, sulla base di tali criteri di selezione, si raccolgono le informazioni in esame su entrambi i data-set di interesse (tabelle 4.1 e 4.9). Procedendo con ordine, nel calcolo delle coppie incrociate sono state definite le seguenti modalità di gestione:

- **Default**: si considerano tutte le coppie aventi *stance* diversa.

- **Ignore-observing**: si considerano solamente le coppie aventi *stance* differente e non uguale ad `observing`.

Data-set	default	ignore-observing	no-filter
CE-ACL-14	17658	5785	31447
CE-EMNLP-15	63826	14288	113631

Tabella 4.1: Numero coppie estratte in base alle modalità di interpretazione della classe *observing*. Procedendo da sinistra verso destra, i campi *default* e *ignore-observing* fanno riferimento alle omonime strategie di selezione precedentemente indicate. Infine, la colonna *no-filter* riporta invece il numero totale di coppie incrociate senza l'applicazione di alcun filtro di selezione.

Analogamente a quanto definito nell'ambito del conteggio delle coppie incrociate negative, occorre stabilire una politica di gestione della classe `observing`. Per tale obiettivo, sono state definite le seguenti modalità di interpretazione:

- **Default**: le coppie aventi *stance* identica sono considerate come esempi positivi, ovvero sussiste un legame di tipo supporto tra *evidence* e *claim*.

- **Overcome-observing**: in aggiunta alla semplice uguaglianza tra *stance* si considerano come esempi positivi anche le coppie aventi una delle due *stance* pari ad `observing`.

- **Ignore-observing**: si escludono dagli esempi positivi tutte le coppie aventi almeno uno dei due valori relativi alla *stance* uguale ad `observing`.

Più precisamente, la tabella 4.2 riporta la tipologia di coppie individuate come esempi positivi per ciascuna delle politiche di gestione della classe `observing` sopraelencate.

Modalità di selezione	Tipologia coppie selezionate come esempi positivi	Tipologia coppie selezionate come esempi negativi
default	for-for, against-against, observing-observing	for-against, for-observing, against-observing,
overcome-observing	for-for, for-observing, against-against, against-observing, observing-observing	for-against
ignore-observing	for-for, against-against	for-against, for-observing, against-observing, observing-observing

Tabella 4.2: Definizione coppie positive e negative per ciascuna modalità di gestione della classe *observing*.

Nell'ambito della *argument structure prediction*, la *stance* predetta nei confronti di un dato *topic* potrebbe rappresentare una *feature* importante per poter stabilire i legami tra *claim* ed *evidence*. A tal fine, costruire una matrice di confusione può essere un indicatore significativo. Di seguito (tabella 4.9), sono riportate le matrici di confusione associate a ciascun data-set di interesse, suddivise in base alle modalità di interpretazione introdotte precedentemente. In ciascun scenario, gli esempi positivi veri sono rappresentati dalle coppie appartenenti ai corpora di interesse. Viceversa, tutte le coppie incrociate individuate nell'ambito di uno stesso *topic* sono trattate in qualità di esempi negativi. Purtroppo, come si può sempre osservare dalla tabella 4.9, i risultati ottenuti non sono soddisfacenti, in quanto i valori appartenenti alla diagonale principale di ciascuna matrice non rappresentano la maggioranza rispetto agli altri. Pertanto, ne consegue che l'utilizzo della *stance* come strumento per la *argument structure prediction* non è sufficiente per poter distinguere correttamente la prevalenza degli esempi. Tuttavia, occorre sottolineare un aspetto di fondamentale importanza, rappresentato dalla distribuzione delle classi all'interno dei corpora di interesse. Come descritto ampiamente nella

precedente sezione, il forte sbilanciamento delle classi relative alla *argument structure prediction* condiziona negativamente anche la presente valutazione, in quanto non è possibile confermare con certezza la validità di tutte le coppie *evidence* - *claim* considerate come esempi negativi.

Tabella 4.3: CE-ACL-14, default

Predicted	link	no-link
True		
link	625	753
no-link	13789	17658

Tabella 4.4: CE-EMNLP-15, default

Predicted	link	no-link
True		
link	2499	2669
no-link	49805	63826

Tabella 4.5: CE-ACL-14, overcome-observing

Predicted	link	no-link
True		
link	1143	235
no-link	25662	5785

Tabella 4.6: CE-EMNLP-15, overcome-observing

Predicted	link	no-link
True		
link	4619	549
no-link	99343	14288

Tabella 4.7: CE-ACL-14, ignore-observing

Predicted	link	no-link
True		
link	514	864
no-link	11449	19998

Tabella 4.8: CE-EMNLP-15, ignore-observing

Predicted	link	no-link
True		
link	1846	3322
no-link	37595	76036

Tabella 4.9: Matrici di confusione suddivise per data-set e per modalità di gestione della classe *observing*.

4.4 Definizione di classificatori per la *argument structure prediction*

Una volta definito l'input per la *argument structure prediction*, si può procedere alla definizione dell'attività di classificazione, mediante l'introduzione di molteplici classificatori distinti. Inizialmente, si considerano set di *feature* molto limitati, in modo tale da costituire delle vere e proprie *baseline* di riferimento, per poi successivamente introdurre insiemi

ben più complessi. Nello specifico, il presente elaborato fa riferimento ai seguenti classificatori.

- *Baseline* basata su *n-gram*.
- *Baseline* basata su *cosine similarity*.
- Classificatore basato su un sottoinsieme delle *feature* introdotte da Stab & Gurevych.
- Classificatore basato su reti neurali ricorrenti (**RNN**).
- Classificatore basato su *feature* proprie della *stance classification*

In particolare, dal punto di vista sperimentale, per ognuno di essi è stato necessario ricorrere alla definizione di opportune ricerche esaustive, i.e. *gridsearch*, al fine di poter individuare i migliori parametri di configurazione relativi ad elementi quali le *feature* e gli *algoritmi* di classificazione. In aggiunta, analogamente a quanto effettuato per la *stance classification*, vengono considerate le seguenti classi di classificatore: `SGDClassifier`, `LogisticRegression` e infine `LinearSVC`. Per quanto riguarda le metriche, invece, si riportano le informazioni relative alla *accuracy* e al *f1-score*, in quanto fare affidamento esclusivamente alla prima metrica può rappresentare un punto di errore se si tiene in considerazione la distribuzione sbilanciata delle classi relative all'*argument structure prediction*.

4.4.1 Le *baseline*

Procedendo con ordine, i primi classificatori che introduciamo sono rappresentati dalle *baseline* relative agli *n-gram* e alla *cosine similarity* rispettivamente. Nello specifico, per quanto riguarda la fase di calibrazione, solamente la prima presenta dei parametri di configurazione relativi alle *feature*. Viceversa, entrambe sono interessate dalla ricerca del miglior algoritmo di classificazione in congiunzione con i rispettivi parametri di configurazione. Concludendo, le tabelle 4.12 e 4.15 riportano i risultati relativi alle azioni di ricerca precedentemente discusse per ciascuna delle due *baseline*.

Tabella 4.10: CE-ACL-14

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 1), max_features: 20000	0.7200	0.1316
Ngrams_second	n-gram range: (1, 1), max_features: 10000		
SGDClassifier	alpha: 0.001, loss: hinge, penalty: elasticnet	0.7987	0.1806

Tabella 4.11: CE-EMNLP-15

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 3), max_features: None	0.7079	0.1415
Ngrams_second	n-gram range: (1, 3), max_features: 10000		
SGDClassifier	alpha: 0.001, loss: hinge, penalty: elasticnet	0.8578	0.1504

Tabella 4.12: Risultati di calibrazione dei parametri associati alla *baseline* basata su *n-gram*. I suffissi *first* e *second* fanno riferimento ai *target* della classificazione. Nel caso di training su **CE-ACL-14** e **CE-EMNLP-15** sono *claim* ed *evidence* rispettivamente.

Tabella 4.13: CE-ACL-14

	Parametri	Accuracy	F1-score
Scelta classificatore		0.5782	0.1065
LogisticRegression	C: 10, penalty: L1	0.5785	0.1066

Tabella 4.14: CE-EMNLP-15

	Parametri	Accuracy	F1-score
Scelta classificatore		0.6998	0.1430
SGDClassifier	alpha: 1e-06, loss: log, penalty: L1	0.8700	0.1663

Tabella 4.15: Risultati calibrazione parametri associati alla *baseline* basata su *cosine similarity*.

4.4.2 Classificatore basato sulle *feature* introdotte da Stab & Gurevych

Seppur sia oggetto recente di interesse, l'*argument structure prediction* presenta già diversi lavori in ambito, volti ad individuare un legame tra coppie *evidence - claim*. Uno di questi é dato da "Identifying Argumentative Discourse Structures in Persuasive Essays" di Christian Stab e Iryna Gurevych[148], all'interno del quale, nonostante la notevole specificità del contesto di interesse, vengono introdotte numerose *feature* di applicabilità generale. A tal fine, risulta interessante individuare e implementare le presenti *feature* per poter successivamente confrontare i risultati della classificazione con quelli ottenuti dagli altri classificatori. Prima di poter procedere con l'introduzione delle *feature* delineate da Stab e Gurevych[148], occorre fornire alcune informazioni di contesto atte a poter comprendere meglio la terminologia utilizzata dai presenti autori. Nello specifico, lo scenario d'applicazione per l'attività di *argument structure prediction* riguarda i saggi di raccomandazione redatti da studenti, i.e. *persuasive essay*, caratterizzati, in particolar modo, da una struttura ben precisa. Dal punto di vista della classificazione sono state individuate come elementi di riferimento tutte le possibili coppie tra *premise*, i.e. *evidence*, *claim* e *major claim*, presenti all'interno dei saggi

oggetto di studio. Quest'ultimi vengono generalmente riferiti rispettivamente mediante i termini *source* e *target component*, mentre si ricorre all'uso dell'espressione *argument component* per indicarli in maniera generale. Procedendo con ordine, le *feature* introdotte da Stab e Gurevych sono le seguenti.

- **Structural features**: specifiche della struttura del testo preso in oggetto, i.e. *persuasive essays*. A tale categoria appartengono le seguenti *feature*.
  - numero di *token* per ciascun *target* della classificazione.
  - differenza assoluta del numero di *tokens*.
  - numero di *punctuation marks* per ciascun *target* della classificazione.
  - differenza assoluta del numero di *punctuation marks*.
  - *feature* basate sulla posizione di ciascun *target* della classificazione:
    - Due *feature* rappresentano la posizione delle frasi associate ai *target* della classificazione all'interno dei saggi.
    - Quattro *feature* booleane indicano se gli *argument component* sono presenti nella prima o nell'ultima frase di un paragrafo.
    - Una *feature* booleana per indicare se il *target component* sia collocato prima del *source component*.
    - Una *feature* per misurare la distanza tra le due frasi associate a ciascun *argument component*.
    - Una *feature* booleana per indicare se entrambi gli *argument component* appartengano alla stessa frase.
- **Lexical features**: relative a coppie di parole, i.e. *word pair*, le prime parole di ciascun *argument component*, i.e. *first word*, e ai verbi modali. A tale categoria appartengono le seguenti *feature*.
  - Tutte le possibili coppie di parole tra i due *argument component*.
  - La *first word feature* per ciascun *argument component*.

  - Coppie di *first word feature* per ciascuna coppia di *argument component*.
  - Una *feature* booleana per indicare se l'*argument component* in oggetto contenga o meno un verbo modale.
  - Conteggio dei termini comuni tra i due *argument component*.

- **Syntactic features**: rappresentate dall'estrazione delle *production rule*, individuate da un opportuno *parser* sintattico. In particolare, si estraggono le *production rule* per ciascun *argument component*. Ogni *production rule* estratta é modellata come una *feature* booleana.

- **Indicators**: si considera una lista specifica di *discourse marker*, estratta dal *Penn Discourse Treebank 2.0 Annotation Manual*[124]. Ogni *discourse marker* individuato viene trattato come una *feature* booleana per ciascun *argument component*.

- **Predicted type**: si utilizza l'*argumentative type* di ciascun *argument component*, ovvero la sua specifica tipologia: *premise*, *claim* o *major claim*. In particolare, si definiscono due *feature*, rappresentanti l'*argumentative type* per ciascuna coppia di *argument component*.

Come si può osservare, alcune delle *feature* elencate risultano essere fin troppo specifiche per poter avere applicabilità generale. Pertanto, solamente le *feature* evidenziate in rosso sono state successivamente oggetto di implementazione (appendice B). Inoltre, un set relativamente limitato di queste ultime, presenta parametri di configurazione, quale l'utilizzo o meno di *stop word* per filtrare inizialmente il testo da cui estrarre le *feature*. Per tal motivo, si é ritenuto appropriato effettuare una ricerca esaustiva, i.e. *gridsearch*, per individuare la miglior configurazione relativa ai parametri di ciascuna delle *feature* selezionate di interesse. La tabella 4.18 riporta le migliori configurazioni individuate per il classificatore basato sulle *feature* selezionate proposte da Stab e Gurevych[148].

Tabella 4.16: CE-ACL-14, Stab & Gurevych

	Parametri	Accuracy	F1-score
couple_first_word_binary	stop_words: english	0.9259	0.2203
first_word_binary_first	stop_words: None		
first_word_binary_second	stop_words: english		
word_pairs_binary	stop_words: english		
SGDClassifier	alpha: 1e-06, loss: hinge, penalty: elasticnet	0.9037	0.2197

Tabella 4.17: CE-EMNLP-15, Stab & Gurevych

	Parametri	Accuracy	F1-score
couple_first_word_binary	stop_words: None	0.9221	0.2896
first_word_binary_first	stop_words: None		
first_word_binary_second	stop_words: None		
word_pairs_binary	stop_words: None		
SGDClassifier	alpha: 1e-05, loss: log, penalty: elasticnet	0.9164	0.2912

Tabella 4.18: Risultati calibrazione parametri associati al classificatore basato sulle *feature* introdotte da Stab e Gurevych.

4.4.3 Classificatore basato su reti neurali ricorrenti

In aggiunta alle semplici *baseline* definite all'inizio della sezione, si è deciso di sperimentare l'utilizzo di un modello di classificazione basato su reti neurali. In particolare, l'architettura delineata all'interno del presente elaborato fa riferimento alle reti neurali ricorrenti (**RNN**), focalizzandosi nella fattispecie sull'implementazione *Long short term memory*[62] **RNN**, i.e. **LSTM**. Tuttavia, dal punto di vista implementativo, prima di poter sfruttare il potenziale di un livello **LSTM**, poichè l'input della classificazione é rappresentato da sequenze di testo, occorre effettuare una trasformazione dei dati appropriata. A tal fine, risulta necessario anteporre un livello di `Embedding`, il quale permette di passare da una semplice collezione di documenti testuali ad una sequenza di *word embedding*. Più precisamente, considerando il presente scenario di studio

specifico, ogni *claim* o *evidence* viene trasformato in un vettore di valori interi, ognuno rappresentante l'indice di una parola appartenente ad un dato vocabolario e presente all'interno del frammento di testo preso in esame. Tale rappresentazione, costituisce il vero e proprio input per il livello di `Embedding`. Successivamente, ad ognuna delle sequenze di interi date in ingresso ne viene associata un'altra in uscita costituita questa volta da valori reali, mediante l'ausilio della matrice di *word embeddings*. In altre parole, il livello di `Embedding` consente di mappare somiglianze semantiche tra parole in vicinanze spaziale tra vettori di numeri reali. La seguente figura 4.4 riporta il modello architetturale della rete neurale utilizzata come classificatore proposta all'interno del presente elaborato, mentre l'appendice B ne descrive la relativa implementazione. Poiché l'input è costituito da coppie *evidence* - *claim*, risulta necessario trattare le due sequenze separatamente, per poi ricongiungerle dopo i livelli **LSTM**. Infine, si può osservare come l'ultimo livello del modello sia caratterizzato da una funzione di attivazione sigmoidea, in quanto la natura della classificazione nell'ambito della *argument structure prediction* è binaria. Infine, un ulteriore possibile miglioramento del modello di rete neurale presentato è dato dall'inserimento di livelli di `Dropout`, in quanto generalmente consentono di incrementare l'efficacia della classificazione.

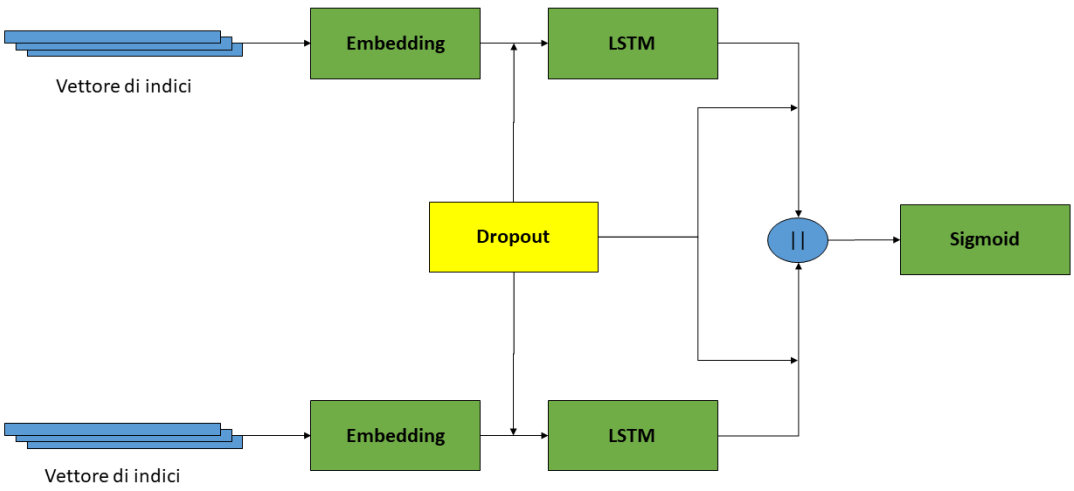


Figura 4.4: Modello rete neurale adottato per la classificazione.

Riprendendo il discorso fatto nella sezione 4.3, risulta interessante valutare l'efficacia o meno della *stance* relativa al *topic* in qualità di *feature* aggiuntiva. A tal fine, viene proposta una variante del modello di rete neurale introdotto precedentemente, la quale inserisce a ciascun elemento della sequenza il valore della *stance*. Più precisamente, occorre mappare ogni classe associata alla *stance* in un valore intero riservato all'interno del vocabolario associato al livello di `Embedding`. Successivamente, la trasformazione effettuata viene concatenata alla sequenza di indici relativa al *claim* o *evidence* a cui la *stance* fa riferimento. La seguente figura 4.5 descrive brevemente quanto enunciato precedentemente.

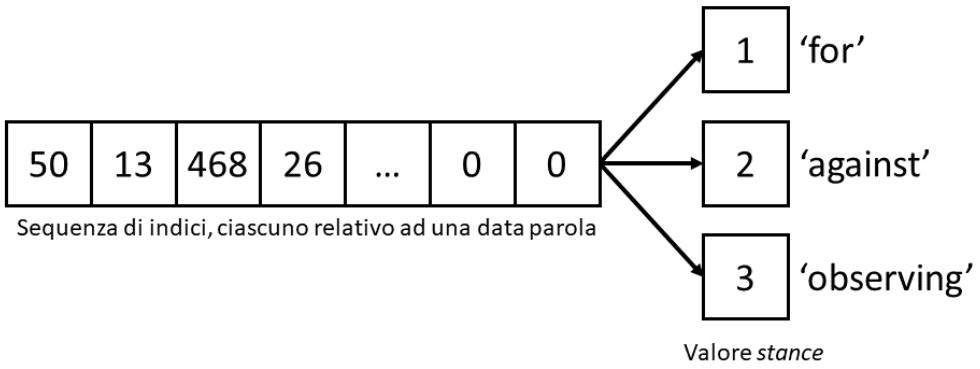


Figura 4.5: Concatenazione della *stance* relativa al *topic* a ciascun vettore di indici relativo ad un dato *claim* o *evidence*. I valori associati a ciascuna classe della *stance classification* appartengono all'intervallo [1..3].

E' importante sottolineare che l'input per il livello di `Embedding` deve avere lunghezza fissa. Pertanto, ogni vettore di indici è soggetto inizialmente ad una azione di *padding* per poi essere successivamente modificato mediante l'aggiunta dello specifico valore associato alla *stance*. Infine, per quanto riguarda la fase di calibrazione, per motivi prettamente legati alle limitate risorse computazionali, è stato considerata solamente la `batch_size` in qualità di unico parametro di configurazione (tabella 4.21).

Tabella 4.19: CE-ACL-14, rete neurale ricorrente

Batch_size	accuracy	f1-score
64	0.6881	0.1439
256	0.7265	0.1507
500	0.7576	0.1592
1000	**0.7808**	**0.1628**

Tabella 4.20: CE-EMNLP-15, rete neurale ricorrente

Batch_size	accuracy	f1-score
64	0.6803	0.1325
256	0.6640	0.1305
500	0.7038	0.1342
1000	**0.7204**	**0.1359**

Tabella 4.21: Risultati calibrazione parametri associati al classificatore basato su reti neurali ricorrenti.

4.4.4 Classificatore basato su *feature* proprie della *stance classification*

Come già sottolineato all'inizio del capitolo, il presente fa riferimento anche alla sperimentazione di tecniche e *feature* specifiche della *stance classification* nell'ambito della *argument structure prediction*, al fine di poterne dapprima valutare l'efficacia per poi successivamente effettuare alcune considerazioni in merito ad un loro potenziale impiego in altri contesti applicativi relativi sempre al settore di ricerca della *argument structure prediction*. Pertanto, si introduce un ulteriore classificatore basato sulla maggior parte delle *feature* introdotte all'interno del precedente capitolo. Nello specifico, a causa della sostanziale differenza in termini di dimensioni tra lo scenario considerato nell'ambito della *stance classification* e delle limitazioni in termini di risorse computazionali, è stato necessario ricorrere all'eliminazione di alcune *feature*, in particolare gli *skip n-gram*, al fine di poter affrontare in tempi ragionevoli tutte le operazioni di classificazione. Pertanto, poiché si fa riferimento a *feature* quali gli *n-gram*, occorre delineare anche per questo caso una fase preliminare di calibrazione dei parametri di configurazione (tabella 4.24).

Tabella 4.22: CE-ACL-14

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 3), max_features: None	0.7740	0.1411
Ngrams_second	n-gram range: (1, 3), max_features: 10000		
LogisticRegression	C: 0.001, penalty: l2	0.8161	0.1868

Tabella 4.23: CE-EMNLP-15

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 2), max_features: 20000	0.8166	0.2016
Ngrams_second	n-gram range: (1, 1), max_features: 20000		
SGDClassifier	alpha: 0.0001, loss: log, penalty: elasticnet	0.7742	0.2169

Tabella 4.24: Risultati di calibrazione dei parametri associati al classificatore basato sulle *feature* utilizzate nell'ambito della *stance classification*. I suffissi *first* e *second* fanno riferimento ai *target* della classificazione. Nel caso di training su **CE-ACL-14** e **CE-EMNLP-15** sono *claim* ed *evidence* rispettivamente.

4.4.5 Comparazione dei risultati

Una volta calibrati tutti i vari classificatori descritti precedentemente, è possibile procedere con la fase di misurazione delle loro performance sui data-set di interesse mediante un processi di *cross validation* con `n_splits=10`, tenendo conto di metriche quali *accuracy* e *f1-score*. In particolare, a partire dai dati riportati nella tabella 4.27, si possono effettuare due importanti osservazioni che si riallacciano alle problematiche evidenziate in merito ai corpora **CE-ACL-14** e **CE-EMNLP-15** in seguito alla fase di costruzione degli esempi negativi per l'attività di *argument structure prediction*. Inizialmente, dato il significativo sbilan-

ciamento delle due classi di dati, la metrica di riferimento *accuracy* risulta fuorviante. Più precisamente, poiché quest'ultima può essere brevemente definita come il rapporto del numero di predizioni corrette rispetto al totale degli esempi riportati e vista la forte predisposizione dei dati per la classe di tipo non supporto, ne risulta che anche una semplice *baseline* associante tutti gli esempi di interesse a quest'ultima classe è in grado di raggiungere valori elevati per la metrica *accuracy*. Pertanto, come strumento principale di misura delle performance dei molteplici classificatori, si fa affidamento alla metrica *f1-score*, relativa alla classe di tipo supporto. Così facendo, si è in grado di ottenere una stima accurata delle prestazioni di tutti i classificatori. Infine, l'altro aspetto di importanza cruciale che si vuole sottolineare riguarda proprio i risultati riportati per quest'ultima metrica di riferimento. Nello specifico, generalmente un classificatore si ritiene sufficientemente performante se ottiene un valore associato alla metrica *f1-score* pari o superiore all'incirca a 0.60. D'altro canto, tutti i classificatori proposti si collocano prevalentemente all'interno dell'intervallo $[0.10, 0.20]$, ad eccezione della *baseline* basata sulla *feature cosine similarity* nel caso del data-set **CE-EMNLP-15**. Come conseguenza, si può constatare come nessuno dei classificatori proposti sia in grado di discriminare correttamente la prevalenza degli esempi di classificazione riportati dai due corpora di interesse. A giustificazione della presente problematica, il forte sbilanciamento della distribuzione delle classi comporta un importante ostacolo per la fase di classificazione, non risolvibile con le presenti risorse a disposizione. Nonostante le problematiche evidenziate, si può comunque osservare sempre dalla tabella 4.27 come le *feature* introdotte da Stab e Gurevych[148] rispondano meglio delle altre all'attività di *argument structure prediction*.

