POLITECNICO DI MILANO

Polo Territoriale di Como
Scuola di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica

LEXICON-BASED DOMAIN-AGNOSTIC MULTILINGUAL SENTIMENT ANALYTICS AS A SERVICE

Relatore: Prof. Emanuele DELLA VALLE

Correlatore: Ing. Christian MARAZZI

Tesi di Laurea di:

Marco TAGLIABUE

Matr. 823989

Anno Accademico 2015 / 2016

Ringraziamenti

Sommario

L'avvento del Web 2.0 ha portato numerosi cambiamenti nel modo di utilizzare Internet, dando la possibilità a chiunque, non solo di creare e reperire contenuti di qualunque tipo, ma anche di condividere opinioni e sensazioni influenzando così il pensiero altrui. Espressione particolare di questa evoluzione sono i social media, in cui i contenuti pubblicati dagli utenti si muovono all'interno di un flusso di dati in real-time. Grazie al Web 2.0 é ora possibile pensare applicazioni che fino a poco tempo fa non erano nemmeno immaginabili e che permettono, ad esempio, di monitorare la reputazione di un ristorante, analizzando le recensioni provenienti da TripAdvisor, Google+ e Yelp.

Un campo di studio nato in questo contesto é chiamato Sentiment Analysis, il cui scopo é quello di identificare in maniera automatica informazioni di tipo soggettivo provenienti da testi scritti o parlati. Un'approccio importante in grado di svolgere Sentiment Analysis in real-time e su grandi quantità di dati provenienti da domini di applicazioni differenti (turismo, app, prodotti di cucina, sport, notizie, ecc.) é chiamato Lexicon-based. Tale approccio utilizza un lessico composto da una lista di termini affiancati da un valore di polarità che rappresenta la connotazione positiva, negativa o neutrale di ciascun termine.

Le barriere principali per l'utilizzo sui Social Media di questo tipo di approccio sono legate alla necessità di avere un lessico per ogni lingua con cui si vuole lavorare e alla difficoltà di funzionamento all'interno di analisi di dati provenienti da domini di applicazione differenti come ad esempio le notizie, gli eventi sportivi o i programmi tv.

La prima parte di questo lavoro si concentra sul concetto di multilinguismo e sulla possibilità di trasportare le risorse lessicali come WordNet e SentiWordNet, prima presenti per la sola lingua inglese, sul più ampio numero di lingue possibile sfruttando le relazioni semantiche che intercorrono fra esse (in particolare utilizzando Global WordNet). La seconda parte si dedica alla creazione di un servizio Web di Sentiment Analysis, basato su un approccio Lexicon-based, in grado di sfruttare in maniera univoca i lessici multilingue precedentemente costruiti. Requisiti fondamentali per questa architettura sono la flessibilità di utilizzo, la possibilità di poter funzionare all'interno di analisi di dati real-time e la propensione ad un suo impiego su diversi domini di applicazione.

Segue una fase di valutazione comparativa mirata all'analisi delle prestazioni sul dominio delle app per Smart Phones (usando un dataset preso da Google Play) e sul dominio del turismo (usando un dataset preso da TripAdvisor) dimostrando di essere comparabile con strumenti monolingue, ma allo stesso tempo, muovendosi contemporaneamente su diciassette lingue differenti. Oggetto della comparazione sono i risultati ottenuti per le lingue

Inglese, Italiano, Francese, Portoghese e Greco dal progetto SentiStrenght, e quelli ottenuti dalla risorsa commerciale Dandelion per Italiano e Inglese. Inoltre è stata valutata la correttezza dello strumento nell'analisi di tweet in lingua Italiano, Inglese e Spagnolo relativi a un evento sportivo.

Viene infine presentato un caso reale di applicazione che dimostra il funzionamento del servizio Web di Sentiment Analysis su dati provenienti da Twitter in real-time. Il risultato ha permesso di monitorare l'andamento dell'opinione positiva o negativa delle persone riguardo ad eventi importanti, brand, programmi tv.

Abstract

The advent of the Web 2.0 brought several changes in the way we use Internet, giving the opportunity to anyone not only to create and trace every kind of contents, but also to share different opinions and feelings influencing in this way other people's thoughts. Social media are the peculiar expressions of this evolution. On social media, the contents published by the users move within a data flow in real-time. Thanks to the Web 2.0 there are now applications that until recently were not even conceivable and that allow, for example, to monitor a restaurant reputation analyzing the reviews from TripAdvisor, Google+ and Yelp.

Sentiment Analysis is a study field born in this context, its purpose is to automatically identify subjective information in written or spoken texts. Lexicon-based sentiment analytics is an important approach able to identify subjective information in real-time and on great data volume coming from different domains (travel, apps, kitchen products, sports, news, etc.). It uses a lexicon made up of a list of tuples each containing a term and a polarity value that carries the positive, negative or neutral connotation of each term.

The main barriers for this kind of approach in Social Media are due to the need of having a specific lexicon for every language with which we want to work and also to the operating complexity within the data analysis coming from domains of different application like news, sports or TV programs.

The first part of this master thesis is focused on the concept of multilingualism and on the opportunity to carry the lexical resources, like WordNet and SentiWordNet, of the English language, on a wider range of languages taking advantage of the semantic relationships that exist among themselves (in particular using Global WordNet). The second part is dedicated to the development of a Web service of Sentiment Analysis