Tabella 4.25: CE-ACL-14, cross validation

Classificatore	accuracy	f1-score
Baseline n-grams	0.8153	0.1848
Baseline cosine similarity	0.5788	0.1066
Stab & Gurevych	0.6281	**0.1954**
Rete neurale	0.7343	0.1535
Rete neurale (dropout = 0.2)	0.7055	0.1443
Rete neurale (stance)	0.6959	0.1410
Classificatore SC	**0.8157**	0.1884

Tabella 4.26: CE-EMNLP-15, cross validation

Classificatore	accuracy	f1-score
Baseline n-grams	**0.8661**	0.1528
Baseline cosine similarity	0.4704	0.088
Stab & Gurevych	0.8583	**0.2276**
Rete neurale	0.6907	0.1352
Rete neurale (dropout = 0.2)	0.6383	0.1260
Rete neurale (stance)	0.7104	0.1359
Classificatore SC	0.7655	0.1998

Tabella 4.27: Performance dei classificatori sui data-set di interesse. Per ognuno di essi sono riportate le metriche *accuracy* e *f1-score*.

Capitolo 5

Un nuovo corpus per la *argument structure prediction* mediante strumenti di *argument detection* quale MARGOT

I data-set finora presi in oggetto in ambiti quali la *stance classification* e l'*argument structure prediction* potrebbero non presentare dati a sufficienza tali da consentire l'individuazione di esempi negativi per la fase di classificazione, ovvero di coppie *opposing evidence-claim*. A tal fine, una potenziale valida alternativa potrebbe essere rappresentata dalla definizione di una nuova collezione di dati, i.e. corpus, basata su criteri di selezione fortemente legati alla natura argomentativa dei dati. In altre parole, si può pensare di usufruire di strumenti atti all'estrazione di *evidence* e *claim* all'interno di testi ricchi di argomentazioni, quali, ad esempio, quelli associati a siti di dibattito, al fine di poter poi successivamente costruire le relative coppie *evidence-claim*, costituenti un vero e proprio corpus. Chiaramente, tale metodologia *naïve*, in quanto tale, non garantisce la validità dei dati costruiti. Pertanto, occorre considerare che non tutti i frammenti di testo individuati come *evidence* o *claim*, rappresentino in realtà dei veri e propri argomenti. Inoltre, tale impossibilità di verifica si riflette anche sulla validità delle coppie di esempi positivi e negativi costruite, in quanto derivanti da processi di selezione regolati da criteri puramente soggettivi e basati sui dati a disposizione. Breve-

mente, una costruzione non manuale dei dati, non consente di agire sotto le stesse ipotesi forti relative al legame tra *evidence* e *claim* che invece un lavoro meticoloso di molteplici persone permette. Tuttavia, mediante l'effettuazione di opportune operazioni di confronto con i data-set di interesse, ovvero **CE-ACL-14** e **CE-EMNLP-15**, è possibile ottenere come risultato delle testimonianze della validità dei dati ottenuti. Ad esempio, una volta effettuata la fase di apprendimento di un dato classificatore, i.e. *training*, sul nuovo corpus in questione, si possono successivamente valutare le performance di quest'ultimo in merito alle coppie *evidence* - *claim* appartenenti ai data-set di ***IBM*** *Research*. Pertanto, nel caso in cui i risultati ottenuti rispecchino la natura dei dati contenuti in tali data-set, allora il presente esperimento può rappresentare una prova sufficiente per la convalidazione del corpus costruito. A tale proposito, il presente capitolo introduce e definisce il processo di costruzione di un nuovo data-set volto a risolvere alcune problematiche di fondo prettamente legate alla tipologia dei corpora presi in esame fino ad ora. Successivamente, si assiste alla sperimentazione degli stessi classificatori introdotti all'interno del capitolo precedente sul nuovo corpus appena definito, al fine di valutarne le caratteristiche in termini di corretta separazione delle classi. Di seguito, mediante la definizione di un opportuno test di comparazione, vengono misurate le performance dei precedenti classificatori sul corpus **CE-EMNLP-15**, con particolare attenzione a ciascun *topic* contenuto all'interno del presente data-set. Infine, si assiste alla sperimentazione del medesimo test di confronto da un punto di vista rovesciato, ossia valutando le prestazioni della stessa tipologia di classificatori sul nuovo data-set introdotto, una volta allenati sul corpus **CE-EMNLP-15**, al fine di fornire un'opportuna controprova di validazione dei risultati ottenuti durante il test iniziale.

5.1 Obiettivo

Il presente capitolo si pone come obiettivo primario la definizione di un nuovo corpus per l'*argument structure prediction*, caratterizzato, in particolare, da coppie *evidence* - *claim* individuate per mezzo di strumenti di *argument detection*, in grado, quindi, di estrarre elementi quali *claim* ed *evidence* a partire da un processo di elaborazione del testo. A tal

fine, si delineano le seguenti fasi operazionali volte a definire i molteplici requisiti necessari per il raggiungimento del presente obiettivo.

- Definizione del contesto di interesse all'interno del quale individuare i documenti testuali da analizzare, contenenti potenziali relazioni di confronto ideologiche.
- Introduzione di opportuni strumenti di estrazione degli argomenti, i.e. *claim* ed *evidence*.
- Formulazione dei criteri di selezione volti a definire le regole per mezzo delle quali i *claim* e le *evidence* individuati sono organizzati in coppie in accordo con le due classi di riferimento per l'attività di *argument structure prediction*, ovvero supporto e non.

Infine, una volta portato a termine il processo di costruzione del nuovo corpus, si procede con la sperimentazione di vari classificatori, effettuando successivamente un opportuno test di comparazione delle loro performance sul data-set **CE-EMNLP-15** preso in esame.

5.2 Processo di costruzione del nuovo corpus

Prima di poter procedere con la fase di determinazione delle coppie *evidence - claim*, in accordo con le due classi relative all'attività di *argument structure prediction*, occorre dapprima soddisfare due importanti requisiti: la determinazione del contesto di applicazione e gli strumenti di estrazione degli argomenti. Procedendo con ordine, per quanto riguarda lo scenario di interesse, nell'ambito del quale si ricercano esempi di *claim* ed *evidence* caratterizzati da relazioni di forte contrasto o supporto, si considera il data-set denominato all'interno del capitolo 1 come *Create Debate Custom Dataset*, introdotto da Hasan e Ng[60]. Più precisamente, quest'ultimo offre 4,902 *post* relativi a scenari di dibattito online vertenti su quattro principali *topic* di discussione: l'ex presidente degli Stati Uniti d'America Barack Obama (*Obama*), la legalizzazione o meno della marijuana (*Marijuana*), l'aborto (*Abortion*) e infine i diritti degli omosessuali (*Gay Rights*). Nello specifico, ogni singolo documento testuale è contornato da alcune informazioni aggiuntive di contesto, quali il tipo di relazione nei confronti del *post* che lo precede all'interno della medesima

discussione, i.e. *rebuttal*, e il valore della *stance* nei confronti del *topic* di riferimento. In particolare, questi ultimi meta-dati rappresentano un'informazione molto utile in quanto costituisce la base per la definizione dei criteri di selezione volti a definire le coppie *evidence - claim*. Successivamente, dal punto di vista sperimentale, si ricorre allo specifico strumento di *argument detection* denominato *Mining ARGuments frOm Text* (**MARGOT**), introdotto da Lippi e Torroni[84], il quale brevemente consente di individuare *claim* ed *evidence* sulla base di informazioni non strettamente legate dal contesto specifico, i.e. *context-independent claim/evidence detection*. Pertanto, una volta definiti gli strumenti e il contesto di interesse per la definizione del processo di costruzione di un nuovo corpus, si può procedere con la formulazione di opportune regole di abbinamento degli argomenti estratti tali da definire delle coppie *evidence - claim* di tipo supporto e non, impiegate successivamente in qualità di input per l'attività di *argument structure prediction*. In particolare, quanto appena descritto può essere facilmente riassunto per punti come segue:

- **Estrazione *evidence* e *claim***: i dati sono suddivisi in quattro domini, relativi ai *topic* di interesse: *Obama*, *Marijuana*, *Gay rights* e infine *Abortion*. Ogni singolo file di testo viene dato in input a **MARGOT**, il quale produce come risultato un documento testuale contenente l'analisi di ogni frase individuata nel testo. E' opportuno sottolineare che **MARGOT** può etichettare un argomento in qualità di *evidence* e *claim* contemporaneamente, i.e. *claim_evidence* (figura 5.1). Pertanto, tali elementi particolari, vengono poi considerati distintamente sia come *evidence* che come *claim* nella fase successiva di costruzione delle coppie.

- **Costruzione esempi positivi e negativi**: una volta terminata la fase di *argumentation mining*, si procede con la costruzione delle coppie *evidence-claim*. A tal fine, si ricorre all'ausilio di meta-dati significativi associati a ciascun file presente nel data-set, quale il *rebuttal*, relativo a coppie adiacenti di post appartenenti ad una stessa discussione. Più precisamente, la presente informazione indica per una data coppia di *post* se l'ultimo dei due, in ordine temporale, supporta, si oppone o è neutrale nei confronti dell'altro. Nello specifico, il campo *rebuttal* può assumere i seguenti valori: `support`,

`oppose` e `null`. Infine, per quanto riguarda il criterio di selezione degli esempi positivi e negativi, quest'ultimo può essere riassunto nel seguente modo.

- **Selezione esempi positivi**: tra tutti i file analizzati, vengono considerati solamente quelli contenenti almeno una *evidence* e una *claim*. Successivamente, per ognuno di questi si considerano tutte le possibili combinazioni tra le *evidence* e i claim, i.e. *evidence* - *claim*. In aggiunta, si esaminano anche le coppie di file aventi `rebuttal=support` per poter in costruire in seguito tutte le possibili coppie *evidence* - *claim* incrociate.
- **Selezione esempi negativi**: analogamente per quanto visto per gli esempi positivi, si considerano coppie di file aventi `rebuttal=oppose` e si costruiscono tutte le possibili coppie *evidence* - *claim* incrociate.

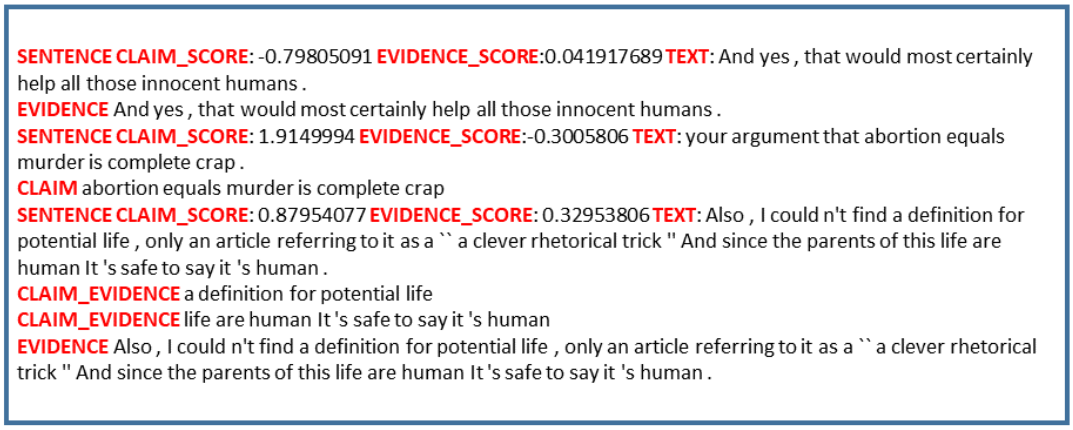


Figura 5.1: Esempio di elaborazione del testo da parte di **MARGOT**. Il frammento testuale preso in analisi appartiene alla conversazione con identificativo **A** nell'ambito del *topic* relativo all'aborto (*Abortion*). In particolare, le frasi contenenti *evidence*, *claim* o *claim_evidence* sono etichettate da opportune *keyword*: *EVIDENCE*, *CLAIM* e *CLAIM_EVIDENCE*.

Innanzitutto, risulta opportuno soffermarsi sul criterio di selezione delle coppie *evidence* - *claim*. In particolare, usufruire di informazioni

aggiuntive quale il *rebuttal* può aiutare ad individuare coppie logicamente corrette di esempi positivi e negativi. Viceversa, non è detto che l'ipotesi di etichettare come esempi positivi le coppie *evidence* - *claim*, contenute in uno stesso file possa risultare egualmente corretta. In aggiunta, si potrebbero sperimentare anche ulteriori criteri di selezione basati, ad esempio, su informazioni quale la *stance* associata a ciascun file e relativa al *topic* di riferimento. Una volta ottenuti i dati, prima di poter passare alla sperimentazione dei vari classificatori, è opportuno effettuare alcune operazioni di pulizia e bilanciamento delle classi, in quanto si ritiene che possano agevolare la successiva fase di classificazione. Nello specifico, la prima fase verte sull'eliminazione di coppie aventi testi molto brevi, mentre la seconda seleziona solamente le coppie più corpose in termini di parole. Più precisamente, poiché il corpus ottenuto verte in maniera significativa in favore degli esempi positivi, al fine di non compromettere la fase di classificazione, vengono selezionati gli stessi numeri di esempi positivi e negativi. La seguente tabella 5.1 presenta le statistiche del nuovo corpus generato, riassumendo, in particolare, le precedenti fasi di elaborazione preliminare dei dati.

	# Esempi Positivi	# Esempi Negativi
Originale	130824	115122
Pulizia	126972	111804
Bilanciamento	30000	30000

Tabella 5.1: Statistiche corpus costruito. Nella prima riga, etichettata dalla voce 'Originale' sono riportati il numero di esempi positivi e negativi ottenuti seguendo i criteri di selezione descritti precedentemente. Successivamente, alle voci 'Pulizia' e 'Bilanciamento' sono associati il numero di esempi positivi e negativi considerati al termine delle analoghe fasi appena introdotte.

5.3 *Mining ARGuments frOm Text* (MARGOT)

Poiché rappresenta lo strumento di riferimento per la definizione del nuovo corpus, occorre soffermarsi maggiormente in dettaglio su **MARGOT**, al fine di comprenderne con cura il funzionamento e le metodologie sulle quali si basa. La grande necessità di strumenti atti a consentire ad un

utente generico la possibilità di poter interagire con informazioni proprie dell'*argumentation mining* in maniera semplice, diretta e senza richiedere all'utente alcuna conoscenza nel presente campo di ricerca, ha portato Lippi e Torroni[84] alla definizione di uno strumento in grado di poter risolvere accuratamente tale problematica. Nello specifico, l'attuale stato dell'arte dell*argumentation mining*, come precedentemente sottolineato all'interno del capitolo 2, offre un'ampia scelta di metodologie e tecniche relative ad una specifica attività del presente settore di ricerca, quale, ad esempio, l'individuazione di relazioni di attacco o supporto tra argomenti o la *claim detection*. Tuttavia, mancano strumenti per l'estrazione degli argomenti, a partire da documenti testuali non strutturati, disponibili e utilizzabili da un'ampia comunità di utenti non necessariamente appartenenti al contesto scientifico. Più precisamente, a motivazione di quanto osservato si può considerare la giovane età dell'*argumentation mining*, ma soprattutto la difficoltà di definizione del concetto stesso di argomento in termini assoluti, problematica dalla quale derivano soluzioni di *argumentation mining* tipicamente legate ad un singolo genere, difficilmente adatto per uno scopo generale. Pertanto, nel presente contesto descritto, dove una delle sfide principali è rappresentata per l'appunto dall'estrazione automatica di argomenti strutturati a partire da testi di vario genere, viene definito lo strumento *Mining ARGuments frOm text* (**MARGOT**), definito come il primo sistema online di *argumentation mining* progettato per raggiungere una vasta fascia di utenti al di fuori della comunità di ricerca.

5.3.1 Definizione e struttura

In particolare, **MARGOT** può essere definito come un sistema web, basato su tecniche specifiche di *argumentation mining* proprie dello stato dell'arte, il cui obiettivo consiste nel poter offrire ad un utente generico la possibilità di interagire con le tecnologie proprie del presente settore di ricerca, senza richiedere alcuna conoscenza a priori nell'ambito lato utente. Inoltre, dal punto di vista delle funzionalità, **MARGOT** estende il lavoro di Lippi e Torroni[82] sulla *context-independent claim detection*, introducendo ulteriori attività quali la *context-independent premise detection* e la *argument component boundary detection*, basandosi sull'osservazione che le frasi argomentative sono spesso caratterizzate da strutture sintattiche

comuni. Pertanto, si individuano le seguenti due fasi principali (figura 5.2): (1) l'individuazione delle frasi argomentative, ovvero contenenti almeno un componente, i.e. *claim* o *evidence*, e (2) successivamente la determinazione dei confini sintattici di ciascun componente. A tal fine, occorre precisare che **MARGOT** si basa esclusivamente sul modello argomentativo di Walton[162], con particolare accezione alle definizioni di *claim* e *premise*, i.e. *evidence*, definite da Aharoni et al.[4] durante la presentazione del corpus di ***IBM** Research*. Pertanto, seppur **MARGOT** ricerchi *claim* ed *evidence* senza basarsi su informazioni a priori relative al *topic*, risulta importante sottolineare i concetti di interesse estratti, in quanto essi rappresentano i dati descritti precedentemente all'interno del processo di costruzione del nuovo corpus.

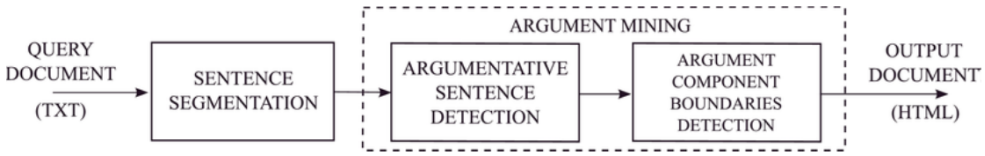


Figura 5.2: Modello di elaborazione del testo definito da **MARGOT**. La presente immagine è cortesia di Lippi e Torroni[84].

5.3.2 Procedura di elaborazione del testo

Una volta definite le tematiche inerenti all'introduzione e definizione dello strumento **MARGOT**, risulta opportuno delinearne il funzionamento dal punto di vista operazionale. Innanzitutto, viene fornito come input un documento testuale al web server, il quale è soggeto ad un'opportuna fase di elaborazione da parte dello strumento di analisi *Stanford Parser*[88], in grado, tra le molteplici funzionalità, di suddividere il documento in frasi e di costruire per ognuna di esse il rispettivo albero di analisi delle dipendenze sintattiche, i.e. *costituency parse tree* (figura 5.3). Successivamente, ogni frase individuata è elaborata da due specifici classificatori, basati sul concetto di *Tree Kernel*, in grado rispettivamente di discriminare quelle contenenti dei *claim* o delle *evidence* dalle rimanenti. Nello specifico, entrambi i classificatori ricorrono all'ausilio del *costituency parse tree* e dei **BoW** per rappresentare l'input, al fine di poter produrre come risultato un punteggio indicante il loro grado di con-

fidenza che la frase analizzata contenga o meno un *claim* o una *evidence* rispettivamente. In seguito, per ogni frase ritenuta argomentativa viene applicato il modulo relativo all'attività di *argument component boundary detection*, formulato nello specifico come un processo di *sequence labeling*, così da poter identificare i confini sintattici di tutti i *claim* ed *evidence* individuate. Più precisamente, dal punto di vista tecnico, si ricorre all'ausilio di *n-gram*, annotazioni **POS**, lemmi e le cosiddette *named-entity* generate dallo strumento *Stanford CoreNLP*. Infine, i risultati ottenuti vengono poi mostrati all'utente sotto forma di pagina **HTML** (figura 5.4), dove i componenti individuati sono evidenziati in grassetto, nel caso di *claim*, in corsivo, nel caso di *evidence* e in entrambi i modi se un frammento di testo è stato ritenuto come argomentativo da entrambi i classificatori, i.e. *claim_ evidence*.

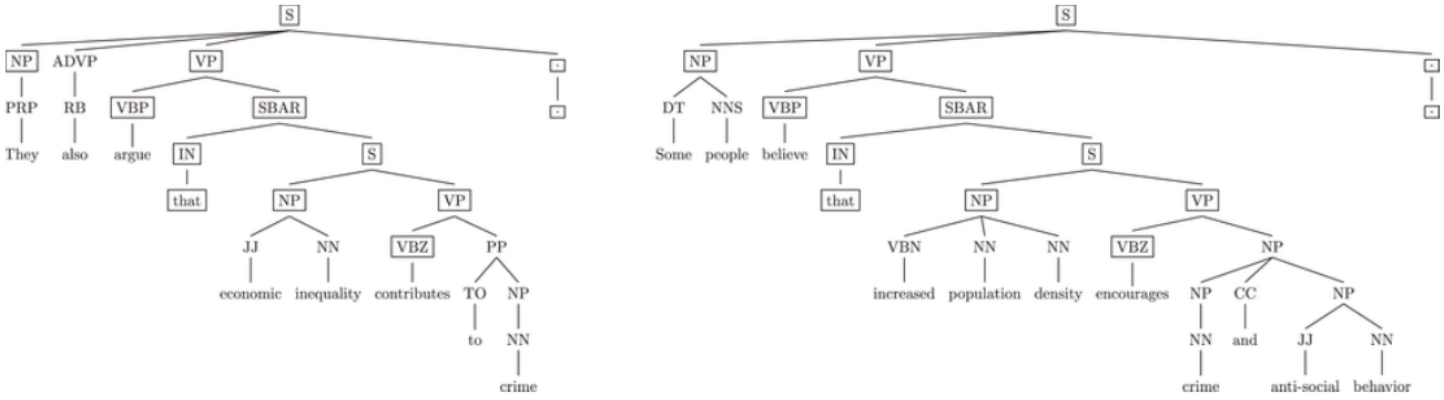


Figura 5.3: Esempi di *costituency parse tree* ottenuti in seguito all'analisi di due *claim* appartenenti al corpus **CE-EMNLP-15**. In particolare, i nodi contornati da un riquadro rappresentano gli elementi in comune tra i due alberi. La presente immagine è cortesia di Lippi e Torroni[84].

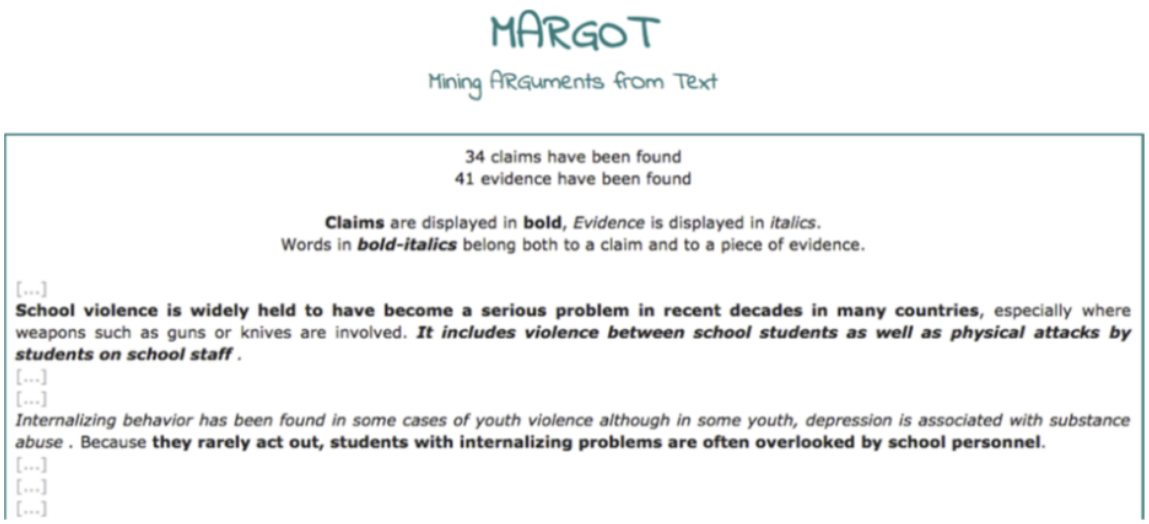


Figura 5.4: Esempio di risultato proposto da **MARGOT** al termine del processo di elaborazione e analisi del testo dato. La presente immagine è cortesia di Lippi e Torroni[84].

5.3.3 Tecniche

Come è già stato precedentemente accennato, **MARGOT** basa il proprio funzionamento su un classificatore **SVM** definito secondo l'applicazione di un *Tree Kernel*. In particolare, quest'ultima tipologia di metodo basato su *Kernel* è stata ampiamente utilizzata in una varietà di differenti problemi di **NLP**, spaziando, ad esempio, dalla *text categorization* fino ad attività ben più specifiche quali il *semantic role labeling*, la *relation extraction*, la *named entity recognition*, la *question/answer classification* e altre ancora. Nello specifico, i *Tree Kernel* sono stati impiegati con successo in molte applicazioni[103][104]. Più precisamente, questi ultimi sono progettati per misurare la similarità tra due alberi, mediante la valutazione del numero di sotto-strutture in comune, denominati tipicamente come frammenti, i.e. *fragment*. A tal fine, considerando molteplici possibili definizioni di *fragment*, si introducono diverse funzioni relative alla delineazione dei *Tree Kernel* (**TK**). In particolare, si distingue principalmente tra le seguenti tipologie:

- **SubTree Kernel (STK)**[157]: il *fragment* può coincidere con qualsiasi nodo dell'albero, inclusi tutti i discendenti, ossia i sotto-alberi.

- **SubSet Tree Kernel (SSTK)**[31]: concetto più generico del **STK** dato che le foglie di un *fragment* possono essere anche dei simboli non terminali, purché non violino i vincoli delle regole grammaticali.

- **Partial Tree Kernel (PTK)**[102]: rappresenta il *Kernel* più generale tra tutti, dove i *fragment* possono essere qualsiasi prozione di un sotto-albero a partire dal nodo inizialmente considerato.

Nello specifico, **MARGOT** impiega il **SSTK** a differenza del **PTK** impiegato da Lippi e Torroni[82]. Il motivo è dato dal fatto che a livello sperimentale le performance di entrambi sono molto simili tra loro, con la distinzione che il *Kernel* **SSTK** risulta essere di gran lunga più efficiente durante entrambe le fasi di *training* e di *classificazione*. Pertanto, nell'ottica di un web server è stato ritenuto preferibile fare affidamento sul **SSTK**. Tuttavia, è opportuno sottolineare come all'interno del presente elaborato si faccia riferimento ad una versione per uso locale[1] di **MARGOT**, interrogabile da linea di comando, in modo tale da semplificare maggiormente la fase di utilizzo del presente strumento di elaborazione testuale.

5.4 Valutazione corpus

Terminata la fase di rifinimento del corpus costruito, si può procedere con la sperimentazione dei molteplici classificatori introdotti all'interno del capitolo precedente, al fine di poter valutare, innanzitutto, l'efficacia delle *feature* utilizzate in precedenza sempre nell'ambito della *argument structure prediction* e infine poter verificare la validità dei dati costruiti. Analogamente a quanto osservato per i data-set proposti da ***IBM Research***, i.e **CE-ACL-14** e **CE-EMNLP-15**, anche in questo scenario è stata richiesta una fase preliminare di calibrazione dei parametri

[1] Disponibile solamente su richiesta.

associati a ciascuna *feature* e tipologia di algoritmo di classificazione impiegato (tabelle 5.2, 5.3, 5.4, 5.5 e 5.6). Successivamente, si procede con la fase di apprendimento e verifica dei classificatori, i.e. *training*, mediante la tecnica di *cross validation*, caratterizzata dai seguenti parametri: `n_splits=10` e `shuffle=True`. Inoltre, come metriche di giudizio si fa sempre riferimento alla *accuracy* e al *f1-score*. Più precisamente, come si può osservare dalla tabella 5.7, ad eccezione della *baseline* basata sulla *feature cosine similarity*, i risultati ottenuti sono piuttosto soddisfacenti, raggiungendo, nello specifico, un valore massimo per la metrica *f1-score* pari a 0.8674 nel caso della *baseline* relativa agli *n-gram*, seguita dal classificatore basato sulle *feature* principali della *stance classification*. In particolare, quest'ultimo dato sembra trovare un'analogia con quanto osservato nell'ambito della *stance classification*, dove molto spesso classificatori basati su *feature* relativamente semplici risultano essere difficili da battere nella prevalenza dei contesti di applicazione di interesse[144][60][98]. Tuttavia, nonostante l'apparente successo dal punto di vista sperimentale, il presente strumento di misurazione delle performance, i.e. *cross validation*, è in grado di dimostrare solamente la marcata differenza in termini sintattici e semantici, nell'ottica delle capacità di astrazione delle *feature* selezionate, che sussiste tra le diverse classi di coppie *evidence* - *claim* individuate. Pertanto, se si vuole verificare la validità dal punto di vista argomentativo, nel rispetto anche dello strumento di *argument detection* usato quale **MARGOT**, occorre necessariamente prendere in considerazione la possibilità di effettuare opportuni test di comparazione con data-set verificati da questo punto di vista, quali **CE-ACL-14** e **CE-EMNLP-15**.

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 3), max_features: None	0.8457	0.8415
Ngrams_second	n-gram range: (1, 3), max_features: None		
LinearSVC	C: 10, loss: hinge	0.8634	0.8591

Tabella 5.2: Risultati della calibrazione dei parametri associati alla *baseline* basata su *n-gram*.

	Parametri	Accuracy	F1-score
Scelta classificatore		0.5494	0.5547
LogisticRegression	C: 0.001, penalty: L1	0.50	0.6666

Tabella 5.3: Risultati della calibrazione dei parametri associati alla *baseline* basata su *cosine similarity*.

	Parametri	Accuracy	F1-score
couple_first_word_binary	stop_words: english	0.7961	0.7976
first_word_binary_first	stop_words: english		
first_word_binary_second	stop_words: english		
word_pairs_binary	stop_words: None		
LinearSVC	C: 0.01, loss: hinge	0.8222	0.8126

Tabella 5.4: Risultati della calibrazione dei parametri associati al classificatore basato sulle *feature* introdotte da Stab & Gurevych.

Batch_size	accuracy	f1-score
64	**0.7883**	**0.7938**
256	0.7788	0.7753
500	0.7662	0.7572
1000	0.7541	0.7709

Tabella 5.5: Risultati della calibrazione dei parametri associati al classificatore basato su reti neurali ricorrenti.

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 3), max_features: None	0.8607	0.8522
Ngrams_second	n-gram range: (1, 2), max_features: 20000		
SGDClassifier	alpha: 0.0001, loss: hinge, penalty: l2	0.8558	0.8497

Tabella 5.6: Risultati della calibrazione dei parametri associati al classificatore basato sulle *feature* proprie della *stance classification*.

Classificatore	accuracy	f1-score
Baseline n-grams	**0.8719**	**0.8674**
Baseline cosine similarity	0.50	0.6666
Classificatore Stab & Gurevych	0.8270	0.8174
Rete neurale	0.7998	0.8033
Rete neurale (dropout = 0.2)	0.7887	0.7893
Classificatore SC	0.8603	0.8575

Tabella 5.7: Performance classificatori sui data-set di interesse. Per ognuno di essi sono riportate le metriche *accuracy* e *f1-score*.

5.5 Comparazione dei risultati di classificazione ottenuti con il corpus CE-EMNLP-15 mediante la tecnica *Leave One Topic Out*

L'introduzione di un nuovo corpus per la *argument structure prediction*, costruito sulla base di criteri soggettivi, richiede test atti a valutarne la validità. Per tale motivo, è stato preso in considerazione il data-set **CE-EMNLP-15**, per il quale sussiste una suddivisione dei dati in base al *topic* in due set differenti, ovvero *train and test* e *held-out*. Più precisamente, sulla base della presente informazione, si è deciso di effettuare un test *Leave One Topic Out*, in base al quale, dopo aver effettuato una

fase preliminare di calibrazione dei parametri di configurazione considerando come corpus di riferimento il data-set relativo ai *topic* di *held-out*, si effettua il training dei molteplici classificatori sul set di *train and test* escludendo di volta in volta i dati associati ad un dato *topic* preso in considerazione. Pertanto, una volta effettuata la calibrazione sul data-set composto dai soli dati aventi *topic* appartenente ai 19 di *held-out*, si procede con il test vero e proprio: a turno vengono considerati come test set tutti i dati aventi un determinato *topic* in questione. Successivamente, sui rimanenti viene effettuato il *training* dei vari classificatori per poi misurare in seguito le loro performance sul test set. Infine, come ultimo passo si procede con il calcolo della media delle metriche di riferimento usate in ciascun test, ovvero *accuracy* e *f1-score*. A questo punto, una volta introdotto il concetto, possiamo procedere alla descrizione dettagliata di tutte le varie sotto-fasi che lo realizzano. Innanzitutto, è necessario ricavare i data-set di *held-out* e *train and test* a partire dal corpus **CE-EMNLP-15** di riferimento per la *argument structure prediction* (tabella 5.8).

Data-set	Dimensione	# link	# no-link	# topic
Held-out	41283	1721	39562	19
Train and test	77516	3447	74069	39

Tabella 5.8: Statistiche data-set di *held-out* e *train and test* relativi a **CE-EMNLP-15**. In particolare, Per ognuno di essi sono riportate le seguenti informazioni: numero di elementi contenuti nel data-set, distribuzione delle classi relative alla *argument structure prediction* e infine il numero di *topic* contenuti.

Successivamente, si passa alla fase di calibrazione dei parametri associati alle *feature* utilizzate da ciascun classificatore, alla selezione del miglior algoritmo di classificazione e dei suoi rispettivi parametri (tabelle 5.9, 5.10, 5.11, 5.12 e 5.13). In particolare, come è già stato precedentemente anticipato, in tale scenario il data-set di riferimento è quello costituito dai *topic* appartenenti al set di *held-out*. Analogamente a quanto effettuato nelle precedenti sezioni, si considerano tutti i classificatori introdotti.

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 1), max_features: None	0.6578	0.1177
Ngrams_second	n-gram range: (1, 1), max_features: None		
SGDClassifier	alpha: 0.001, loss: hinge, penalty: L2	0.6296	0.1199

Tabella 5.9: Risultati della calibrazione dei parametri associati alla *baseline* basata su *n-gram*.

	Parametri	Accuracy	F1-score
Scelta classificatore		0.7205	0.1395
SGDClassifier	alpha: 1e-05, loss: hinge, penalty: elasticnet	0.8568	0.1586

Tabella 5.10: Risultati della calibrazione dei parametri associati alla *baseline* basata su *cosine similarity*.

	Parametri	Accuracy	F1-score
couple_first_word_binary	stop_words: None	0.8855	0.2423
first_word_binary_first	stop_words: None		
first_word_binary_second	stop_words: None		
word_pairs_binary	stop_words: None		
SGDClassifier	alpha: 1e-05, loss: hinge, penalty: L1	0.9197	0.2584

Tabella 5.11: Risultati della calibrazione dei parametri associati al classificatore basato sulle *feature* introdotte da Stab & Gurevych.

Batch_size	accuracy	f1-score
64	0.6651	0.1185
64 (dropout)	**0.7188**	**0.1271**
256	0.6366	0.1162
500	0.5945	0.1138
1000	0.6135	0.1170

Tabella 5.12: Risultati della calibrazione dei parametri associati al classificatore basato su reti neurali ricorrenti.

	Parametri	Accuracy	F1-score
Ngrams_first	n-gram range: (1, 1), max_features: None	0.7763	0.1553
Ngrams_second	n-gram range: (1, 3), max_features: 10000		
LogisticRegression	C: 1.0, penalty: L1	0.7783	0.1762

Tabella 5.13: Risultati della calibrazione dei parametri associati al classificatore basato sulle *feature* proprie della *stance classification*.

Terminata quest'ultima fase, si considera il data-set di *train and test* e si effettua il test *Leave One Topic Out*. Nello specifico, vengono riportate le informazioni relative alle metriche principali di riferimento, i.e. *accuracy* e *f1-score*, nell'ottica di ciascun *topic* (grafici 5.5 e 5.6) e la media complessiva (tabella 5.14). In particolare, come si può osservare da quest'ultima, analogamente a quanto riportato nel capitolo precedente per i due corpora **CE-ACL-14** e **CE-EMNLP-15**, i risultati di classificazione riportati dai differenti classificatori non sono soddisfacenti, raggiungendo, nello specifico, il valore massimo per la metrica di riferimento principale, i.e. *f1-score*, di 0.2749. A tal fine, occorre però sottolineare come il classificatore basato sul sottoinsieme di *feature* proposte da Stab e Gurevych[148] riesca anche in questo contesto ad adattarsi meglio allo scenario di studio in questione. Infine, ancora una volta il classificatore basato sulle *feature* proprie della *stance classification* si è dimostrato sufficientemente adatto anche per l'attività dell'*argument structure prediction* rimanendo sempre nell'ottica di una semplice comparazione dei risultati ottenuti tra i differenti classificatori.

Classificatore	accuracy	f1-score
Baseline n-grams	0.6828	0.1199
Baseline cosine similarity	0.5886	0.2323
Classificatore Stab & Gurevych	**0.7414**	**0.2749**
Rete neurale (dropout)	0.5511	0.1345
Classificatore SC	0.6922	0.2660

Tabella 5.14: Risultati ottenuti dai classificatori al termine del test *Leave One Topic Out*, in accordo con le metriche di riferimento, ovvero *accuracy* e *f1-score*.

Figura 5.5: Performance relative alla metrica *accuracy* per ciascun *topic* appartenente al set di *train and test* nell'ambito del test *Leave One Topic Out*. Al fine di garantire una migliore distinzione dei valori riportati da ogni classificatori, i dati sono stati ordinati rispetto al migliore classificatore, i.e. *Classificatore Stab & Gurevych*, per la metrica di riferimento scelta. I risultati dettagliati per ciascun classificatore sono riportati all'interno dell'appendice C (tabella C.2).

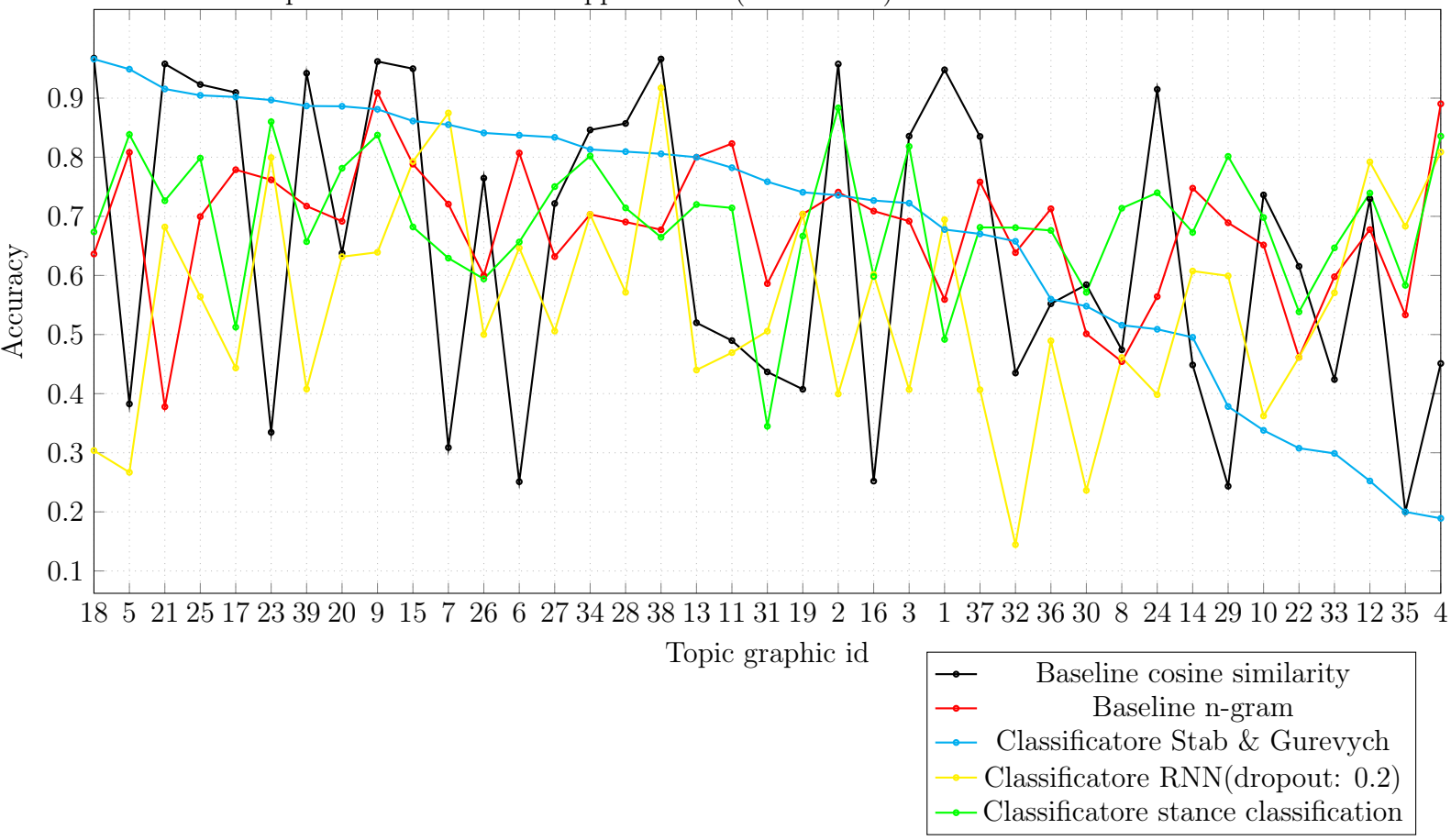
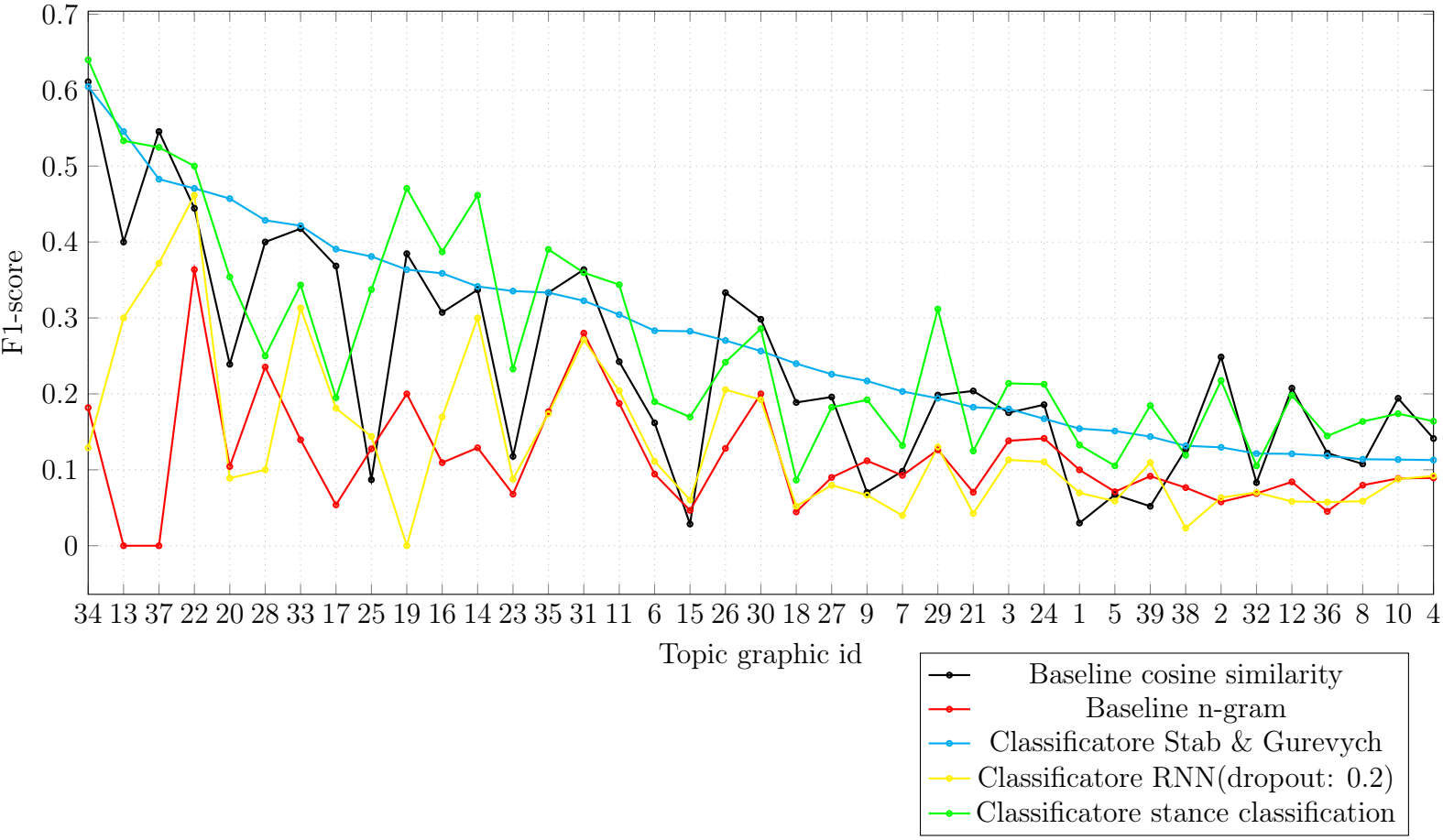


Figura 5.6: Performance relative alla metrica *f1-score* per ciascun *topic* appartenente al set di *train and test.* nell'ambito del test *Leave One Topic Out.* Al fine di garantire una migliore distinzione dei valori riportati da ogni classificatori, i dati sono stati ordinati rispetto al migliore classificatore, i.e. *Classificatore Stab & Gurevych*, per la metrica di riferimento scelta. I risultati dettagliati per ciascun classificatore sono riportati all'interno dell'appendice C (tabella C.2).



Infine, ricollegandosi al discorso iniziale, vengono valutate le performance dei vari classificatori, dopo aver effettuato il training sul corpus presentato all'interno del capitolo, ottenuto tramite l'utilizzo del software **MARGOT**, sul data-set di *train and test* relativo a **CE-EMNLP-15**. Più precisamente, si ricavano informazioni di tipo statistico quali le ma-

trici di confusione e i *report* di classificazione per ciascun *topic* appartenente al set di *train and test* (tabelle 5.20, 5.26, 5.5 e 5.6) e si calcolano i valori medi relativi alle metriche *accuracy* e *f1-score*. Infine, vengono riportate anche le statistiche generali riguardo alla distribuzione delle classi in merito alle predizioni effettuate dai diversi classificatori (tabella 5.27). A tal proposito, come si può osservare con particolare attenzione dai risultati generali relativi alle metriche *accuracy* e *f1-score* riportati da quest'ultima tabella, si può constatare come i risultati ottenuti dai vari classificatori testimonino la loro estrema difficoltà nel riuscire a discriminare correttamente le due classi di esempi. In particolare, il valore massimo per la metrica principale di riferimento, i.e. *f1-score*, è pari a 0.1858, raggiunto dal classificatore basato sulle *feature* proposte per l'attività di *stance classification*, seguito da quello relativo alle *feature* introdotte da Stab e Gurevych[148]. A motivazione dei risultati insoddisfacenti ottenuti, occorre ribadire il concetto relativo alla problematica dello forte sbilanciamento dei dati, il quale, nel caso specifico dell'analisi per singolo *topic*, si presenta in maniera ancora più marcata. Infine, occorre sottolineare un ulteriore aspetto di notevole importanza all'interno del presente test di comparazione, maggiormente evidenziato dai risultati specifici per *topic* (figure 5.7 e 5.8), relativo alla difficoltà di adattamento per la particolare attività di *argument structure prediction* su ambiti di diverso genere. In altre parole, un elemento aggiuntivo di difficoltà è rappresentato per l'appunto dalla differente natura dei dati riportati all'interno dei data-set **CE-EMNLP-15** e quello introdotto nel presente capitolo mediante l'utilizzo dello strumento di *argument detection* **MARGOT**.

Figura 5.7: Performance relative alla metrica *accuracy* per ciascun *topic* appartenente al set di *train and test* da parte dei classificatori allenati sul data-set ottenuto mediante lo strumento **MARGOT** nell'ambito del test *Leave One Topic Out*. Al fine di garantire una migliore distinzione dei valori riportati da ogni classificatori, i dati sono stati ordinati rispetto al migliore classificatore, i.e. *Classificatore RNN*, per la metrica di riferimento scelta. I risultati dettagliati per ciascun classificatore sono riportati all'interno dell'appendice C (tabella C.3).

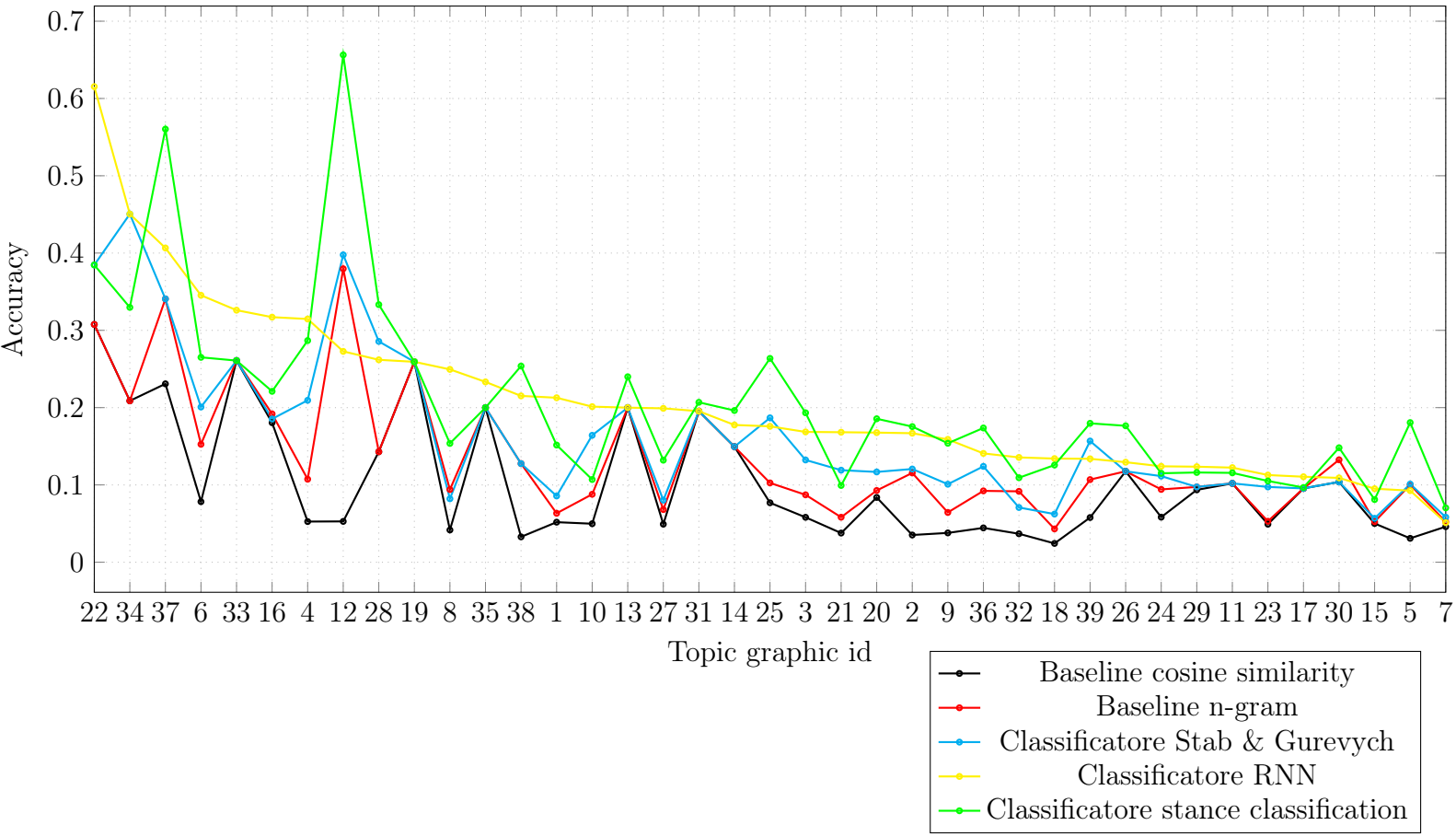


Figura 5.8: Performance relative alla metrica *f1-score* per ciascun *topic* appartenente al set di *train and test* da parte dei classificatori allenati sul data-set ottenuto mediante lo strumento **MARGOT** nell'ambito del test *Leave One Topic Out*. Al fine di garantire una migliore distinzione dei valori riportati da ogni classificatori, i dati sono stati ordinati rispetto al migliore classificatore, i.e. *Classificatore stance classification*, per la metrica di riferimento scelta. I risultati dettagliati per ciascun classificatore sono riportati all'interno dell'appendice C (tabella C.3).

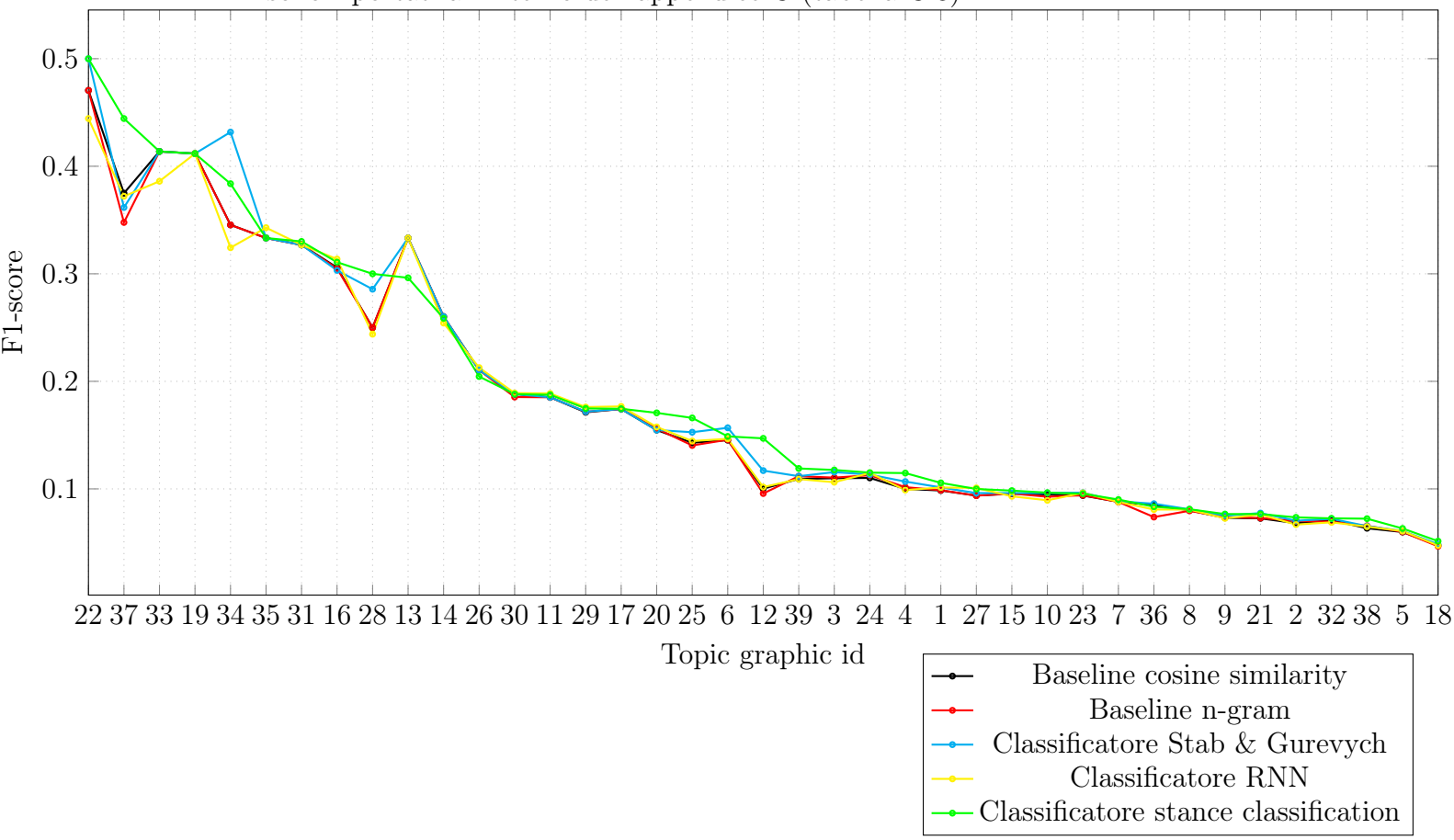


Tabella 5.15: N-grams

	Precision	Recall	F1-score	Support
Link	0.1020	0.9589	0.1762	3447
No-link	0.6529	0.0397	0.0689	74069
Avg/Total	0.6284	0.0806	0.0737	77516

Tabella 5.16: Cosine similarity

	Precision	Recall	F1-score	Support
Link	0.1021	1.00	0.1772	3447
No-link	0.00	0.00	0.00	74069
Avg/Total	0.0045	0.0444	0.0078	77516

Tabella 5.17: Stab & Gurevych

	Precision	Recall	F1-score	Support
Link	0.1064	0.9658	0.1828	3447
No-link	0.7154	0.0713	0.1223	74069
Avg/Total	0.6883	0.1111	0.1250	77516

Tabella 5.18: Rete neurale (dropout)

	Precision	Recall	F1-score	Support
Link	0.1060	0.8813	0.1760	3447
No-link	0.8600	0.1352	0.2129	74069
Avg/Total	0.8265	0.1684	0.2113	77516

Tabella 5.19: Classificatore SC

	Precision	Recall	F1-score	Support
Link	0.1091	0.9374	0.1858	3447
No-link	0.8862	0.1215	0.1962	74069
Avg/Total	0.8516	0.1577	0.1957	77516

Tabella 5.20: *Report* di classificazione dei classificatori presi in esame. In ognuno, sono riportate le informazioni relative ad ogni singola classe in accordo con le metriche *precision*, *recall* e *f1-score*.

Tabella 5.21: N-grams

Predicted	link	no-link
True		
link	3255	192
no-link	70225	3844

Tabella 5.22: Cosine similarity

Predicted	link	no-link
True		
link	3447	0
no-link	74069	0

Tabella 5.23: Stab & Gurevych

Predicted	link	no-link
True		
link	3326	121
no-link	69290	4779

Tabella 5.24: Rete neurale (dropout)

Predicted	link	no-link
True		
link	3027	420
no-link	64426	9643

Tabella 5.25: Classificatore SC

Predicted	link	no-link
True		
link	3221	226
no-link	64679	9390

Tabella 5.26: Matrici di confusione relative a ciascun classificatore preso in esame.

Classificatore	Accuracy	F1-score	# link predetti	# no-link predetti
Baseline n-grams	0.1356	0.1762	73480	4036
Baseline cosine similarity	0.1021	0.1772	77516	0
Classificatore Stab & Gurevych	0.1638	0.1828	72616	4900
Rete neurale	**0.2070**	0.1760	67453	10063
Classificatore SC	0.2067	**0.1858**	67900	9616

Tabella 5.27: Risultati ottenuti dai classificatori in accordo con le metriche di riferimento quali *accuracy* e *f1-score*. Quest'ultimo rappresenta una misura più precisa della voce relativa alla classe positiva *link* riportata nel *report* di classificazione. Infine, per ogni classificatore è riportata la distribuzione delle classi predette, confrontabile con la colonna *support* esposta nel *report* di classificazione.

5.6 Un'ulteriore verifica mediante l'inversione dello stesso procedimento

Come ulteriore esperimento, si vuole effettuare il ragionamento inverso di quanto esposto nella sezione precedente. In altri termini, si desidera analizzare il comportamento dei vari classificatori sul nuovo corpus, dopo aver effettuato il *training* sul data-set di **CE-EMNLP-15** relativo ai *topic* di *train and test*. In particolare, come configurazione di ciascun classificatore, si considera quella ricavata dalle ricerche esaustive effettuate sul set di *held-out*. Analogamente a quanto presentato nella sezione precedente, si ricavano le matrici di confusione e i *report* di classificazione per ciascun *topic*, la media delle metriche di riferimento *accuracy* e *f1-score* e infine le statistiche relative alla distribuzione delle classi (tabelle 5.33, 5.39 e 5.40). A differenza di quanto osservato precedentemente, i risultati riportati in particolar modo dalla tabella 5.40 relative ai valori medi delle metriche di riferimento, riportano misure decisamente migliori ma ancora purtroppo non sufficienti per considerarli soddisfacenti. Nello specifico, il valore massimo relativo alla metrica *f1-score* è pari a 0.6117, riportato dalla *baseline* basata sulla *feature cosine similarity*, mentre il classificatore relativo alle *feature* di Stab e Gurevych, che fino ad ora ha quasi sempre prevalso, ottiene un valore basso pari a 0.3553. In particolare, sulla base dei risultati esposti da entrambi i test sui data-set **CE-EMNLP-15** e quello costruito per mezzo di **MARGOT**, si può constatare come le difficoltà riscontrate nell'ambito dell'assenza di un vero e proprio data-set adatto per l'attività di *argument structure prediction* rappresentino l'ostacolo principale agli obiettivi delineati dal presente elaborato. Pertanto, una volta superata tale problematica, si può procedere con la valutazione di altri aspetti di notevole importanza quali l'analisi di potenziali legami tra la *stance classification* e l'*argument structure prediction*, la verifica dell'applicabilità di *feature* specifiche per il primo settore di ricerca citato nell'ambito di quest'ultimo e infine la validazione delle metodologie di generazione di nuovi corpora per l'*argument structure prediction*, in grado potenzialmente di rappresentare un importante strumento d'ausilio per risolvere le problematiche relative alla definizione di opportuni data-set adatti per tale scopo.

Tabella 5.28: N-grams

	Precision	Recall	F1-score	Support
Link	0.50	0.24	0.33	30000
No-link	0.50	0.76	0.60	30000
Avg/Total	0.50	0.50	0.47	60000

Tabella 5.29: Cosine similarity

	Precision	Recall	F1-score	Support
Link	0.53	0.73	0.61	30000
No-link	0.56	0.35	0.43	30000
Avg/Total	0.54	0.54	0.52	60000

Tabella 5.30: Stab & Gurevych

	Precision	Recall	F1-score	Support
Link	0.63	0.25	0.35	30000
No-link	0.53	0.86	0.66	30000
Avg/Total	0.58	0.55	0.51	60000

Tabella 5.31: Rete neurale (dropout)

	Precision	Recall	F1-score	Support
Link	0.53	0.45	0.49	30000
No-link	0.52	0.60	0.56	30000
Avg/Total	0.53	0.53	0.52	60000

Tabella 5.32: Classificatore SC

	Precision	Recall	F1-score	Support
Link	0.56	0.36	0.44	30000
No-link	0.53	0.71	0.61	30000
Avg/Total	0.54	0.54	0.52	60000

Tabella 5.33: *Report* di classificazione dei classificatori presi in esame. In ognuno, sono riportate le informazioni relative ad ogni singola classe in accordo con le metriche *precision*, *recall* e *f1-score*.

Tabella 5.34: N-grams

Predicted	link	no-link
True		
link	7263	22737
no-link	7188	22812

Tabella 5.35: Cosine similarity

Predicted	link	no-link
True		
link	21813	8187
no-link	19505	10495

Tabella 5.36: Stab & Gurevych

Predicted	link	no-link
True		
link	7415	22585
no-link	4316	25684

Tabella 5.37: Rete neurale (dropout)

Predicted	link	no-link
True		
link	13536	16464
no-link	11991	18009

Tabella 5.38: Classificatore SC

Predicted	link	no-link
True		
link	10872	19128
no-link	8552	21448

Tabella 5.39: Matrici di confusione relative a ciascun classificatore preso in esame.

Classificatore	Accuracy	F1-score	# link predetti	# no-link predetti
Baseline n-grams	0.5012	0.3267	14451	45549
Baseline cosine similarity	0.5384	**0.6117**	41318	18682
Classificatore Stab & Gurevych	**0.5516**	0.3553	11731	48269
Rete neurale (dropout)	0.5257	0.4875	25527	34473
Classificatore SC	0.5386	0.4399	19424	40576

Tabella 5.40: Risultati ottenuti dai classificatori in accordo con le metriche di riferimento quali *accuracy* e *f1-score*. Quest'ultimo rappresenta una misura più precisa della voce relativa alla classe positiva *link* riportata nel *report* di classificazione. Infine, per ogni classificatore è riportata la distribuzione delle classi predette, confrontabile con la colonna *support* esposta nel *report* di classificazione.

5.7 Un ultimo test di validazione del corpus ottenuto mediante MARGOT basandosi sulle informazioni relative al *rebuttal*

Infine, un ultimo test di validazione riguardo al processo di creazione del data-set ottenuto per mezzo dello strumento **MARGOT** prende come riferimento i meta-dati forniti all'interno del corpus *Create Debate custom data-set*. Più precisamente, risulta interessante usufruire delle informazioni relative al *rebuttal* tra post consecutivi appartenenti ad una data discussione, in quanto esse rappresentano un termine di confronto di validità certificata. Nello specifico, l'esperimento consiste nella predizione dei legami tra coppie di *post* basandosi sulle informazioni ottenute nell'ambito della *argument relation*, relativa alle coppie *evidence* - *claim* presenti al loro interno, e avendo come valore di riferimento il *rebuttal* associato a ciascuna coppia di post. Dal punto di vista del procedimento, l'esperimento relativo alla predizione dei legami di supporto - non supporto tra coppie di *post* adiacenti può essere riassunto come segue:

- Estrazione delle coppie *evidence* - *claim*: poiché i corpora a disposizione fanno riferimento ai componenti di un argomento, i.e. *evidence* e *claim*, occorre innanzitutto definire un processo di classificazione atto a predire la *argument relation* di ciascuna coppia *evidence* - *claim* estratta all'interno del corpus di **MARGOT**. Pertanto, come prima fase, risulta necessario costruire coppie incrociate *evidence* - *claim* tra post adiacenti appartenenti ad una stessa discussione. In questo modo, si ottengono delle coppie volte a modellare la relazione espressa dal *rebuttal*.

- Classificazione delle coppie *evidence* - *claim*: una volta definito il corpus di riferimento per l'esperimento, si considerano tutti i classificatori introdotti e si effettua una fase di *training* sul data-set **CE-EMNLP-15**. Successivamente, si procede con la predizione della *argument relation* di ogni singola coppia *evidence* - *claim* estratta.

- Predizione dei legami a livello di post: al fine di poter definire dei dati atti al confronto con le informazioni relative al *rebuttal*, occorre passare dal livello di singola frase a quello di *post*. Nel presente

esperimento, si procede secondo il seguente criterio: dati due post consecutivi `p1` e `p2`, esiste un legame di tipo supporto, i.e. `link`, se è presente almeno una coppia costituita da una *evidence* appartenente a `p1` e un *claim* presente in `p2` per la quale il classificatore ha predetto una relazione di supporto.

Al termine della prima fase sono state estratte 146,786 coppie *evidence - claim* relative a post adiacenti all'interno di una stessa discussione. Più precisamente, sono stati applicati gli stessi filtri di selezione dei dati descritti per il data-set di **MARGOT**. Tuttavia, per motivi prettamente computazionali legati all'estrazione delle *feature* più onerose in termini di risorse, sono state selezionate solamente le prime 50,000 coppie di lunghezza maggiore, appartenenti a 2,112 coppie di *post*. A tale proposito, come si può osservare dalla tabella 5.41, è importante sottolineare come la validità del test sia comunque mantenuta dato che le proporzioni del numero di esempi positivi e negativi sono pressoché invariate. Infine, una volta terminata la fase di classificazione, i risultati ottenuti per ciascuna coppia di post vengono confrontati con le informazioni relative al *rebuttal* secondo le metriche *precision* e *recall*. In particolare, come si può osservare dalla tabella 5.42, i valori insoddisfacenti ottenuti testimoniano l'incapacità dei classificatori proposti nel distinguere correttamente gli esempi positivi dai negativi. Tuttavia, il presente risultato trova spiegazione se si considera la distribuzione dei valori associati all'informazione *rebuttal*. Nello specifico, solamente 179 delle 2,112 coppie di *post* presentano un legame positivo, i.e. `rebuttal=support`. Pertanto, un forte sbilanciamento dei dati verso una determinata classe in congiunzione con le problematiche riscontrate nell'ambito del data-set **CE-EMNLP-15** comportano notevoli difficoltà da parte dei classificatori nel poter discriminare correttamente le due tipologie di legami di interesse tra post.

	Coppie evidence - claim	Coppie di post	Coppie di post aventi rebuttal=support	Coppie di post aventi rebuttal=none|oppose	Percentuale esempi positivi
Selezionati	50000	2112	179	1933	8.4753%
Totali	146786	2860	285	2575	9.9650%

Tabella 5.41: Statistiche corpus selezionato per l'esperimento in questione e di quello originale previsa fase di selezione. In particolare, la percentuale di esempi positivi fa riferimento al numero di coppie di post aventi `rebuttal=support` all'interno del data-set.

Classificatore	Precision	Recall
Baseline n-gram	0.0762	0.0949
Baseline cosine similarity	0.0794	**0.5642**
Classificatore Stab & Gurevych	0.0737	0.1899
Rete neurale	0.0763	0.4078
Classificatore SC	**0.0851**	**0.5642**

Tabella 5.42: Risultati al termine dell'esperimento ottenuti da ciascun classificatore.

Conclusioni

Quanto proposto all'interno del presente elaborato rappresenta solamente una delle molteplici varianti applicabili atte a rispondere correttamente agli obiettivi imposti. Tuttavia, sulla base delle informazioni raccolte ed esposte all'interno dei capitoli sperimentali risulta comunque possibile presentare alcune considerazioni di validità generale.

Avanzamento dello state dell'arte

Innanzitutto, occorre sottolineare come i risultati soddisfacenti ottenuti nell'ambito della *stance classification* avanzano lo stato dell'arte su uno dei molteplici corpora esistenti. Nello specifico, si evidenzia l'importanza delle *feature* introdotte ed implementate, le quali si sono dimostrate in grado di catturare gli aspetti sintattici e semantici di interesse in modo equivalente ai metodi più complessi utilizzati da Ferreira e Vlachos[46]. A tale proposito, l'aggiunta delle *feature* scartate per molteplici motivi e la ricerca di una soluzione ottima nella fase di calibrazione del classificatore possono certamente rappresentare un punto di partenza per lo sviluppo di nuovi miglioramenti nell'ambito della *stance classification*.

Il problema dei dati

Inoltre, un aspetto di fondamentale importanza riguarda l'importanza delle problematiche introdotte ed evidenziate parallelamente all'esposizione degli approcci adottati e dei relativi risultati ottenuti. Più precisamente, l'ostacolo maggiore è stato rappresentato dalla significativa difficoltà nell'individuare o definire dei data-set adatti per le attività di *stance classification* e, in particolare modo, di *argument structure prediction*. In primo luogo, solamente il progetto Emergent[46] e il corpus definito da Bar-Haim et al.[14], tuttavia non disponibile durante lo sviluppo del presente elaborato,

contengono dati la cui tipologia presenta elementi di notevole affinità con quelli di interesse per l'*argument structure prediction*. Come conseguenza, risulta certamente necessario tenere conto del presente divario durante la fase di classificazione dei corpora introdotti da ***IBM*** *Research*, presi come riferimento per l'attività di *argument structure prediction*. In aggiunta, anche per quest'ultimo contesto applicativo, la mancanza di dati organizzati secondo le due categorie di classificazione scelte, i.e. supporto e opposizione, ha rappresentato un importante elemento di scelta dal punto di vista della definizione di un'opportuna soluzione. Pertanto, come conseguenza diretta di quanto appena osservato, i risultati di classificazione ottenuti nell'ambito della *argument structure prediction* sono decisamente insoddisfacenti. Ad esempio, la notevole varianza in termini quantitativi di esempi per ciascun *topic* nel contesto del test *Leave One Topic Out* ha comportato come esito prestazioni notevolmente basse e altalenanti. A tal fine, subentra necessariamente l'analisi atta ad individuare nuovi strumenti o approcci in grado di semplificare la problematica relativa alla mancanza di opportuni corpora per le molteplici attività proprie dell'*argumentation mining*. Nello specifico, come è già stato osservato all'interno del capitolo relativo a quest'ultimo settore di ricerca (sezione 2.6), l'ausilio di strumenti di collaborazione collettiva, quale *Amazon Mechanical Turk*, o l'introduzione di accurati modelli computazionali, quali le reti neurali, possono certamente velocizzare il processo di costruzione di nuovi corpora; tuttavia occorre sempre ribadire la necessità di dover sormontare l'ostacolo principale relativo alla selezione di un modello argomentativo universalmente riconosciuto, questione, purtroppo, la cui attuabilità non è per forza verificata.

Il criterio di classificazione

Inoltre, occorre precisare un ulteriore fattore arbitrario potenzialmente incidente in maniera significativa sulle performance riportate. In particolar modo, anche la scelta della tipologia di classificazione, ovvero le classi relative alla *argument relation*, può rappresentare una causa aggiuntiva alle problematiche precedentemente esposte. Più precisamente, risulta improbabile pensare che tutti gli esempi riportati e costruiti secondo degli opportuni principi di

selezione nell'ambito dei corpora di ***IBM*** *Research* siano collocabili secondo una suddivisione netta espressa dalle classi 'opposizione' e 'supporto'. Pertanto, la sperimentazione di un set di valori di maggior granularità e specificità potrebbero catturare in modo migliore la diversità dei dati selezionati per l'attività di *argument structure prediction*. In aggiunta, quanto appena osservato è altresì applicabile per il data-set definito mediante l'ausilio dello strumento di *argument detection* **MARGOT**; tuttavia, il contesto particolare dei siti di dibattito bilaterali presenta certamente maggiori affinità con un tipo di classificazione binaria, osservazione che viene inoltre rispecchiata dagli incoraggianti risultati ottenuti.

L'importanza degli esperimenti effettuati

Nonostante le evidenti problematiche riscontrate, in un'ottica relativa alla singola comparazione delle prestazioni di ciascun classificatore, quello definito secondo le *feature* principali adottate per l'attività di *stance classification* riporta risultati superiori alle semplici *baseline* introdotte e si avvicina nella prevalenza dei casi al migliore classificatore, basato sulle *feature* proposte da Stab e Gurevych[148] nell'ambito dei saggi di raccomandazione. Pertanto, una volta risolte le principali problematiche di fondo, l'applicazione degli approcci presentati all'interno del presente elaborato potrebbero evidenziare un netto legame tra il valore informativo rappresentato dalla *stance classification* e gli obiettivi imposti dall'*argument structure prediction*, delineando di conseguenza una solida base per lo sviluppo delle fasi più avanzate di quest'ultimo settore di ricerca. Inoltre, nel caso di risultati di comparazione positivi, l'applicazione del test *Leave One Topic Out* potrebbe certamente convalidare la *ground-truth* dietro ai criteri di selezione che hanno portato alla definizione di un nuovo corpus, aprendo la strada verso nuove tipologie di approcci volti a semplificare ulteriormente il processo di costruzione di nuove collezioni di dati per l'*argument structure prediction*.

Una valida alternativa?

Infine, come ulteriore proposito per il futuro, la definizione di nuove metodologie e approcci regolati da principi totalmente diversi da quelli descritti all'interno del presente elaborato, atti a replicare il processo di apprendimento degli individui, quale, ad esempio, il

reinforcement learning, potrebbero portare a risultati inaspettati e contribuire a ridurre maggiormente il cosiddetto *abstraction gap* che sussiste tra la dimensione sintattica e lessicale delle espressioni e quella puramente semantica e legata ai processi razionali e dialogici dell'essere umano.

Appendice A

Implementazione e ottimizzazione delle *feature* per la *stance classification*

La presente appendice si pone come obiettivo la descrizione dettagliata dal punto di vista implementativo dei seguenti aspetti: (1) le *feature* impiegate per l'attività di *stance classification*, (2) il processo di ottimizzazione del loro utilizzo durante le fasi di classificazione e infine (3) l'architettura secondo il modello *pipeline* alla base dei molteplici classificatori impiegati.

A.1 *Feature* di classificazione

Procedendo con ordine, occorre precisare le caratteristiche tecniche relative all'estrazione delle *feature* utilizzate. Inizialmente, per quanto concerne gli *n-gram* e la relativa estensione degli *skip n-gram* si ricorre all'utilizzo degli strumenti di supporto forniti dalla libreria `sklearn`. Più precisamente, nel primo caso occorre semplicemente far uso dell'estrattore `TfidfVectorizer`, il quale consente di estrarre tutti i diversi gradi di *n-gram* a partire da una sequenza di documenti testuali, con particolare accezione alla variante **TF-IDF**. D'altro canto, l'implementazione della *feature skip n-gram* non è nativamente supportata ed è stato quindi necessario definire un opportuno estrattore (codice A.1), a cui segue il trasformatore **TF-IDF**, i.e. `TfidfTransformer`, propo-

sto dalla libreria `sklearn`. Infine, rimanendo sempre nel contesto degli *n-gram*, è stato seguito un ragionamento analogo per quanto riguarda l'implementazione dei *sn-gram*, introducendo, nello specifico, la classe `SyntacticCountVectorizer` (codice A.2), basata prevalentemente sull'ausilio dello strumento *Stanford Parser*. Successivamente, si introducono *feature* relativamente semplici atte a catturare la dimensione sintattiche dei testi analizzati: *basic* e *repeated punctuation*, riportate in dettaglio dalle figure A.3 e A.4. In aggiunta, per quanto concerne le *feature* relative alle dipendenze sintattiche, si definisce un'opportuna variante della classe `CountVectorizer` (codice A.5), ovvero la versione base del `TfidfVectorizer` sprovvista del concetto di **TF-IDF**. Infine, si introducono *feature* più particolari quali la *cosine similarity* e le triple **SVO**. Nel primo caso, come si può osservare dal codice riportato nella figura A.6, i documenti testuali in oggetto vengono dapprima codificati nei rispettivi *word embedding* mediante l'ausilio dello strumento *Word2Vec* introdotto da Google. Nello specifico, come implementato da Ferreira e Vlachos[46], tutti i vettori, ciascuno dei quali associato ad una specifica parola, sono sommati tra loro per ottenere infine un'unica rappresentazione vettoriale sulla quale effettuare l'operazione di similarità. Pertanto, come riportato dalla figura A.7, si effettua il calcolo della similarità coseno tra i due vettori risultanti ottenuti precedentemente. Per quanto riguarda, invece, le triple **SVO**, occorre usufruire dello *Stanford Parser* per poter dapprima suddividere i documenti testuali nelle rispettive proposizioni e infine poter costruire i relativi *costituency parse tree* (codice A.8). Una volta estratte tutte le triple **SVO** si procede con la fase di comparazione tra quelle appartenenti ad un dato *claim* e le altre associate alla rispettiva *evidence* (codice A.9). In particolare, si fa riferimento al *Paraphrase Database*[117] (**PPDB**) come quanto descritto da Ferreira e Vlachos[46].

Listing A.1: Classe *SkipGramVectorizer* in grado di estrarre *skip n-gram* di diverso grado.

```
class SkipGramVectorizer(CountVectorizer, PickleCompliantVectorizer):
    """

    Simple CountVectorizer extension for skip n-grams extraction.

    """

    [...]
```

```
9
    def _extract_skipgrams(self, tokens, stop_words=None):
11      """
        Extracts skip n-grams from given tokenized sentence. Input
    data can be further pre-processed via given stop
13      words.

15      :param tokens: tokenized input via Stanford Parser
        :param stop_words: set of stop words used in order to filter
    input data
17      :return: list of extracted skip n-grams
        """
19
        result = []
21
        if stop_words is not None:
23          tokens = [word for word in tokens if word not in
    stop_words]

25      for n in range(self.ngram_range[0], self.ngram_range[1] + 1):
            for k in range(self.skip_range[0], self.skip_range[1] + 1)
    :
27              result.append(list(skipgrams(tokens, n, k)))

29      return [item for sublist in result for item in sublist]
```

Listing A.2: Classe *SyntacticCountVectorizer* la quale consente di estrarre *sn-gram* di grado arbitrario.

```
1 class SyntacticCountVectorizer(CountVectorizer,
    PickleCompliantVectorizer):

3   """

5   Simple CountVectorizer extension for syntactic n-grams feature
    extraction.

7   """

9   [...]

11  def _extract_sngrams(self, text):
        """
13      Extracts syntactic n-grams from given input text. Sngrams are
    retrieved via Stanford Dependency Parser.
```

```
        :param text: input text
        :return: list of sngrams
        """

        parsed_doc = self.utils.dependency_parser.raw_parse(text)
        dependencies = parsed_doc.next()

        sn_bigrams = []

        # bigrams
        for triple in list(dependencies.triples()):
            sn_bigrams.append((triple[0][0], triple[2][0]))

        if self.sngram_range > 2:
            return sn_bigrams + self._build_higher_level_sngrams(
    sn_bigrams)
        else:
            return sn_bigrams

    [...]
```

Listing A.3: Funzione principale nell'ambito dell'estrazione di tutte le *feature basic*. Nello specifico, l'estrattore associato si limita all'invocazione della presente funzione per ciascun documento testuale preso in analisi.

```
    def extract_basic_features_from_text(self, data):
        """
        Extracts the following basic features:

            - Number of characters
            - Number of words
            - Number of sentences
            - Average word length
            - Average sentence length
            * Number of question marks ('?')
            * Number of exclamation marks ('!')
            * Number of apostrophes ('"', ''')
            * Number of parentheses couples ('()', '[]', '{}')

        Each * feature is normalized with respect to the text length
    in terms of characters.
```

```
17          :param data: text to analyse.
            :return: dictionary whose keys are feature names and whose
        values are their associated results
19          """

21          features = {
                'character_amount': lambda text: len(text),
23              'words_amount': lambda text: len(nltk.word_tokenize(text))
        ,
                'sentences_amount': lambda text: len(self.sent_detector.
        tokenize(text)),
25              'average_word_length': lambda text: self.
        average_word_length(text),
                'average_sentence_length': lambda text: self.
        average_sentence_length(text),
27              'long_words_percentage': lambda text: self.
        calculate_long_words_percentage(text),
                'pronouns_percentage': lambda text: self.
        calculate_pronouns_percentage(text),
29              'sentiment_words_percentage': lambda text: self.
        calculate_sentiment_words_percentage(text),
                'question_marks': lambda text: float(text.count('?')) /
        len(text),
31              'exclamation_marks': lambda text: float(text.count('!')) /
         len(text),
                'apostrophes': lambda text: (text.count('\"') + text.count
        ('\'')) / float(len(text)),
33              'parentheses': lambda text:
                (text.count('(') + text.count('[') + text.count('{')) /
        float(len(text))  # naive method
35          }

37          args = {
                'character_amount': [data],
39              'words_amount': [data],
                'sentences_amount': [data],
41              'average_word_length': [data],
                'average_sentence_length': [data],
43              'long_words_percentage': [data],
                'pronouns_percentage': [data],
45              'sentiment_words_percentage': [data],
                'question_marks': [data],
47              'exclamation_marks': [data],
                'apostrophes': [data],
49              'parentheses': [data]
```

```
        }
51
        return FunctionUtils.apply_functions(features, args)
```

Listing A.4: Funzione principale relativa all'estrazione della *feature repeated punctuation*, utilizzata in maniera analoga a quanto osservato per la *feature basic*

```
    def count_repeated_punctuation(self, text, targets=("?!", "??", "
    !!", "!?")):
2       """
        Counts how many times given sequences of symbols are repeated
    within given text.
4       Each sequence counter is normalized by the number of unigrams
    extracted from the same text.

6       :param text: text to analyse
        :param targets: sequences of symbols to count in the text
8       :return: dictionary that for each target sequence of symbols
    associates its normalized count
        """
10
        query = re.compile("[^\s\w]+")
12      repeated_punctuation = query.findall(text)

14      targets_count = {}

16      # Normalize
        for match in repeated_punctuation:
18          for target in targets:
                targets_count[target] = 0.
20              if match.startswith(target):
                    if target in targets_count:
22                      targets_count[target] += 1

24      unigrams = list(ngrams(nltk.word_tokenize(text), 1))

26      for key in targets_count:
            targets_count[key] = float(targets_count[key]) / len(
    unigrams)
28
        return targets_count
```

Listing A.5: Classe *DependencyVectorizer* in grado di estrarre tutte le tipologie di *dependency relation* definite.

```
class DependencyVectorizer(CountVectorizer, PickleCompliantVectorizer):
    """
    Extracts dependency features from given text.
    The following features are selected:

        1. Syntatic dependency features, i.e. triples like (head, relation, tail), where head and tail
         are words from text, whereas relation is the syntatic relation that the Stanford Dependency Parser
         has found between them.

        2. POS generalized dependency features, i.e. triples like the ones described in 1., but with the difference
        that the head element is substituted with its associated POS tag.

        3. Opinion generalized dependency features, i.e. triples like the ones described in 1., but with the difference
        that only triples containing sentiment words are selected. Moreover, sentiment words are replaced with their
        respective symbol ('+' for positive sentiment, '-' for negative sentiment)

    """

    [...]

    def _extract_syntactic_dependencies(self, text):
        """
        Extracts syntactic dependencies via Stanford Dependency Parser.
        Each dependency triple is structured as follow: (head, relation, tail)
        where head and tail are words extracted from text and relation is the dependency relation between them

        :param text: text to parse
        :return: list of syntactic dependencies triples
        """

        result = self.utils.dependency_parser.raw_parse(text)
```

```
        dependencies = result.next()

        syntactic_dependencies = []

        for triple in list(dependencies.triples()):
            syntactic_dependencies.append((triple[0][0], triple[1],
    triple[2][0]))

        print("Completed: {}".format(text))
        return ['-'.join(triple) for triple in syntactic_dependencies]

    def _extract_pos_generalized_dependencies(self, text):
        """
        Extracts POS generalized dependencies, i.e. syntactic
    dependencies whose head is replaced with its associated
        POS tag.

        :param text: text to parse
        :return: list of POS generalized dependencies triples
        """

        result = self.utils.dependency_parser.raw_parse(text)
        dependencies = result.next()

        pos_generalized_dependencies = []

        for triple in list(dependencies.triples()):
            pos_generalized_dependencies.append((triple[0][1], triple
    [1], triple[2][0]))

        return ['-'.join(triple) for triple in
    pos_generalized_dependencies]

    def _extract_opinion_generalized_dependencies(self, text, use_mpqa
    =False):
        """
        Extracts opinion generalized dependencies, i.e. syntactic
    dependencies whose head and/or tail are replaced
        with its/their associated sentiment symbols ('+' for positive
    sentiment, '-' for negative sentiment).
        If a certain triple has both head and tail with neutral
    sentiment, it is then not considered.
        Sentiment scores are computed via VADER analyzer or by
    consulting MPQA subjectivity lexicon.

```

```
        :param text: text to parse
69      :return: list of opinion generalized dependencies triples
        """
71
        triples = self._extract_syntactic_dependencies(text)
73
        # Loading lexicon
75      if use_mpqa:
            lexicon = self.utils.parse_mpqa_lexicon()
77          filter = lambda word: lexicon[word] if word in lexicon
    else None
            checker = lambda score: True if score is not None else
    False
79          mapper = lambda score: '+' if score == 'positive' else '-'
        else:   # use VADER
81          sid = SentimentIntensityAnalyzer()
            filter = lambda word: sid.polarity_scores(word)["compound"
    ]
83          checker = lambda score: True if abs(score) >= 0.5 else
    False
            mapper = lambda score, original: '+' if score >= 0.5 else
    '-' if score <= -0.5 else original
85
        opinion_generalized_dependencies = []
87
        for triple in triples:
89          head_score = filter(triple[0])
            tail_score = filter(triple[2])
91
            if checker(head_score) or checker(tail_score):
93              relation = triple[1]

95              head = mapper(head_score, triple[0])
                tail = mapper(tail_score, triple[2])
97
                opinion_generalized_dependencies.append((head,
    relation, tail))
99
        print(opinion_generalized_dependencies)
101     return ['-'.join(triple) for triple in
    opinion_generalized_dependencies]
```

Listing A.6: Funzioni principali relative all'estrazione della *feature cosine similarity*. In particolare il metodo *compute_ cosine_ similarity* invoca la

funzione *convert_ text_ to_ vec*, la quale si incarica di effettuare la codifica *word embedding* dei due documenti testuali presi in analisi.

```
    def compute_cosine_similarity(self, text1, text2):
        """
        Compute the cosine similarity between two texts.
        Wrapper function of _calculate_cosine_similarity

        :param text1:
        :param text2:
        :return: float value
        """

        vector1 = self.convert_text_to_vec(text1)
        vector2 = self.convert_text_to_vec(text2)

        return self.calculate_cosine_similarity(vector1, vector2)

    def convert_text_to_vec(self, text, dim=300, mode=op.add):
        """
        Maps a given plain text to its word vector obtained by
    combining each text token word vector
        by a given operator.

        :param text: text to convert to vector
        :param dim: dimension of each word vector
        :param mode: reduction operator to apply to list of vectors in
     order to obtain a single vector
        :return: vector representation of text
        """

        default_value = np.zeros(dim)

        if not text:
            print("Default vector returned")
            return default_value

        m = []

        tokens = 0

        for token in self.get_tokenized_lemmas(text):
            tokens += 1
            try:
                vec = self.w2v_model[token]
```

```
            except KeyError:
                vec = default_value

            m.append(vec)

        return functools.reduce(mode, m) if m else default_value
```

Listing A.7: Funzione relativa al calcolo effettivo della *feature cosine similarity*: dati due *word vector*, il metodo *calculate_ cosine_ similarity* restituisce il valore ottenuto dall'equazione 3.1.

```
    def calculate_cosine_similarity(self, u, v):
        """
        Computes the cosine similarity between two 1-D vectors, u and
    v"

        :param u: first 1-D vector
        :param v: second 1-D vector
        :return: cosine similarity of given 1-D vectors
        """

        cosine_similarity = np.dot(u, v) / (np.linalg.norm(u) * np.
    linalg.norm(v))
        return np.nan_to_num(cosine_similarity)
```

Listing A.8: Funzione principale relativa all'estrazione delle triple **SVO**. In particolare si fa riferimento ad una libreria esterna per l'estrazione delle singole triple.

```
    def extract_svo_triples(self, text):
        """
        Extract SVO triplets from given text by exploiting external
    SVO library

        :param text: text to parse in order to extract SVO triplets
        :return: list of SVO triples extracted from text
        """

        sentences = self.svo.sentence_split(text)
        triples = []
        for sent in sentences:
            root_tree = self.svo.get_parse_tree(sent)
            triples.append(self.svo.process_parse_tree(next(root_tree)
    ))
```

```
15          return triples
```

Listing A.9: Calcolo della relazione di *entailment* tra triple **SVO** appartenenti ad una coppia *evidence* - *claim*, mediante l'ausilio del **PPDB**. Nello specifico, la funzione *calc_ entailment_ vec* fa riferimento a quella usata da Ferreira e Vlachos[46].

```
1       def transform(self, triples_couple_list, **transform_params):
            """
3           Computes SVO entailment matrix. For each SVO triples couple
        obtained by the product of
            the first target SVO triples and the second's, the PPDB
        relationship data is used in order to
5           identify the entailment score between each S,V,O couple.

7           :param triples_couple_list: list of tuples containing targets
        svo triples
            :return: SVO entailment matrix
9           """

11          columns = 3*len(set(FeatureTransformerUtils.entailment_map.
        values()))
            mat = np.zeros((len(triples_couple_list), columns))
13
            for id, (target_1_svos, target_2_svos) in enumerate(
        triples_couple_list):
15              try:
                    vec = np.zeros((1, columns))
17
                    for (target1_svo, target2_svo) in itertools.product(
        target_1_svos, target_2_svos):
19
                        if not target1_svo or not target2_svo:
21                          continue

23                      entailments = {}
                        for key in target1_svo[0]:
25                          entailments[key] = FeatureTransformerUtils.
        calc_entailment_vec(target1_svo[0][key][0], target2_svo[0][key
        ][0])

27                      for index, key in zip(range(0, 3), entailments):
```

```
                    start_value = len(set(FeatureTransformerUtils.
entailment_map.values())) * index
                    end_value = start_value + len(set(
FeatureTransformerUtils.entailment_map.values()))
                    vec[0, start_value:end_value] += entailments[
key][0]

                mat[id, :] = vec

            except Exception, e:
                print(e)
                traceback.print_exc()
                pass

        return csr_matrix(mat)

    @staticmethod
    def calc_entailment_vec(v, w):
        """
        Computes the entailment vector between two words via
        Paraphrase Database.

        :param v: word
        :param w: word
        :return: integer vector
        """

        vec = np.zeros((1, len(set(FeatureTransformerUtils.
entailment_map.values()))))

        if v == w:
            vec[0, FeatureTransformerUtils.entailment_map['Equivalence
']] = 1
            return vec

        relationships = [(x, s, e) for (x, s, e) in
                         FeatureTransformerUtils.pre_loaded_data['ppdb
.pickle'].get(v, [])
                         if e in FeatureTransformerUtils.
entailment_map.keys() and x == w]
        if relationships:
            relationship = max(relationships, key=lambda t: t[1])[2]
            vec[0, FeatureTransformerUtils.entailment_map[relationship
]] = 1
```

```
65          return vec
```

A.2 Ottimizzazione del processo di elaborazione delle *feature*

Poiché la prevalenza delle *feature* impiegate per l'attività di classificazione richiede l'ausilio di strumenti di analisi esterni, comportando un'elevato consumo di risorse computazionali, le tempistiche nell'ambito dell'estrazione ed elaborazione delle *feature* risultano essere proibitive. Pertanto, è necessario definire un opportuno sistema di supporto atto a rendere maggiormente efficiente la presente procedura computazionale. Nello specifico, ogni *feature* di classificazione impiegata, ad eccezione degli *n-gram* e *skip n-gram*, viene salvata durante una fase preliminare di estrazione in maniera persistente nel formato *pickle*. In particolare, tutti gli estrattori sono organizzati secondo una struttura gerarchica in cui le classi di riferimento sono rappresentate dalle seguenti `PickleCompliantVectorizer` e `PickleCompliantExtractor`, specifiche rispettivamente per estrattori basati sulle classi proprie della libreria `sklearn`: `CountVectorizer` e `BaseEstimator` (codice A.10 e A.11). Così facendo, per tutte le successive operazioni di classificazione, quale, ad esempio, la *cross validation*, occorre solamente caricare preliminarmente in memoria i dati precedentemente salvati relativi alle *feature* impiegate.

Listing A.10: Classe *PickleCompliantExtractor* la quale consente di effettuare il salvataggio dei dati specifici di un determinato estratto in formato *pickle*.

```
1  class PickleCompliantExtractor(BaseCustomExtractor):
2
3      [...]
4
5      def _online_transform(self, data):
6          """
7          Abstract methods which defines the extractor on-line feature
       computation behaviour.
8
9          :param data: input data
10         :return: None
11         """
```

```
13          pass
15      def save_data_to_pickle(self, data):
            """
17          If 'save to pickle' mode is enabled, it saves computed into
        associated pickle file.
            If the pickle files already exists, the new content is
        appended to the existing one.
19
            :param data: computed content by extractor
21          :return: None
            """
23
            if self.save_to_pickle:
25              if self.pickle_folder is None:
                    pickle_to_verify = self.pickle_file
27              else:
                    pickle_to_verify = os.path.join(self.pickle_folder,
        self.pickle_file)
29
                if PickleUtility.verify_file(pickle_to_verify):
31                  existing_data = PickleUtility.load_pickle(self.
        pickle_file, folder=self.pickle_folder)
                    existing_data.update(data)
33                  PickleUtility.save_to_file(existing_data, self.
        pickle_file, folder=self.pickle_folder, mode='wb')
            else:
35              PickleUtility.save_to_file(data, self.pickle_file,
        folder=self.pickle_folder, mode='wb')
37      def _check_speedup(self, data):
            """
39          Checks whether the extractor is configured for on-line feature
         computation or it has to load
            pre-computed data only.
41          In particular if 'pickle mode' is enabled, it checks whether
        the data has already been pre-loaded externally.
            If yes, the pre-computed data is used in order to extract the
        data associated specific extractor features.
43          Alternatively, it loads the pre-computed data and builds the
        data associated specific extractor features.
            In the case 'pickle mode' is not enabled, the extractor
        proceeds the feature on-line computation.
45
```

```
        :param data: input data
47      :return: extractor data associated features.
        """
49
        if self.pickle_mode:
51          if self.pre_loaded:
                print("{}: Using pre-loaded data..".format(self.
    __class__.__name__))
53              return [FeatureExtractorUtils.pre_loaded_data[self.
    pickle_file][key] for key in data]
            else:
55              print(self.__class__.__name__, ": loading pre-computed
     data...")
                if self.pickle_folder:
57                  pre_computed_data = PickleUtility.load_pickle(os.
    path.join(self.pickle_folder, self.pickle_file))
                else:
59                  pre_computed_data = PickleUtility.load_all_pickles
    (self.pickle_file)
                return [pre_computed_data[key] for key in data]
61      else:
            print(self.__class__.__name__, ": on-line computation...")
63          return self._online_transform(data)
```

Listing A.11: Versione relativa ai *CountVectorizer* del *PickleCompliantExtractor*.

```
1   class PickleCompliantVectorizer(BaseCustomExtractor):

3       [...]

5       def _online_transform(self, data):
            [...]
7
        def save_data_to_pickle(self, data):
9           [...]

11      def _check_speedup_analyzer(self, doc):
            """
13          Equivalent behaviour of '_check_speedup()' function in the
        case of a CountVectorizer instance.

15          :param doc: input text
            :return: extractor data associated features.
17          """
```

```
19          if self.pickle_mode:
                if self.pre_loaded:
21                  # print("{}: Using pre-loaded data..".format(self.
    __class__.__name__))
                    return FeatureExtractorUtils.pre_loaded_data[self.
    pickle_file][doc]
23              else:
                    # print(self.__class__.__name__, ": loading pre-
    computed data...")
25                  return self.pickle_data[doc]
            else:
27              # print(self.__class__.__name__, ": on-line computation...
    ")
                return self._online_transform(doc)
```

A.3 La *pipeline* di classificazione

Dal punto di vista implementativo, una volta definiti tutti i processi associati all'estrazione delle *feature*, occorre delineare la struttura del classificatore, ossia modellare il problema dell'utilizzo simultaneo di molteplici matrici di valori, ognuna relativa ad una data *feature*. A tale proposito, si ricorre ancora una volta all'ausilio della libreria `sklearn`, la quale consente di definire una vera e propria *pipeline* di elaborazione dal testo. Nello specifico, la figura A.12 riporta le due funzioni principali volte a definire i classificatori introdotti all'interno dei capitolo sperimentali. In particolare modo, il metodo `_build_steps_with_args()` fa riferimento ad una specifica struttura dati (codice A.13), all'interno della quale si possono distinguere i selettori testuali, volti a definire il vero e proprio input per la computazione delle *feature*, e gli estrattori precedentemente introdotti all'interno della presente appendice.

Listing A.12: Funzioni principali relative alle definizione di un classificatore secondo il modello architetturale *pipeline*.

```
    def _build_steps_with_args(self):
2       """
        Instantiates each step of the pipeline associated
4       with the specified used features with their arguments
        respectively.
6
```

```
        :return: list of instantiated pipeline steps
        """

        instantiated_feature_list = []
        for block_name in self.features:
            args_list = self.args[block_name]
            steps = []
            for idx, step in enumerate(self.default_feature_dict[self.
features[block_name]]):
                args_dict = args_list[idx] if idx < len(args_list)
else {}
                steps.append(step(**args_dict))

            instantiated_feature_list.append(steps)

        return instantiated_feature_list

    def build(self, classifier, transformer_weights=None):
        """
        Builds a Pipeline instance from the given features list.
        For each feature list element, its pipeline steps are
retrieved. Successively, a pipeline is built via
        make_pipeline method. Then, a FeatureUnion object is created
according to the previously created
        list of pipelines. Lastly, the result pipeline is built by
invoking make_pipeline, taking the FeatureUnion
        object as the first step.

        :param classifier
        :param transformer_weights
        :return: pipeline object
        """

        instantiated_feature_list = self._build_steps_with_args()
        pipelines = []

        for block_name, steps in zip(self.features,
instantiated_feature_list):
            pipeline = make_pipeline(*steps)
            pipelines.append((block_name, pipeline))

        union = FeatureUnion(transformer_list=pipelines,
transformer_weights=transformer_weights, n_jobs=1)

        return make_pipeline(union, classifier)
```

Listing A.13: Struttura di supporto associante a ciascuna *feature* le relative fasi all'interno della *pipeline* per la sua estrazione.

```
        self.default_feature_dict =
        {
            'cosine_similarity': [CoupleSelector,
                                  CosineSimilarityExtractor,
                                  CosineSimilarityTransformer],

            'svo_triples': [CoupleSelector,
                            SVOTriplesExtractor,
                            SVOTriplesTransformer],

            'basic': [ItemSelector,
                      BasicExtractor,
                      DictVectorizer,
                      TfidfTransformer],

            'single_ngrams': [ItemSelector,
                              TfidfVectorizer],

            'single_skipgrams': [ItemSelector,
                                 SkipGramVectorizer,
                                 TfidfTransformer],

            'dependency_relations': [ItemSelector,
                                     DependencyVectorizer,
                                     TfidfTransformer],

            'repeated_punctuation': [ItemSelector,
                                     RepeatedPunctuationExtractor,
                                     DictVectorizer,
                                     TfidfTransformer],

            'discourse_cues': [ItemSelector,
                               DiscourseCueVectorizer,
                               TfidfTransformer],

            'sngrams': [ItemSelector,
                        SyntacticCountVectorizer,
                        TfidfTransformer],

              [...]
```

```
        }
```

Appendice B

Implementazione delle *feature* introdotte da Stab e Gurevych e del modello neurale basato su reti ricorrenti e *word embedding*

Analogamente a quanto osservato nella precedente appendice, si descrivono i frammenti di codice relativi alla definizione ed estrazione delle *feature* proposte da Stab e Gurevych[148]. Successivamente, viene discusso in maggior dettaglio il modello neurale introdotto in qualità di ulteriore classificatore.

B.1 Le *feature* di Stab e Gurevych

Così come quanto effettuato nell'ambito della *stance classification*, la definizione delle *feature* introdotte da Stab e Gurevych[148] ha seguito lo stesso procedimento implementativo. Più precisamente, la tutte le *feature* adottate, ad eccezione della *first word binary* e della *couple first word*, hanno richiesto il processo di ottimizzazione descritto nella sezione A.2 della precedente appendice. Inoltre, dal punto di vista della loro realizzazione, si è partiti dalla definizione delle *feauture* più semplici, ovvero *count token*, *difference token*, *punctuation mark* e infine *difference punctuation mark*. Successivamente, si è affrontato il problema dell'implementazione di *feature* leggermente più complesse, quali *word pairs*, *first word*

binary, *couple first word binary* e infine *production rule*, dove è stato necessario definire un'opportuna estensione della classe `CountVectorizer` (codice B.1, B.2, B.3 e B.4). Infine, le *feature modal verb* e *common element* sono state implementate per mezzo dell'ausilio dello strumento *WordNet*, mediante il quale è stato possibile estrarre i lemmi relativi alle parole osservate all'interno dei documenti testuali (codice B.5 e B.6).

Listing B.1: Classe *WordPairsBinaryVectorizer* relativa all'estrazione della *feature word pair*

```
class WordPairsBinaryVectorizer(CountVectorizer,
    PickleCompliantVectorizer):

    [...]

    def _extract_word_pairs(self, target1_tokens, target2_tokens,
    stop_words):
        """
        Computes all possible word pairs between two given token
    sequences.
        Tokens can be initially pre-processed by eliminating stop
    words if the latter is specified as a parameter

        :param target1_tokens:
        :param target2_tokens:
        :param stop_words:
        :return: iterator over word pairs
        """

        if stop_words is not None:
            target1_tokens = [token for token in target1_tokens if
    token not in stop_words]
            target2_tokens = [token for token in target2_tokens if
    token not in stop_words]

        return product(target1_tokens, target2_tokens)
```

Listing B.2: Classe *FirstWordBinaryExtractor* in grado di associare a ciascun documento testuale la sua prima parola valida, i.e. *token*.

```
class FirstWordBinaryExtractor(CountVectorizer):

    [...]

```

```
    def _extract_first_word(self, tokens, stop_words):
        """
        Extracts the first token from a given token sequence. The
    latter can be initially pre-processed by eliminating
         stop words if the associated parameter is specified.

        :param tokens:
        :param stop_words:
        :return: list containing the first word extracted
        """

        if stop_words is not None:
            tokens = [token for token in tokens if token not in
    stop_words]

        return [tokens[0]]
```

Listing B.3: Classe *FirstWordBinaryCoupleExtractor* relativa all'estrazione della *feature first word binary*.

```
class FirstWordBinaryCoupleExtractor(CountVectorizer):

    [...]

    def _extract_first_word_couple(self, target1_tokens,
    target2_tokens, stop_words):
        """
        Extracts a first word couple from two given token sequences.
    The latter can be initially pre-processed by
        eliminating stop words.

        :param target1_tokens:
        :param target2_tokens:
        :param stop_words:
        :return: list containing the extracted first word couple
        """

        if stop_words is not None:
            target1_tokens = [token for token in target1_tokens if
    token not in stop_words]
            target2_tokens = [token for token in target2_tokens if
    token not in stop_words]

        return [(target1_tokens[0], target2_tokens[0])]
```

Listing B.4: Classe *ProductionRulesBinaryVectorizer* in grado di estrarre tutte le *production rule* per ciascun documento testuale preso in analisi.

```
class ProductionRulesBinaryVectorizer(CountVectorizer,
    PickleCompliantVectorizer):

    [...]

    def _online_transform(self, doc):
        """
        Extracts production rules from given text. Stanford parser is
    used in order to retrieve production rules from
        text.

        :param doc:
        :return: list of extracted production rules
        """

        doc_production_rules = []
        for sentence in sent_tokenize(doc):
            tree = list(self.utils.parser.raw_parse(sentence))[0]
            production_rules = [rule.unicode_repr() for rule in tree.
    productions()]
            doc_production_rules += production_rules

        if self.pickle_data is None:
            self.pickle_data = {doc: doc_production_rules}
        else:
            self.pickle_data.update({doc: doc_production_rules})

        return doc_production_rules

    [...]
```

Listing B.5: Classe *ModalVerbBinaryExtractor* relativa all'estrazione della *feature modal verb*

```
class ModalVerbBinaryExtractor(PickleCompliantExtractor):

    def _online_transform(self, target_list):
        """
        Checks whether each element in given list contains any modal
    verb.

        :param target_list:
```

```
        :return: list of boolean values (1 or 0), where 1 means that
    the associated item (text) contains a modal verb,
        0 otherwise.
        """

        modal_vector = []
        modal_dict = {}

        for item in target_list:
            item_tokens = [self.utils.wordnet_lemmatizer.lemmatize(
    token).lower() for token in word_tokenize(item)]
            if any(modal in item_tokens for modal in
    FeatureExtractorUtils.modals):
                modal_vector.append(1)
                modal_dict[item] = 1
            else:
                modal_vector.append(0)
                modal_dict[item] = 0

        self.save_data_to_pickle(modal_dict)

        return modal_vector

    [...]
```

Listing B.6: Classe *CommonElementsExtractor* la quale consente di individuare l'insieme di parole in comune tra due documenti testuali presi in analisi. Nello specifico, ciascun *token* individuato è trasformato nel rispettivo lemma mediante l'ausilio dello sturmento *WordNet*.

```
class CommonElementsExtractor(PickleCompliantExtractor):

    def _online_transform(self, couple_list):
        """
        Counts all the common terms for each couple in given list.

        :param couple_list:
        :return: list of integer values, in which each number refers
    to the amount of common terms in the associated
        couple.
        """

        common_elements_vector = []
        common_elements_dict = {}
```

```
14
        for (target1, target2) in couple_list:
16          target1_tokens = [self.utils.wordnet_lemmatizer.lemmatize(
    token).lower()
                              for token in word_tokenize(target1)]
18          target2_tokens = [self.utils.wordnet_lemmatizer.lemmatize(
    token).lower()
                              for token in word_tokenize(target2)]
20
            common_tokens = list(set(target1_tokens).intersection(
    target2_tokens))
22          common_elements_vector.append(len(common_tokens))
            common_elements_dict[(target1, target2)] = len(
    common_tokens)
24
        self.save_data_to_pickle(common_elements_dict)
26
        return common_elements_vector
28
    [...]
```

B.2 Il modello neurale

Per quanto riguarda il modello introdotto di rete neurale basato su reti ricorrenti, nella fattispecie le **RNN**, si vuole dare una maggiore visione dell'architettura descritta graficamente all'interno della sezione 4.4.3. In particolar modo, la figura B.7 descrive l'implementazione del modello di rete neurale adottato, mentre la figura B.8 riporta un esempio di creazione di una sua istanza, in accordo con i parametri di configurazione richiesti. Infine, la figura B.9 riporta il frammento di codice relativo all'aggiunta della *stance* come ulteriore *feature* di classificazione. Si può osservare come la funzione `preprocess_sequence` racchiuda il processo di trasformazione della sequenza di documenti testuali nella loro rispettiva collezione di indici, i.e. `padded_docs` appartenenti ad un vocabolario di parole. Successivamente, si passa alla definizione della matrice di *word embedding*, i.e. `embedding_matrix`, mediante l'ausilio del modello strutturato *Word2Vec*. Infine, si può notare come anche la presente fase di calcolo supporti l'ottimizzazione tramite file *pickle*, in quanto il ca-

ricamento in memoria della struttura *Word2Vec* richiede ingenti risorse computazionali.

Listing B.7: Modello neurale basato su reti ricorrenti e *word embedding*.

```
class ArgumentStructurePrediction_RNN(object):

    def build_model(self):
        branches = []

        for idx in self.ids:
            model = Sequential()
            model.add(Embedding(self.vocab_size[idx], self.
    embedding_vector_length[idx], weights=[self.weights[idx]],
                                input_length=self.input_length[idx],
    trainable=False))
            if self.add_dropout:
                model.add(Dropout(self.dropout))

            model.add(LSTM(self.lstm_neurons[idx], recurrent_dropout=
    self.recurrent_dropout, dropout=self.dropout))

            if self.add_dropout:
                model.add(Dropout(self.dropout))

            branches.append(model)

        merged = Sequential()
        merged.add(Merge(branches, mode='concat'))
        merged.add(Dense(1, activation='sigmoid'))

        merged.compile(loss=self.loss, optimizer=self.optimizer,
    metrics=self.metrics)
        return merged
```

Listing B.8: Esempio di instanziazione del modello neurale.

```
classifier = ArgumentStructurePrediction_RNN(
  ids=['Claim', 'Evidence'],
  vocab_size={'Claim': 6267,
              'Evidence': 8752},
  input_length={'Claim': 11,
                'Evidence': 29},
  embedding_vector_length={'Claim': 300,
                           'Evidence': 300},
```

```
  weights={'Claim': np.zeros((6267, 300)),
           'Evidence': np.zeros((8752, 300))},
  lstm_neurons={'Claim': 100,
                'Evidence': 100},
  metrics=['accuracy']).build_model()
```

Listing B.9: Processo di aggiunta della *stance* all'interno delle sequenze di indici rappresentanti l'input per il livello di *Embedding*.

```
def preprocess_sequence(sequence, model, key, load_pickle=False,
    pickle_file=None,
                        vocab_size=None, padding_max_length=None,
    add_stance=False, stance_vector=None,
                        embedding_matrix=None):

    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(sequence)
    if vocab_size is None:
        vocab_size = len(tokenizer.word_index) + 1
    encoded_data = tokenizer.texts_to_sequences(sequence)
    if padding_max_length is None:
        padding_max_length = sum([len(row) for row in encoded_data]) /
     len(encoded_data)
    print("Padding max length: {}".format(padding_max_length))
    padded_docs = pad_sequences(encoded_data, maxlen=
    padding_max_length, padding='post')

    # Add stances values in padded_docs: shifting indexes by 3, i.e.
    reserving values for stances
    if add_stance:
        padding_max_length += 1
        vocab_size += 3
        new_padded_docs = np.zeros((padded_docs.shape[0], padded_docs.
    shape[1] + 1),
                                   dtype=np.int32)
        for idx, row in enumerate(padded_docs):
            new_padded_docs[idx] = [val + 3 if val != 0 else val for
    val in row] + [stance_vector[key][idx]]
        padded_docs = new_padded_docs

    if not load_pickle:
        if embedding_matrix is None:
            embedding_matrix = np.zeros((vocab_size, 300))
        for word, i in tokenizer.word_index.items():
            try:
```

```
                embedding_vector = model[word]
            except KeyError:
                embedding_vector = np.zeros(300)
            if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector

        if pickle_file is not None:
            if PickleUtility.verify_file(pickle_file):
                existing_data = PickleUtility.load_pickle(pickle_file)
                existing_data.update({key: embedding_matrix})
                PickleUtility.save_to_file(existing_data, pickle_file)
            else:
                PickleUtility.save_to_file({key: embedding_matrix},
    pickle_file)
    else:
        embedding_matrix = PickleUtility.load_pickle(pickle_file)[key]

    return vocab_size, padding_max_length, padded_docs,
    embedding_matrix

[...]

# Step 0: Retrieving positive and negative examples
 training_data_path = os.path.join(POST_PROCESS_DIR,
                                   '
    ce_emnlp_15_argument_structure_prediction_training.csv')
data = pd.read_csv(training_data_path, encoding="utf-8")
target = data['Argument relation'].values

claim_data = data['Claim'].values
evidence_data = data['Evidence'].values

add_stance_mode = 'y'

if add_stance_mode.lower() == 'y':
    claim_stances = data['Claim stance']
    evidence_stances = data['Evidence stance']

    claim_stances = [1 if val == 'for' else 2 if val == 'against' else
     3 for val in claim_stances]
    evidence_stances = [1 if val == 'for' else 2 if val == 'against'
    else 3 for val in evidence_stances]

    add_stance = True
    stance_vector = {'Claim': claim_stances,
```

```
                    'Evidence': evidence_stances}
else:
     add_stance = False
     stance_vector = None

# Step 1: Loading Word2Vec data
[...]

# Step 2: Pre-processing data for classifier

# Claim
claim_vocab_size, claim_padding_max_length, claim_padded_docs,
    claim_embedding_matrix = \
    preprocess_sequence(claim_data, utils.w2v_model, pickle_file=
    pickle_file, key='Claim',
                        load_pickle=load_pickle, add_stance=add_stance
    , stance_vector=stance_vector)

 # Evidence
evidence_vocab_size, evidence_padding_max_length, evidence_padded_docs
    , evidence_embedding_matrix = \
    preprocess_sequence(evidence_data, utils.w2v_model, pickle_file=
    pickle_file, key='Evidence',
                        load_pickle=load_pickle, add_stance=add_stance,
     stance_vector=stance_vector)

[...]
```

Appendice C

Il test di *Leave One Topic Out* e le informazioni relative ai test di comparazione

In quest'ultima appendice vengono riportati alcuni dettagli importanti relativi al test di *Leave One Topic Out*, al fine di favorirne la comprensione, e le informazioni per esteso relative ai test di comparazione tra il data-set **CE-EMNLP-15**, ristretto ai soli *topic* appartenenti al set di *train and test*, e il corpus definito mediante l'ausilio dello strumento di analisi **MARGOT**.

C.1 Il test *Leave One Topic Out*

Per quanto riguarda il test effettuato sul data-set **CE-EMNLP-15** nell'ottica dei *topic* appartenenti al set di *train and test*, previa fase di calibrazione sulla controparte relativa ai *topic* del set di *held-out*, è opportuno soffermarsi sulla realizzazione del principio di funzionamento descritto nella sezione 5.5. In particolar modo, come si può osservare dalla figura C.1, la definizione dei molteplici set di *training* e di *test* risulta essere molto semplice. Nello specifico, si ricorre all'ausilio della classe `LeaveOneGroupOut` offerta dalla libreria `sklearn`, mediante la quale i dati di input vengono separati nei set di interesse sulla base di un parametro in comune, per mezzo del quale è possibile organizzare i dati in gruppi. In particolare si utilizza il *topic* come valore di riferimento atto

a discriminare le coppie *evidence* - *claim*. Infine, si procede con la fase di misurazione delle performance C.1.

Listing C.1: Test *Leave One Topic Out* su data-set **CE-EMNLP-15**.

```
[...]

logo = LeaveOneGroupOut()
topic_map = {topic: idx for idx, topic in enumerate(np.unique(
    train_data['Topic'].values))}
groups = [topic_map[topic] for topic in train_data['Topic'].values]
n_splits = logo.get_n_splits(X=train_data, y=train_target, groups=
    groups)

for idx, (train_index, test_index) in enumerate(logo.split(X=
    train_data, y=train_target, groups=groups)):
  [...]

  X_train, X_test = train_data.iloc[train_index], train_data.iloc[
    test_index]
    y_train, y_test = train_target[train_index], train_target[
    test_index]
    pipeline.fit(X=X_train, y=y_train)
    predictions = pipeline.predict(X_test)

    f1 = f1_score(y_pred=predictions, y_true=y_test, pos_label='link')
    accuracy = accuracy_score(y_pred=predictions, y_true=y_test)
    print("Fold scores:\n F1: {}\nAccuracy: {}".format(f1, accuracy))
    scores[group_excluded] = {'f1-score': f1, 'accuracy': accuracy}

# [Step 4]: Viewing classification results
print("All scores: {}".format(scores))
print("Average f1: {}".format(np.mean([scores[idx]['f1-score'] for idx
     in scores])))
print("Average accuracy: {}".format(np.mean([scores[idx]['accuracy']
    for idx in scores])))
```

C.2 Informazioni relative ai test di comparazione

Al fine di focalizzare l'attenzione verso i principi e le motivazioni dietro all'impiego dei vari esperimenti descritti all'interno del presente elabo-

rato, si è deciso di rilegare in questo spazio la descrizione dettagliata di alcune informazioni di riferimento. Nello specifico, la tabella C.1 riporta l'elenco completo dei *topic* appartenenti ai set di *train and test* e di *held-out*, in modo tale da favorire la comprensione dei grafici riportati nella sezione 5.5. A tal proposito, la tabella C.2 riporta i risultati dettagliati per ciascun *topic* di ogni classificatore impiegato nell'ambito del test *Leave One Topic Out*, con particolare riferimento alle metriche principali, quali *accuracy* e *f1-score*. La tabella C.3 riporta le stesse metriche precedentemente evidenziate nell'ambito del test di predizione della *argument relation* sullo stesso data-set da parte della stessa tipologia di classificatori, previa fase di *training* sul nuovo corpus definito per mezzo dello strumento **MARGOT**. Infine, le tabelle C.4, C.5, C.6, C.7, C.8 e C.9 contengono, rispettivamente, i valori relativi al *report* di classificazione di ciascun classificatore e le relative matrici di confusione.

Tabella C.1: Elenco dei *topic* appartenenti al data-set **CE-EMNLP-15**. Nello specifico, per ognuno di essi sono riportati il loro identificativo e set di appartenenza: *train and test* o *held-out*.

Topic id	Topic	Data-set
1	This house believes that the sale of violent video games to minors should be banned	train and test
21	This house supports the one-child policy of the republic of China	train and test
61	This house would permit the use of performance enhancing drugs in professional sports	train and test
81	This house would make physical education compulsory	train and test
101	This house believes in the use of affirmative action	train and test
121	This house would ban boxing	train and test
181	This house would embrace multiculturalism	train and test
221	This house would ban gambling	train and test
323	This house believes that the right to asylum should not be absolute	train and test
381	This house would abolish the monarchy	train and test
641	This house would abolish all collective bargaining rights claimed by trades unions	train and test
642	This house would ban partial birth abortions	train and test

Continua nella pagina seguente

Tabella C.1 – *Continuazione della pagina precedente.*

Topic id	Topic	Data-set
643	This house believes that countries with an imbalanced male/female ratio skewed towards males should encourage parents to produce girls	train and test
644	This house would introduce year round schooling	train and test
645	This house would subsidize poor communities	train and test
646	This house believes the United States is responsible for Mexico's drugs war	train and test
647	This house believes the US is justified in using force to prevent states from acquiring nuclear weapons	train and test
648	This house believes atheism is the only way	train and test
662	This house believes that democratic governments should require voters to present photo identification at the polling station	train and test
663	This house would reintroduce national service	train and test
664	This house prefers trade to aid	train and test
665	This house believes that housewives should be paid for their work	train and test
681	This house would abolish intellectual property rights	train and test
683	This house believes that wind power should be a primary focus of future energy supply	train and test
701	This house believes that endangered species should be protected	train and test
721	This house believes that Europe should weaken its austerity measures to guarantee its citizens greater social support	train and test
742	This house would limit the right to bear arms	train and test
743	This house believes that bribery is sometimes acceptable	train and test
744	This house would re-engage with Myanmar	train and test
761	This house would build the Keystone XL pipeline	train and test
821	This house believes that opinion polls harm the democratic process	train and test
861	This house believes that Israel should lift the blockade of Gaza	train and test
921	This house believes that the Church of England should be separated from the British state	train and test
923	This house would encourage the creation of private universities in the UK	train and test

Continua nella pagina seguente

Tabella C.1 – *Continuazione della pagina precedente.*

Topic id	Topic	Data-set
925	This house would use foreign aid funds to research and distribute software that allows bloggers and journalists in non-democratic countries to evade censorship and conceal their online activities	train and test
941	This house would only teach abstinence for sex education in schools	train and test
943	This house would remove United States military bases from Japan	train and test
945	This house believes that male infant circumcision is tantamount to child abuse	train and test
947	This house would ban all unsustainable logging	train and test
441	This house believes that open primaries are the most effective method of selecting candidates for elections	held-out
442	This house believes that the Catholic Church is justified in forbidding the use of barrier methods of contraception	held-out
443	This house supports raising the school leaving age to 18	held-out
481	This house believes that the leaking of military documents by Anat Kamm was justified	held-out
482	This house believes that states should not subsidize the growing of tobacco	held-out
483	This house believes that it is sometimes right for the government to restrict freedom of speech	held-out
601	This house would pass the American Jobs Act	held-out
602	This house would fund education using a voucher scheme	held-out
621	This house believes all nations have a right to nuclear weapons	held-out
801	This house believes that Israel's 2008-2009 military operations against Gaza were justified	held-out
803	This house would build high rises for housing	held-out
841	This house would criminalize blasphemy	held-out
881	This house believes that Holocaust denial should be a criminal offence	held-out
902	This house supports direct election of city mayors	held-out
926	This house would disband ASEAN	held-out
944	This house would prohibit burning the stars and stripes	held-out
946	This house would implement playoffs in collegiate level American football	held-out
961	This house believes social deprivation causes crime	held-out
942	This house would enforce term limits on the legislative branch of government	held-out

Tabella C.2: Risultati relativi a ciascun classificatore proposto ottenuti al termine del test *Leave One Topic Out*. In particolare, si riportano le informazioni associate alle metriche di riferimento *accuracy* e *f1-score*. Infine, il campo *topic graphic id* fa riferimento ai valori delle ascisse riportate nei grafici esposti all'interno del capitolo 5.

Topic graph id	Topic id	Baseline Cosine similarity		Baseline n-gram		Stab and Gurevych		RNN (d = 0.2)		SC	
		acc	f1	acc	f1	acc	f1	acc	f1	acc	f1
1	1	**0.948**	0.0299	0.5593	0.0999	0.6777	**0.1542**	0.6943	0.0696	0.4918	0.1327
2	21	**0.9578**	**0.2486**	0.741	0.0577	0.7356	0.1296	0.3996	0.0634	0.8835	0.2175
3	61	**0.8358**	0.1752	0.6919	0.1382	0.7224	0.1803	0.407	0.113	0.8183	**0.2138**
4	81	0.4511	0.1412	**0.8904**	0.0893	0.189	0.1128	0.8088	0.0918	0.8357	**0.1639**
5	101	0.3826	0.0672	0.8082	0.0711	0.949	**0.1511**	0.2669	0.0588	**0.8385**	0.1053
6	121	0.251	0.1618	0.8072	0.0943	**0.8373**	**0.2832**	0.6466	0.1111	0.6566	0.1896
7	181	0.3088	0.0981	0.7207	0.0926	**0.8552**	**0.2032**	0.8749	0.0398	0.6293	0.132
8	221	0.4742	0.1077	0.4543	0.0797	0.5159	0.1139	0.4611	0.0587	**0.7138**	**0.1636**
9	323	**0.9621**	0.0702	0.9091	0.1119	0.8812	**0.217**	0.6392	0.0667	0.8375	0.1922
10	381	**0.7362**	**0.1942**	0.6515	0.0886	0.3379	0.1135	0.3623	0.0879	0.6981	0.1739
11	641	0.4898	0.2424	**0.8231**	0.1875	0.7823	0.3043	0.4694	0.2041	0.7143	**0.3438**
12	642	0.7297	**0.2075**	0.6776	0.0841	0.2523	0.121	**0.7921**	0.0583	0.7394	0.198
13	643	0.52	0.4	**0.8**	0.0	**0.8**	**0.5455**	0.44	0.3	0.72	0.5333
14	644	0.4486	0.3371	**0.7477**	0.129	0.4953	**0.3415**	0.6075	0.3	0.6729	0.4615
15	645	**0.9499**	0.0286	0.7883	0.0465	0.8614	**0.2824**	0.7928	0.0602	0.6822	0.1696
16	646	0.252	0.3072	0.7089	0.1095	**0.7268**	0.3588	0.6016	0.1695	0.5984	**0.3871**
17	647	**0.9095**	0.3684	0.7789	0.0538	0.902	**0.3906**	0.4435	0.1811	0.5126	0.195
18	648	**0.9677**	0.1887	0.6363	0.0445	0.966	**0.2398**	0.3037	0.052	0.6736	0.0864
19	662	0.4074	0.3846	0.7037	0.2	**0.7407**	0.3636	0.7037	0.0	0.6667	**0.4706**
20	663	0.6377	0.239	0.6916	0.1043	**0.8862**	**0.4571**	0.6317	0.0889	0.7814	0.354
21	664	**0.9579**	**0.2038**	0.3779	0.0704	0.9155	0.1824	0.682	0.0426	0.7264	0.1249
22	665	**0.6154**	0.4444	0.4615	0.3636	0.3077	**0.4706**	0.4615	0.4615	0.5385	0.5
23	681	0.3346	0.1176	0.7618	0.0679	**0.8968**	**0.3354**	0.7994	0.0877	0.8602	0.2328
24	683	**0.9148**	0.1857	0.5643	0.1414	0.509	0.1673	0.3984	0.1105	0.7399	**0.2127**
25	701	**0.9231**	0.087	0.6996	0.1277	0.9048	**0.381**	0.5641	0.1439	0.7985	0.3373
26	721	0.7647	**0.3333**	0.6	0.1282	**0.8412**	0.2703	0.5	0.2056	0.5941	0.2418
27	742	0.7217	**0.1959**	0.6318	0.0899	**0.8337**	0.2259	0.5059	0.0797	0.7503	0.1822
28	743	**0.8571**	0.4	0.6905	0.2353	0.8095	**0.4286**	0.5714	0.1	0.7143	0.25
29	744	0.2434	0.1984	0.6891	0.1263	0.3783	0.1942	0.5993	0.1301	**0.8015**	**0.3117**
30	761	**0.5844**	**0.2982**	0.5013	0.2	0.5481	0.2564	0.2364	0.1923	0.5714	0.2857
31	821	0.4368	**0.3636**	0.5862	0.28	**0.7586**	0.3226	0.5057	0.2712	0.3448	0.3596
32	861	0.435	0.0831	0.6385	0.0688	0.6578	**0.1214**	0.1445	0.0702	**0.6809**	0.1053

Continua nella pagina seguente

Tabella C.2 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Baseline Cosine similarity		Baseline n-gram		Stab and Gurevych		RNN (d = 0.2)		SC	
33	921	0.4239	0.4176	0.5978	0.1395	0.2989	**0.4215**	0.5707	0.313	**0.6467**	0.3434
34	923	**0.8462**	0.6111	0.7033	0.1818	0.8132	0.6047	0.7033	0.129	0.8022	**0.64**
35	925	0.2	0.3333	0.5333	0.1765	0.2	0.3333	**0.6833**	0.1739	0.5833	**0.3902**
36	941	0.5522	0.1221	**0.7128**	0.0452	0.5598	0.1183	0.4894	0.0575	0.6761	**0.1446**
37	943	**0.8352**	**0.5455**	0.7582	0.0	0.6703	0.4828	0.4066	0.3721	0.6813	0.5246
38	945	**0.9663**	0.127	0.6773	0.0765	0.8058	**0.1317**	0.9173	0.0232	0.6646	0.1192
39	947	**0.9422**	0.0519	0.7173	0.0916	0.8868	0.1437	0.4078	0.1095	0.6572	**0.1846**

Tabella C.3: Risultati ottenuti da ciascun classificatore proposto nell'ambito della predizione della *argument relation* dei dati esposti all'interno del data-set **CE-EMNLP-15**, in seguito ad una fase preliminare di *training* sul corpus definito mediante l'ausilio di **MARGOT**. Nello specifico, si riportano i valori relativi alle metriche *accuracy* e *f1-score*.

Topic graph id	Topic id	Baseline Cosine similarity		Baseline n-gram		Stab and Gurevych		RNN		SC	
		acc	f1	acc	f1	acc	f1	acc	f1	acc	f1
1	1	0.0518	0.0985	0.0633	0.0988	0.0857	0.1014	**0.2128**	0.1013	0.1516	**0.1056**
2	21	0.0352	0.0681	0.1155	0.0701	0.1206	0.0704	0.1669	0.0668	**0.1755**	**0.0735**
3	61	0.0581	0.1099	0.0872	0.1105	0.1323	0.1156	0.1686	0.1062	**0.1933**	**0.1176**
4	81	0.0526	0.1	0.1074	0.1016	0.2095	0.1068	**0.3147**	0.0989	0.2868	**0.1147**
5	101	0.0309	0.0599	0.1002	0.06	0.1011	0.0609	0.0926	0.0608	**0.1807**	**0.0632**
6	121	0.0783	0.1453	0.1526	0.1457	0.2008	**0.1568**	**0.3454**	0.1466	0.2651	0.1488
7	181	0.046	0.088	0.0522	0.0879	0.0583	0.0885	0.0512	0.0879	**0.0703**	**0.0901**
8	221	0.0417	0.08	0.094	0.0796	0.0821	0.081	**0.2495**	0.0808	0.1537	**0.0811**
9	323	0.0379	0.0731	0.0644	0.0737	0.1009	0.0751	**0.1589**	0.0726	0.1539	**0.0766**
10	381	0.0498	0.0949	0.0879	0.0927	0.1642	0.0962	**0.2013**	0.0894	0.107	**0.0965**
11	641	0.102	0.1852	0.102	0.1852	0.102	0.1852	**0.1224**	**0.1887**	0.1156	0.1875
12	642	0.0528	0.1002	0.3797	0.0957	0.3977	0.117	0.2728	0.1017	**0.6564**	**0.147**
13	643	0.2	**0.3333**	0.2	**0.3333**	0.2	**0.3333**	0.2	**0.3333**	**0.24**	0.2963
14	644	0.1495	**0.2602**	0.1495	**0.2602**	0.1495	**0.2602**	0.1776	0.2542	**0.1963**	0.2586
15	645	0.0501	0.0955	0.0524	0.0957	0.0568	0.0961	**0.0951**	0.0931	0.0811	**0.0984**
16	646	0.1805	0.3058	0.1919	0.3049	0.1854	0.3032	**0.3171**	**0.3137**	0.2211	0.3108
17	647	0.0955	0.1743	0.0955	0.1743	0.0955	0.1743	**0.1106**	**0.1767**	0.0967	0.1745
18	648	0.0243	0.0475	0.0431	0.0466	0.0623	0.0485	**0.1341**	0.0474	0.1257	**0.0514**
19	662	**0.2593**	**0.4118**	**0.2593**	**0.4118**	**0.2593**	**0.4118**	**0.2593**	**0.4118**	**0.2593**	**0.4118**

Continua nella pagina seguente

Tabella C.3 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Baseline Cosine similarity		Baseline n-gram		Stab and Gurevych		RNN		SC	
20	663	0.0838	0.1547	0.0928	0.156	0.1168	0.1547	0.1677	0.1576	**0.1856**	**0.1707**
21	664	0.0377	0.0726	0.0582	0.0729	0.1191	**0.0775**	**0.1682**	0.0762	0.0993	0.0766
22	665	0.3077	0.4706	0.3077	0.4706	0.3846	**0.5**	**0.6154**	0.4444	0.3846	**0.5**
23	681	0.0492	0.0938	0.053	0.0941	0.0974	**0.0965**	**0.1128**	0.0963	0.1051	0.0955
24	683	0.0583	0.1102	0.0942	0.1127	0.1114	0.1133	**0.1241**	0.1148	0.1151	**0.1151**
25	701	0.0769	0.1429	0.1026	0.1404	0.1868	0.1527	0.1758	0.1445	**0.2637**	**0.166**
26	721	0.1176	0.2105	0.1176	0.2105	0.1176	0.2105	0.1294	**0.2128**	**0.1765**	0.2045
27	742	0.0492	0.0938	0.0681	0.0937	0.0803	0.0961	**0.1991**	**0.1009**	0.132	0.0999
28	743	0.1429	0.25	0.1429	0.25	0.2857	0.2857	0.2619	0.2439	**0.3333**	**0.3**
29	744	0.0936	0.1712	0.0974	0.1718	0.0974	0.1718	**0.1236**	**0.1761**	0.1161	0.1748
30	761	0.1039	0.1882	0.1325	0.1854	0.1039	0.1882	0.1091	**0.1891**	**0.1481**	0.1881
31	821	0.1954	0.3269	0.1954	0.3269	0.1954	0.3269	0.1954	0.3269	**0.2069**	**0.3301**
32	861	0.0368	0.0709	0.0917	0.0695	0.0708	0.0722	**0.1356**	0.0689	0.1093	**0.0726**
33	921	0.2609	**0.4138**	0.2609	**0.4138**	0.2609	**0.4138**	**0.3261**	0.3861	0.2609	**0.4138**
34	923	0.2088	0.3455	0.2088	0.3455	**0.4505**	**0.4318**	**0.4505**	0.3243	0.3297	0.3838
35	925	0.2	0.3333	0.2	0.3333	0.2	0.3333	**0.2333**	**0.3429**	0.2	0.3333
36	941	0.0444	0.085	0.0923	0.0738	0.124	**0.0862**	0.1408	0.0808	**0.1737**	0.0834
37	943	0.2308	0.375	0.3407	0.3478	0.3407	0.3617	0.4066	0.3721	**0.5604**	**0.4444**
38	945	0.0328	0.0634	0.1276	0.0657	0.1276	0.0654	0.2152	0.0649	**0.2538**	**0.0723**
39	947	0.0578	0.1093	0.1069	0.1118	0.1568	0.1118	0.1338	0.1091	**0.1797**	**0.119**

Tabella C.4: *Report* di classificazione di ciascun *topic* per la *baseline* basata sugli *n-gram*.

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
		link	no-link	link	no-link	link	no-link	link	no-link
1	1	0.052	0.9633	0.9912	0.0127	0.0988	0.025	453.0	8294.0
2	21	0.0364	0.9774	0.9459	0.0852	0.0701	0.1567	111.0	3040.0
3	61	0.0586	0.9545	0.975	0.0324	0.1105	0.0627	40.0	648.0
4	81	0.0537	0.9636	0.9592	0.0601	0.1016	0.1131	49.0	882.0
5	101	0.031	0.9707	0.9301	0.0738	0.06	0.1371	143.0	4488.0
6	121	0.0791	0.9302	0.9231	0.0871	0.1457	0.1594	39.0	459.0
7	181	0.046	0.9524	0.993	0.0068	0.0879	0.0135	142.0	2944.0
8	221	0.0416	0.9565	0.9403	0.0571	0.0796	0.1078	67.0	1540.0
9	323	0.0383	0.9744	0.9811	0.0283	0.0737	0.055	53.0	1344.0

Continua nella pagina seguente

Tabella C.4 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
10	381	0.0488	0.9286	0.9362	0.0435	0.0927	0.0831	47.0	897.0
11	641	0.102	0.0	1.0	0.0	0.1852	0.0	15.0	132.0
12	642	0.0518	0.9456	0.622	0.3662	0.0957	0.5279	82.0	1472.0
13	643	0.2	0.0	1.0	0.0	0.3333	0.0	5.0	20.0
14	644	0.1495	0.0	1.0	0.0	0.2602	0.0	16.0	91.0
15	645	0.0503	1.0	1.0	0.0023	0.0957	0.0046	68.0	1288.0
16	646	0.1805	0.8182	0.982	0.0179	0.3049	0.035	111.0	504.0
17	647	0.0955	0.0	1.0	0.0	0.1743	0.0	76.0	720.0
18	648	0.0239	0.9534	0.9604	0.0202	0.0466	0.0396	227.0	9100.0
19	662	0.2593	0.0	1.0	0.0	0.4118	0.0	7.0	20.0
20	663	0.0846	1.0	1.0	0.0098	0.156	0.0194	28.0	306.0
21	664	0.0378	0.9692	0.9821	0.022	0.0729	0.0431	112.0	2860.0
22	665	0.3077	0.0	1.0	0.0	0.4706	0.0	4.0	9.0
23	681	0.0494	1.0	1.0	0.0041	0.0941	0.0081	51.0	986.0
24	683	0.0598	0.98	0.9872	0.0389	0.1127	0.0748	78.0	1260.0
25	701	0.0758	0.8889	0.9524	0.0317	0.1404	0.0613	21.0	252.0
26	721	0.1176	0.0	1.0	0.0	0.2105	0.0	20.0	150.0
27	742	0.0492	0.9492	0.9783	0.021	0.0937	0.0411	138.0	2665.0
28	743	0.1429	0.0	1.0	0.0	0.25	0.0	6.0	36.0
29	744	0.094	1.0	1.0	0.0041	0.1718	0.0082	25.0	242.0
30	761	0.1027	0.8667	0.95	0.0377	0.1854	0.0722	40.0	345.0
31	821	0.1954	0.0	1.0	0.0	0.3269	0.0	17.0	70.0
32	861	0.0361	0.9533	0.9231	0.0599	0.0695	0.1128	468.0	12264.0
33	921	0.2609	0.0	1.0	0.0	0.4138	0.0	48.0	136.0
34	923	0.2088	0.0	1.0	0.0	0.3455	0.0	19.0	72.0
35	925	0.2	0.0	1.0	0.0	0.3333	0.0	12.0	48.0
36	941	0.0386	0.8722	0.8145	0.0588	0.0738	0.1102	248.0	5341.0
37	943	0.2254	0.75	0.7619	0.2143	0.3478	0.3333	21.0	70.0
38	945	0.034	0.9789	0.9363	0.1002	0.0657	0.1818	267.0	7884.0
39	947	0.0593	0.9697	0.9726	0.0538	0.1118	0.1019	73.0	1190.0

Tabella C.5: *Report* di classificazione di ciascun *topic* per la *baseline* basata sulla *cosine similarity*.

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
		link	no-link	link	no-link	link	no-link	link	no-link
1	1	0.0518	0.0	1.0	0.0	0.0985	0.0	453.0	8294.0
2	21	0.0352	0.0	1.0	0.0	0.0681	0.0	111.0	3040.0
3	61	0.0581	0.0	1.0	0.0	0.1099	0.0	40.0	648.0
4	81	0.0526	0.0	1.0	0.0	0.1	0.0	49.0	882.0
5	101	0.0309	0.0	1.0	0.0	0.0599	0.0	143.0	4488.0
6	121	0.0783	0.0	1.0	0.0	0.1453	0.0	39.0	459.0
7	181	0.046	0.0	1.0	0.0	0.088	0.0	142.0	2944.0
8	221	0.0417	0.0	1.0	0.0	0.08	0.0	67.0	1540.0
9	323	0.0379	0.0	1.0	0.0	0.0731	0.0	53.0	1344.0
10	381	0.0498	0.0	1.0	0.0	0.0949	0.0	47.0	897.0
11	641	0.102	0.0	1.0	0.0	0.1852	0.0	15.0	132.0
12	642	0.0528	0.0	1.0	0.0	0.1002	0.0	82.0	1472.0
13	643	0.2	0.0	1.0	0.0	0.3333	0.0	5.0	20.0
14	644	0.1495	0.0	1.0	0.0	0.2602	0.0	16.0	91.0
15	645	0.0501	0.0	1.0	0.0	0.0955	0.0	68.0	1288.0
16	646	0.1805	0.0	1.0	0.0	0.3058	0.0	111.0	504.0
17	647	0.0955	0.0	1.0	0.0	0.1743	0.0	76.0	720.0
18	648	0.0243	0.0	1.0	0.0	0.0475	0.0	227.0	9100.0
19	662	0.2593	0.0	1.0	0.0	0.4118	0.0	7.0	20.0
20	663	0.0838	0.0	1.0	0.0	0.1547	0.0	28.0	306.0
21	664	0.0377	0.0	1.0	0.0	0.0726	0.0	112.0	2860.0
22	665	0.3077	0.0	1.0	0.0	0.4706	0.0	4.0	9.0
23	681	0.0492	0.0	1.0	0.0	0.0938	0.0	51.0	986.0
24	683	0.0583	0.0	1.0	0.0	0.1102	0.0	78.0	1260.0
25	701	0.0769	0.0	1.0	0.0	0.1429	0.0	21.0	252.0
26	721	0.1176	0.0	1.0	0.0	0.2105	0.0	20.0	150.0
27	742	0.0492	0.0	1.0	0.0	0.0938	0.0	138.0	2665.0
28	743	0.1429	0.0	1.0	0.0	0.25	0.0	6.0	36.0
29	744	0.0936	0.0	1.0	0.0	0.1712	0.0	25.0	242.0
30	761	0.1039	0.0	1.0	0.0	0.1882	0.0	40.0	345.0

Continua nella pagina seguente

Tabella C.5 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
31	821	0.1954	0.0	1.0	0.0	0.3269	0.0	17.0	70.0
32	861	0.0368	0.0	1.0	0.0	0.0709	0.0	468.0	12264.0
33	921	0.2609	0.0	1.0	0.0	0.4138	0.0	48.0	136.0
34	923	0.2088	0.0	1.0	0.0	0.3455	0.0	19.0	72.0
35	925	0.2	0.0	1.0	0.0	0.3333	0.0	12.0	48.0
36	941	0.0444	0.0	1.0	0.0	0.085	0.0	248.0	5341.0
37	943	0.2308	0.0	1.0	0.0	0.375	0.0	21.0	70.0
38	945	0.0328	0.0	1.0	0.0	0.0634	0.0	267.0	7884.0
39	947	0.0578	0.0	1.0	0.0	0.1093	0.0	73.0	1190.0

Tabella C.6: *Report* di classificazione di ciascun *topic* per il classificatore basato sulle *feature* introdotte da Stab e Gurevych[148].

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
		link	no-link	link	no-link	link	no-link	link	no-link
1	1	0.0534	0.9934	0.9956	0.0361	0.1014	0.0696	453.0	8294.0
2	21	0.0366	0.9786	0.9459	0.0905	0.0704	0.1656	111.0	3040.0
3	61	0.0614	0.9811	0.975	0.0802	0.1156	0.1484	40.0	648.0
4	81	0.0568	0.9679	0.898	0.1712	0.1068	0.2909	49.0	882.0
5	101	0.0315	0.9765	0.9441	0.0742	0.0609	0.1379	143.0	4488.0
6	121	0.0855	0.9692	0.9487	0.1373	0.1568	0.2405	39.0	459.0
7	181	0.0463	0.975	0.993	0.0132	0.0885	0.0261	142.0	2944.0
8	221	0.0423	0.971	0.9701	0.0435	0.081	0.0833	67.0	1540.0
9	323	0.0391	0.9783	0.9623	0.067	0.0751	0.1253	53.0	1344.0
10	381	0.0508	0.9576	0.8936	0.126	0.0962	0.2227	47.0	897.0
11	641	0.102	0.0	1.0	0.0	0.1852	0.0	15.0	132.0
12	642	0.0634	0.9653	0.7561	0.3777	0.117	0.543	82.0	1472.0
13	643	0.2	0.0	1.0	0.0	0.3333	0.0	5.0	20.0
14	644	0.1495	0.0	1.0	0.0	0.2602	0.0	16.0	91.0
15	645	0.0505	1.0	1.0	0.007	0.0961	0.0139	68.0	1288.0
16	646	0.1793	0.7143	0.982	0.0099	0.3032	0.0196	111.0	504.0
17	647	0.0955	0.0	1.0	0.0	0.1743	0.0	76.0	720.0

Continua nella pagina seguente

Tabella C.6 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
18	648	0.0249	0.989	0.9824	0.0393	0.0485	0.0757	227.0	9100.0
19	662	0.2593	0.0	1.0	0.0	0.4118	0.0	7.0	20.0
20	663	0.0841	0.9231	0.9643	0.0392	0.1547	0.0752	28.0	306.0
21	664	0.0404	0.9919	0.9821	0.0853	0.0775	0.1571	112.0	2860.0
22	665	0.3333	1.0	1.0	0.1111	0.5	0.2	4.0	9.0
23	681	0.0508	0.9808	0.9804	0.0517	0.0965	0.0983	51.0	986.0
24	683	0.0602	0.9733	0.9744	0.0579	0.1133	0.1094	78.0	1260.0
25	701	0.083	0.9688	0.9524	0.123	0.1527	0.2183	21.0	252.0
26	721	0.1176	0.0	1.0	0.0	0.2105	0.0	20.0	150.0
27	742	0.0505	0.9888	0.9928	0.033	0.0961	0.0639	138.0	2665.0
28	743	0.1667	1.0	1.0	0.1667	0.2857	0.2857	6.0	36.0
29	744	0.094	1.0	1.0	0.0041	0.1718	0.0082	25.0	242.0
30	761	0.1039	0.0	1.0	0.0	0.1882	0.0	40.0	345.0
31	821	0.1954	0.0	1.0	0.0	0.3269	0.0	17.0	70.0
32	861	0.0375	0.9822	0.9829	0.036	0.0722	0.0695	468.0	12264.0
33	921	0.2609	0.0	1.0	0.0	0.4138	0.0	48.0	136.0
34	923	0.2754	1.0	1.0	0.3056	0.4318	0.4681	19.0	72.0
35	925	0.2	0.0	1.0	0.0	0.3333	0.0	12.0	48.0
36	941	0.0452	0.9645	0.9315	0.0865	0.0862	0.1588	248.0	5341.0
37	943	0.2329	0.7778	0.8095	0.2	0.3617	0.3182	21.0	70.0
38	945	0.0339	0.9778	0.9326	0.1003	0.0654	0.182	267.0	7884.0
39	947	0.0595	0.9562	0.9178	0.1101	0.1118	0.1974	73.0	1190.0

Tabella C.7: *Report* di classificazione di ciascun *topic* per il classificatore neurale.

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
		link	no-link	link	no-link	link	no-link	link	no-link
1	1	0.0538	0.9577	0.8565	0.1776	0.1013	0.2996	453.0	8294.0
2	21	0.0348	0.9621	0.8468	0.1421	0.0668	0.2476	111.0	3040.0
3	61	0.0567	0.9318	0.85	0.1265	0.1062	0.2228	40.0	648.0
4	81	0.0531	0.9485	0.7143	0.2925	0.0989	0.4471	49.0	882.0
5	101	0.0314	0.9767	0.951	0.0653	0.0608	0.1224	143.0	4488.0

Continua nella pagina seguente

Tabella C.7 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
6	121	0.0816	0.929	0.7179	0.3137	0.1466	0.4691	39.0	459.0
7	181	0.046	0.9444	0.993	0.0058	0.0879	0.0115	142.0	2944.0
8	221	0.0426	0.9613	0.791	0.226	0.0808	0.3659	67.0	1540.0
9	323	0.0379	0.9617	0.8679	0.131	0.0726	0.2305	53.0	1344.0
10	381	0.0474	0.9387	0.7872	0.1706	0.0894	0.2887	47.0	897.0
11	641	0.1042	1.0	1.0	0.0227	0.1887	0.0444	15.0	132.0
12	642	0.0544	0.9524	0.7805	0.2446	0.1017	0.3892	82.0	1472.0
13	643	0.2	0.0	1.0	0.0	0.3333	0.0	5.0	20.0
14	644	0.1471	0.8	0.9375	0.044	0.2542	0.0833	16.0	91.0
15	645	0.049	0.9296	0.9265	0.0512	0.0931	0.0971	68.0	1288.0
16	646	0.1916	0.8684	0.8649	0.1964	0.3137	0.3204	111.0	504.0
17	647	0.0969	1.0	1.0	0.0167	0.1767	0.0328	76.0	720.0
18	648	0.0244	0.9758	0.8855	0.1154	0.0474	0.2064	227.0	9100.0
19	662	0.2593	0.0	1.0	0.0	0.4118	0.0	7.0	20.0
20	663	0.0861	0.9375	0.9286	0.098	0.1576	0.1775	28.0	306.0
21	664	0.0398	0.9755	0.9107	0.1392	0.0762	0.2436	112.0	2860.0
22	665	0.4	0.75	0.5	0.6667	0.4444	0.7059	4.0	9.0
23	681	0.0507	0.9714	0.9608	0.069	0.0963	0.1288	51.0	986.0
24	683	0.061	0.9783	0.9744	0.0714	0.1148	0.1331	78.0	1260.0
25	701	0.0785	0.9355	0.9048	0.1151	0.1445	0.2049	21.0	252.0
26	721	0.119	1.0	1.0	0.0133	0.2128	0.0263	20.0	150.0
27	742	0.0534	0.973	0.913	0.1621	0.1009	0.2779	138.0	2665.0
28	743	0.1429	0.8571	0.8333	0.1667	0.2439	0.2791	6.0	36.0
29	744	0.0965	1.0	1.0	0.0331	0.1761	0.064	25.0	242.0
30	761	0.1044	1.0	1.0	0.0058	0.1891	0.0115	40.0	345.0
31	821	0.1954	0.0	1.0	0.0	0.3269	0.0	17.0	70.0
32	861	0.0359	0.9558	0.8697	0.1076	0.0689	0.1933	468.0	12264.0
33	921	0.2532	0.7	0.8125	0.1544	0.3861	0.253	48.0	136.0
34	923	0.2182	0.8056	0.6316	0.4028	0.3243	0.537	19.0	72.0
35	925	0.2069	1.0	1.0	0.0417	0.3429	0.08	12.0	48.0
36	941	0.0424	0.9396	0.8508	0.1078	0.0808	0.1935	248.0	5341.0
37	943	0.2462	0.8077	0.7619	0.3	0.3721	0.4375	21.0	70.0
38	945	0.0338	0.9715	0.8315	0.1943	0.0649	0.3239	267.0	7884.0

Continua nella pagina seguente

Tabella C.7 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
39	947	0.058	0.9444	0.9178	0.0857	0.1091	0.1572	73.0	1190.0

Tabella C.8: *Report* di classificazione di ciascun *topic* per il classificatore basato sulle principali *feature* della *stance classification* selezionate all'interno del capitolo 3.

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
		link	no-link	link	no-link	link	no-link	link	no-link
1	1	0.0558	0.9834	0.9669	0.1071	0.1056	0.1931	453.0	8294.0
2	21	0.0382	0.9825	0.9279	0.148	0.0735	0.2573	111.0	3040.0
3	61	0.0628	0.9697	0.925	0.1481	0.1176	0.257	40.0	648.0
4	81	0.0613	0.9739	0.8776	0.254	0.1147	0.4029	49.0	882.0
5	101	0.0328	0.9793	0.8951	0.158	0.0632	0.2721	143.0	4488.0
6	121	0.0818	0.9346	0.8205	0.2179	0.1488	0.3534	39.0	459.0
7	181	0.0472	1.0	1.0	0.0255	0.0901	0.0497	142.0	2944.0
8	221	0.0425	0.9639	0.8955	0.1214	0.0811	0.2157	67.0	1540.0
9	323	0.0399	0.9765	0.9245	0.1235	0.0766	0.2193	53.0	1344.0
10	381	0.0508	0.9655	0.9574	0.0624	0.0965	0.1173	47.0	897.0
11	641	0.1034	1.0	1.0	0.0152	0.1875	0.0299	15.0	132.0
12	642	0.0846	0.9644	0.561	0.6617	0.147	0.7849	82.0	1472.0
13	643	0.1818	0.6667	0.8	0.1	0.2963	0.1739	5.0	20.0
14	644	0.15	0.8571	0.9375	0.0659	0.2586	0.1224	16.0	91.0
15	645	0.0518	1.0	1.0	0.0326	0.0984	0.0632	68.0	1288.0
16	646	0.1849	0.9032	0.973	0.0556	0.3108	0.1047	111.0	504.0
17	647	0.0956	1.0	1.0	0.0014	0.1745	0.0028	76.0	720.0
18	648	0.0264	0.9937	0.9736	0.1045	0.0514	0.1891	227.0	9100.0
19	662	0.2593	0.0	1.0	0.0	0.4118	0.0	7.0	20.0
20	663	0.0933	1.0	1.0	0.1111	0.1707	0.2	28.0	306.0
21	664	0.0398	0.9946	0.9911	0.0643	0.0766	0.1209	112.0	2860.0
22	665	0.3333	1.0	1.0	0.1111	0.5	0.2	4.0	9.0
23	681	0.0503	0.9677	0.9608	0.0609	0.0955	0.1145	51.0	986.0
24	683	0.0611	0.9872	0.9872	0.0611	0.1151	0.1151	78.0	1260.0
25	701	0.0909	0.9811	0.9524	0.2063	0.166	0.341	21.0	252.0

Continua nella pagina seguente

Tabella C.8 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Precision		Recall		F1-score		Support	
26	721	0.1154	0.8571	0.9	0.08	0.2045	0.1463	20.0	150.0
27	742	0.0526	0.9874	0.9783	0.0882	0.0999	0.1619	138.0	2665.0
28	743	0.1765	1.0	1.0	0.2222	0.3	0.3636	6.0	36.0
29	744	0.0958	1.0	1.0	0.0248	0.1748	0.0484	25.0	242.0
30	761	0.1044	0.9048	0.95	0.0551	0.1881	0.1038	40.0	345.0
31	821	0.1977	1.0	1.0	0.0143	0.3301	0.0282	17.0	70.0
32	861	0.0378	0.9753	0.9487	0.0773	0.0726	0.1432	468.0	12264.0
33	921	0.2609	0.0	1.0	0.0	0.4138	0.0	48.0	136.0
34	923	0.2375	1.0	1.0	0.1528	0.3838	0.2651	19.0	72.0
35	925	0.2	0.0	1.0	0.0	0.3333	0.0	12.0	48.0
36	941	0.0438	0.9524	0.8468	0.1425	0.0834	0.2479	248.0	5341.0
37	943	0.3137	0.875	0.7619	0.5	0.4444	0.6364	21.0	70.0
38	945	0.0377	0.9839	0.8876	0.2324	0.0723	0.3759	267.0	7884.0
39	947	0.0635	0.9812	0.9589	0.1319	0.119	0.2326	73.0	1190.0

Tabella C.9: Matrici di confusione per ciascun classificatore proposto rispetto ad ogni *topic* appartenente al set di *train and test*.

Topic graph id	Topic id	Baseline CS Predicted		Baseline n-gram Predicted		Stab and Gurevych Predicted		RNN Predicted		SC Predicted			
		link	no-link	link	no-link	link	no-link	link	no-link	link	no-link		
1	1	453	0	449	4	451	2	388	65	438	15	link	True
		8294	0	8189	105	7995	299	6821	1473	7406	888	no-link	
2	21	111	0	105	6	105	6	94	17	103	8	link	True
		3040	0	2781	259	2765	275	2608	432	2590	450	no-link	
3	61	40	0	39	1	39	1	34	6	37	3	link	True
		648	0	627	21	596	52	566	82	552	96	no-link	
4	81	49	0	47	2	44	5	35	14	43	6	link	True
		882	0	829	53	731	151	624	258	658	224	no-link	
5	101	143	0	133	10	135	8	136	7	128	15	link	True

Continua nella pagina seguente

Tabella C.9 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Baseline CS Predicted		Baseline n-gram Predicted		Stab and Gurevych Predicted		RNN Predicted		SC Predicted			
		4488	0	4157	331	4155	333	4195	293	3779	709	no-link	
6	121	39	0	36	3	37	2	28	11	32	7	link	True
		459	0	419	40	396	63	315	144	359	100	no-link	
7	181	142	0	141	1	141	1	141	1	142	0	link	True
		2944	0	2924	20	2905	39	2927	17	2869	75	no-link	
8	221	67	0	63	4	65	2	53	14	60	7	link	True
		1540	0	1452	88	1473	67	1192	348	1353	187	no-link	
9	323	53	0	52	1	51	2	46	7	49	4	link	True
		1344	0	1306	38	1254	90	1168	176	1178	166	no-link	
10	381	47	0	44	3	42	5	37	10	45	2	link	True
		897	0	982	39	784	113	744	153	841	56	no-link	
11	641	15	0	15	0	15	0	15	0	15	0	link	True
		132	0	132	0	132	0	129	3	130	2	no-link	
12	642	82	0	51	31	62	20	64	18	46	36	link	True
		1472	0	933	539	916	556	1112	360	498	974	no-link	
13	643	5	0	5	0	5	0	5	0	4	1	link	True
		20	0	20	0	20	0	20	0	18	2	no-link	
14	644	16	0	16	0	16	0	15	1	15	1	link	True
		91	0	91	0	91	0	87	4	85	6	no-link	
15	645	68	0	68	0	68	0	63	5	68	0	link	True
		1288	0	1285	3	1279	9	1222	66	1246	42	no-link	
16	646	111	0	109	2	109	2	96	15	108	3	link	True
		504	0	495	9	499	5	405	99	476	28	no-link	
17	647	76	0	76	0	76	0	76	0	76	0	link	True
		720	0	720	0	720	0	708	12	719	1	no-link	
18	648	227	0	218	9	223	4	201	26	221	6	link	True
		9100	0	8916	184	8742	358	8050	1050	8149	951	no-link	
19	662	7	0	7	0	7	0	7	0	7	0	link	True
		20	0	20	0	20	0	20	0	20	0	no-link	
20	663	28	0	28	0	27	1	26	2	28	0	link	True
		306	0	303	3	294	12	276	30	272	34	no-link	

Continua nella pagina seguente

Tabella C.9 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Baseline CS Predicted		Baseline n-gram Predicted		Stab and Gurevych Predicted		RNN Predicted		SC Predicted			
21	664	112	0	110	2	110	2	102	10	111	1	link	True
		2860	0	2797	63	2616	244	2462	398	2676	184	no-link	
22	665	4	0	4	0	4	0	2	2	4	0	link	True
		9	0	9	0	8	1	3	6	8	1	no-link	
23	681	51	0	51	0	50	935	49	2	49	2	link	True
		986	0	982	4	1	51	918	68	926	60	no-link	
24	683	78	0	77	1	76	2	76	2	77	1	link	True
		1260	0	1211	49	1187	73	1170	90	1183	77	no-link	
25	701	21	0	20	1	20	1	19	2	20	1	link	True
		252	0	244	8	221	31	223	29	200	52	no-link	
26	721	20	0	20	0	20	0	20	0	18	2	link	True
		150	0	150	0	150	0	148	2	138	12	no-link	
27	742	138	0	135	3	137	1	126	12	135	3	link	True
		2665	0	2609	56	2577	88	2233	432	2430	235	no-link	
28	743	6	0	6	0	6	0	5	1	6	0	link	True
		36	0	36	0	30	6	30	6	28	8	no-link	
29	744	25	0	25	0	25	0	25	0	25	0	link	True
		242	0	241	1	241	1	234	8	236	6	no-link	
30	761	40	0	38	2	40	0	40	0	38	2	link	True
		345	0	332	13	345	0	343	2	326	19	no-link	
31	821	17	0	17	0	17	0	17	0	17	0	link	True
		70	0	70	0	70	0	70	0	69	1	no-link	
32	861	468	0	432	35	460	8	407	61	444	24	link	True
		12264	0	11529	735	11822	442	10945	1319	11316	984	no-link	
33	921	48	0	48	0	48	0	39	9	48	0	link	True
		136	0	136	0	136	0	115	21	136	0	no-link	
34	923	19	0	19	0	19	0	12	7	19	0	link	True
		72	0	72	0	50	22	43	29	61	11	no-link	
35	925	12	0	12	0	12	0	12	0	12	0	link	True
		48	0	48	0	48	0	46	2	48	0	no-link	
36	941	248	0	202	46	231	17	211	37	210	38	link	True

Continua nella pagina seguente

Tabella C.9 – *Continuazione della pagina precedente.*

Topic graph id	Topic id	Baseline CS Predicted		Baseline n-gram Predicted		Stab and Gurevych Predicted		RNN Predicted		SC Predicted			
		5341	0	5027	314	4879	462	4765	576	4580	761	no-link	
37	943	21	0	16	5	17	4	16	5	16	5	link	True
		70	0	55	15	56	14	49	21	35	35	no-link	
38	945	267	0	250	17	249	18	222	45	237	30	link	True
		7884	0	7094	790	7093	791	6352	1532	6052	1832	no-link	
39	947	73	0	71	2	67	6	67	6	70	3	link	True
		1190	0	1126	64	1059	131	1088	102	1033	157	no-link	

Bibliografia

[1] Amjad Abu-Jbara, Pradeep Dasigi, Mona T. Diab, and Dragomir R. Radev. Subgroup detection in ideological discussions. In *ACL*, 2012.

[2] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. The evaluation of sentence similarity measures. In *DaWaK*, 2008.

[3] Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. Mining newsgroups using networks arising from social behavior. In *WWW*, 2003.

[4] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *ArgMining@ACL*, 2014.

[5] Pranav Anand, Marilyn A. Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael S. Minor. Cats rule and dogs drool!: Classifying stance in online debate. 2011.

[6] Kevin D. Ashley and Vern R. Walker. Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In *ICAIL*, 2013.

[7] Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. Stance detection with bidirectional conditional encoding. In *EMNLP*, 2016.

[8] Eriq Augustine and Dhawal Joharapurkar. Cmps 245, winter 17 project: Ideology-backed stance classification. 2017.

[9] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, 2010.

[10] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[11] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *COLING-ACL*, 1998.

[12] Alexandra Balahur, Zornitsa Kozareva, and Andrés Montoyo. Determining the polarity and source of opinions expressed in political debates. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '09, pages 468–480, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Mohit Bansal, Claire Cardie, and Lillian Lee. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *COLING*, 2008.

[14] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. Stance classification of context-dependent claims. In *EACL*, 2017.

[15] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. Automatic evaluation of design alternatives with quantitative argumentation. *Argument and Computation*, 6:24–49, 2015.

[16] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171:619–641, 2007.

[17] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence Review*, 33:211–259, 2010.

[18] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introduction to structured argumentation. *Argument and Computation*, 5:1–4, 2014.

[19] Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. Implementing the argument web. *Commun. ACM*, 56:66–73, 2013.

[20] Or Biran and Owen Rambow. Identifying justifications in written dialogs by classifying text as argumentative. *Int. J. Semantic Computing*, 5:363–381, 2011.

[21] Filip Boltuzic and Jan Snajder. Back up your stance: Recognizing arguments in online discussions. In *ArgMining@ACL*, 2014.

[22] Katarzyna Budzynska, Mathilde Janier, Juyeon Kang, Chris Reed, Patrick Saint-Dizier, Manfred Stede, and Olena Yaskorska. Towards argument mining from dialogue. In *COMMA*, 2014.

[23] Clinton Burfoot, Steven B Bird, and Timothy Baldwin. Collective classification of congressional floor-debate transcripts. In *ACL*, 2011.

[24] Elena Cabrio and Serena Villata. Combining textual entailment and argumentation theory for supporting online debates interactions. In *ACL*, 2012.

[25] Elena Cabrio and Serena Villata. Natural language arguments: A combined approach. In *ECAI*, 2012.

[26] Elena Cabrio and Serena Villata. A natural language bipolar argumentation approach to support users in online debate interactions. *Argument & Computation*, 4:209–230, 2013.

[27] Elena Cabrio and Serena Villata. Node: A benchmark of natural language arguments. In *COMMA*, 2014.

[28] Amparo Elizabeth Cano, Yulan He, Kang Liu, and Jun Zhao. A weakly supervised bayesian model for violence detection in social media. In *IJCNLP*, 2013.

[29] L. Carlson, M.E. Okurowski, D. Marcu, and Linguistic Data Consortium. *RST Discourse Treebank*. Linguistic Data Consortium. Linguistic Data Consortium, University of Pennsylvania, 2002.

[30] Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP*, pages 109–117, 2001.

[31] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL*, 2002.

[32] Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *HLT-NAACL*, 2006.

[33] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. Semafor 1.0: A probabilistic frame-semantic parser. 2010.

[34] Jia Deng, Jonathan Krause, and Fei fei Li. Fine-grained crowdsourcing for fine-grained recognition. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2013.

[35] Lipika Dey and S. K. Mirajul Haque. Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12:205–226, 2008.

[36] Sarah Djemili, Julien Longhi, Claudia Marinica, Dimitris Kotzinos, and Georges-Elia Sarfati. What does Twitter have to say about ideology? In Gertrud Faaß & Josef Ruppenhofer, editor, *NLP 4 CMC: Natural Language Processing for Computer-Mediated Communication / Social Media - Pre-conference workshop at Konvens 2014*, volume 1, pages http://www.uni–hildesheim.de/konvens2014/data/konvens2014–workshop–proceedings.pdf: p.16–25, Hildesheim, Germany, October 2014. Universitätsverlag Hildesheim.

[37] Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. Stance classification with target-specific neural attention networks. 2017.

[38] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77:321–358, 1995.

[39] David A. Easley and Jon M. Kleinberg. Networks, crowds, and markets: Reasoning about a highly connected world (easley, d. and kleinberg, j.; 2010) [book review]. *IEEE Technology and Society Magazine*, 32:10–30, 2010.

[40] Javid Ebrahimi, Dejing Dou, and Daniel Lowd. A joint sentiment-target-stance model for stance classification in tweets. In *COLING*, 2016.

[41] Javid Ebrahimi, Dejing Dou, and Daniel Lowd. Weakly supervised tweet stance classification by relational bootstrapping. In *EMNLP*, 2016.

[42] Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *EMNLP*, 2015.

[43] Andrea Esuli and Fabrizio Sebastiani. Determining term subjectivity and term orientation for opinion mining. In *EACL*, 2006.

[44] James Z Fan, Aditya Kalyanpur, David Gondek, and David A. Ferrucci. Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56:5, 2012.

[45] Adam Faulkner. Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure. In *FLAIRS Conference*, 2014.

[46] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *HLT-NAACL*, 2016.

[47] J.B. Freeman. *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. Pragmatics and Discourse Analysis Series. Foris Publications, 1991.

[48] Simone Gabbriellini and Paolo Torroni. A new framework for abms based on argumentative reasoning. In *ESSA*, 2013.

[49] Simone Gabbriellini and Paolo Torroni. Microdebates: Structuring debates without a structuring tool. *AI Commun.*, 29:31–51, 2016.

[50] Jean Mark Gawron, Dipak K Gupta, Kellen Stephens, Ming-Hsiang Tsou, Brian H. Spitzberg, and Li An. Using group membership markers for group identification in web logs. 2012.

[51] Lise Getoor. Tutorial on statistical relational learning. In *Proceedings of the 15th International Conference on Inductive Logic Programming*, ILP'05, pages 415–415, Berlin, Heidelberg, 2005. Springer-Verlag.

[52] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 7:3–12, 2005.

[53] Theodosis Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news, blogs, and the social web. *International Journal on Artificial Intelligence Tools*, 24(05):1540024, 2015.

[54] Nancy Green. Towards creation of a corpus for argumentation mining the biomedical genetics research literature. In *ArgMining@ACL*, 2014.

[55] Lin Gui, Ruifeng Xu, Yulan He, Qin Lu, and Zhongyu Wei. Intersubjectivity and sentiment: From language to knowledge. In *IJCAI*, 2016.

[56] Adrien Guille. Information diffusion in online social networks. In *SIGMOD/PODS Ph.D. Symposium*, 2013.

[57] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining on the web from information seeking perspective. In *ArgNLP*, 2014.

[58] Yaakov HaCohen-Kerner, Ziv Ido, and Ronen Ya'akobov. Stance classification of tweets using skip char ngrams. 2017.

[59] Kazi Saidul Hasan and Vincent Ng. Extra-linguistic constraints on stance recognition in ideological debates. In *ACL*, 2013.

[60] Kazi Saidul Hasan and Vincent Ng. Stance classification of ideological debates: Data, models, features, and constraints. In *IJCNLP*, 2013.

[61] Kazi Saidul Hasan and Vincent Ng. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *EMNLP*, 2014.

[62] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 8:1735–80, 1997.

[63] Alexander Hogenboom, Frederik Hogenboom, Uzay Kaymak, Paul Wouters, and Franciska de Jong. Mining economic sentiment using argumentation structures. In *ER Workshops*, 2010.

[64] Hospice Houngbo and Robert E. Mercer. An automated method to build a corpus of rhetorically-classified sentences in biomedical texts. In *ArgMining@ACL*, 2014.

[65] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[66] Clayton J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*, 2014.

[67] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Practical extraction of disaster-relevant information from social media. In *WWW*, 2013.

[68] Diana Inkpen, Xiao-Dan Zhu, and Parinaz Sobhani. A dataset for multi-target stance detection. In *EACL*, 2017.

[69] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *ACL*, 2011.

[70] David Jurgens. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *ICWSM*, 2013.

[71] Noriaki Kawamae. Predicting future reviews: sentiment analysis models for collaborative filtering. In *WSDM*, 2011.

[72] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[73] Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *ArgMining@HLT-NAACL*, 2015.

[74] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[75] Werner Kunz, Horst W. J. Rittel, We Messrs, H. Dehlinger, T. Mann, and J. J. Protzen. Issues as elements of information systems. Technical report, 1970.

[76] Namhee Kwon, Liang Zhou, Eduard H. Hovy, and Stuart W. Shulman. Identifying and classifying subjective claims. In *DG.O*, 2007.

[77] John Lawrence, Chris Reed, Colin Allen, Simon McAlister, and Andrew Ravenscroft. Mining arguments from 19th century philosophical texts using topic based modelling. In *ArgMining@ACL*, 2014.

[78] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[79] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In *COLING*, 2014.

[80] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing implicit discourse relations in the penn discourse treebank. In *EMNLP*, 2009.

[81] Eric Lindahl, Stephen O'Hara, and Qiuming Zhu. A multi-agent system of evidential reasoning for intelligence analyses. In *AAMAS*, 2007.

[82] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In *IJCAI*, 2015.

[83] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Techn.*, 16:10:1–10:25, 2016.

[84] Marco Lippi and Paolo Torroni. Margot: A web server for argumentation mining. *Expert Syst. Appl.*, 65:292–303, 2016.

[85] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, 2012.

[86] Michael Maes and Andreas Flache. Differentiation without distancing. explaining bi-polarization of opinions without negative influence. In *PloS one*, 2013.

[87] Christopher D. Manning and Hinrich Schütze. Foundations of statistical natural language processing. *Information Retrieval*, 4:80–81, 2001.

[88] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, 2014.

[89] Hugo Mercier and Dan Sperber. Why do humans reason? arguments for an argumentative theory. *The Behavioral and brain sciences*, 34 2:57–74; discussion 74–111, 2011.

[90] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[91] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[92] Michela Milano, Barry O'Sullivan, and Marco Gavanelli. Sustainable policy making: A strategic challenge for artificial intelligence. *AI Magazine*, 35:22–35, 2014.

[93] Amita Misra, Brian Ecker, Theodore Handleman, Nicolas Hahn, and Marilyn A. Walker. Nlds-ucsc at semeval-2016 task 6: A semi-supervised approach to detecting stance in tweets. In *SemEval@NAACL-HLT*, 2016.

[94] Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos Ivan Chesnevar, Wolfgang Dvorak, Marcelo A. Falappa, Xiuyi Fan, Sarah Alice Gaggl, Alejandro J. Garcia, Maria P. Gonzalez, Thomas F. Gordon, Joao Leite, Martin Mozina, Chris Reed, Guillermo Ricardo Simari, Stefan Szeider, Paolo Torroni, and Stefan Woltran. The added value of argumentation. 2012.

[95] Marie-Francine Moens. Argumentation mining: Where are we now, where do we want to be and how do we get there? In *FIRE*, 2013.

[96] Saif Mohammad. Emotional tweets. In **SEM@NAACL-HLT*, 2012.

[97] Saif Mohammad. A practical guide to sentiment annotation: Challenges and solutions. In *WASSA@NAACL-HLT*, 2016.

[98] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval@NAACL-HLT*, 2016.

[99] Saif Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *ACM Trans. Internet Techn.*, 17:26:1–26:23, 2017.

[100] Saif Mohammad and Peter D. Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. 2010.

[101] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

[102] Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*, 2006.

[103] Alessandro Moschitti. Making tree kernels practical for natural language learning. In *EACL*, 2006.

[104] Alessandro Moschitti. State-of-the-art kernels for natural language processing. In *ACL*, 2012.

[105] James R. Munkres. Algorithms for the assignment and transportation problems. 1957.

[106] Akiko Murakami and Raymond H. Putra. Support or oppose? classifying positions in online debates from reply activities and opinion expressions. In *COLING*, 2010.

[107] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. 2007.

[108] Susan E. Newman and Catherine C. Marshall. Pushing toulmin too far: Learning from an argument representation scheme. 1992.

[109] Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *ICML*, 2007.

[110] Stefanie Nowak and Stefan M. Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Multimedia Information Retrieval*, 2010.

[111] Sebastian Padó, Tae-Gil Noh, Asher Stern, Rui Wang, and Roberto Zanoli. Design and realization of a modular architecture for textual entailment. *Natural Language Engineering*, 21:167–200, 2015.

[112] Raquel Mochales Palau and Aagje Ieven. Creating an argumentation corpus: do theories apply to real arguments?: a case study on the legal argumentation of the echr. In *ICAIL*, 2009.

[113] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19:1–22, 2010.

[114] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2007.

[115] Joonsuk Park and Claire Cardie. Identifying appropriate support for propositions in online user comments. In *ArgMining@ACL*, 2014.

[116] Joonsuk Park, Arzoo Katiyar, and Bishan Yang. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *ArgMining@HLT-NAACL*, 2015.

[117] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *ACL*, 2015.

[118] Andreas Peldszus. Towards segment-based recognition of argumentation structure in short texts. In *ArgMining@ACL*, 2014.

[119] Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts: A survey. *IJCINI*, 7:1–31, 2013.

[120] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[121] John L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[122] Hoifung Poon and Pedro M. Domingos. Joint inference in information extraction. In *AAAI*, 2007.

[123] Hoifung Poon and Pedro M. Domingos. Unsupervised semantic parsing. In *EMNLP*, 2009.

[124] Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind K. Joshi, and Livio Robaldo. The penn discourse treebank 2.0 annotation manual. 2007.

[125] Minghui Qiu, Liu Yang, and Jing Jiang. Modeling interaction features for debate side clustering. In *CIKM*, 2013.

[126] J R Martin and Peter White. *The Language of Evaluation: Appraisal in English*. 01 2005.

[127] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Twitter user geolocation using a unified text and network prediction model. In *ACL*, 2015.

[128] Ashwin Rajadesingan and Huan Liu. Identifying users with opposing opinions in twitter debates. In *SBP*, 2014.

[129] Pavithra Rajendran, Danushka Bollegala, and Simon Parsons. Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews. In *ArgMining@ACL*, 2016.

[130] Paul Reisert, Junta Mizuno, Miwa Kanno, Naoaki Okazaki, and Kentaro Inui. A corpus study for identifying evidence on microblogs. In *LAW@COLING*, 2014.

[131] Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence - an automatic method for context dependent evidence detection. In *EMNLP*, 2015.

[132] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[133] Patrick Saint-Dizier. Processing natural language arguments with theplatform. *Argument & Computation*, 3:49–82, 2012.

[134] Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news. In *ArgMining@HLT-NAACL*, 2015.

[135] John Scott. *Social network analysis*. Sage, 2017.

[136] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, 2002.

[137] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander F. Gelbukh, and Liliana Chanona-Hernández. Syntactic n-grams as machine learning features for natural language processing. *Expert Syst. Appl.*, 41:853–860, 2014.

[138] Vasiliki Simaki, Carita Paradis, and Andreas Kerren. *Stance Classification in Texts from Blogs on the 2016 British Referendum*, pages 700–709. Springer International Publishing, Cham, 2017.

[139] Guillermo Ricardo Simari and Ronald Prescott Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artif. Intell.*, 53:125–157, 1992.

[140] Parinaz Sobhani, Diana Inkpen, and Stan Matwin. From argumentation mining to stance classification. In *ArgMining@HLT-NAACL*, 2015.

[141] Richard Socher, A. V. Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[142] Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *EMNLP*, 2009.

[143] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in online debates. In *ACL/IJCNLP*, 2009.

[144] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological on-line debates. 2010.

[145] Dhanya Sridhar, James R. Foulds, Bert Huang, Lise Getoor, and Marilyn A. Walker. Joint models of disagreement and stance in online debate. In *ACL*, 2015.

[146] Dhanya Sridhar, Lise Getoor, and Marilyn A. Walker. Collective stance classification of posts in online debate forums. 2014.

[147] Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *COLING*, 2014.

[148] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *EMNLP*, 2014.

[149] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[150] Simone Teufel, Vasilis Karaiskos, Anne Wilson, and David McKelvie. Argumentative zoning: Information extraction from scientific text. 1999.

[151] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *EMNLP*, 2006.

[152] Geoff Thompson. Evaluation in text: Authorial stance and the construction of discourses. 2013.

[153] Orith Toledo-Ronen, Roy Bar-Haim, and Noam Slonim. Expert stance graphs for computational argumentation. In *ArgMining@ACL*, 2016.

[154] S.E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2003.

[155] Mathias Verbeke, Paolo Frasconi, Vincent Van Asch, Roser Morante, Walter Daelemans, and Luc De Raedt. Kernel-based logical and relational learning with klog for hedge cue detection. In *ILP*, 2011.

[156] Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. In *SemEval@NAACL-HLT*, 2016.

[157] S. V. N. Vishwanathan and Alexander J. Smola. Fast kernels for string and tree matching. In *NIPS*, 2002.

[158] Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. A review corpus for argumentation analysis. In *CICLing*, 2014.

[159] Marilyn A. Walker, Pranav Anand, Rob Abbott, and Ricky Grant. Stance classification using dialogic properties of persuasion. In *HLT-NAACL*, 2012.

[160] Marilyn A. Walker, Pranav Anand, Rob Abbott, Jean E. Fox Tree, Craig H. Martell, and Joseph King. That is your evidence?: Classifying stance in online political debate. *Decision Support Systems*, 53:719–729, 2012.

[161] Marilyn A. Walker, Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *LREC*, 2012.

[162] Douglas Walton. Argumentation theory: A very short introduction. In *Argumentation in Artificial Intelligence*, 2009.

[163] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, 2016.

[164] Yi-Chia Wang and Carolyn Penstein Rosé. Making conversational structure explicit: Identification of initiation-response pairs within online discussions. In *HLT-NAACL*, 2010.

[165] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*, 2005.

[166] Cane Wing-ki. Integrating collaborative filtering and sentiment analysis: A rating inference approach. 2006.

[167] Ainur Yessenalina, Yisong Yue, and Claire Cardie. Multi-level structured models for document-level sentiment classification. In *EMNLP*, 2010.

[168] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *SIGIR*, 2003.

[169] Arkaitz Zubiaga, Bo Wang, Maria Liakata, and Rob Procter. Stance classification of social media users in independence movements. *CoRR*, abs/1702.08388, 2017.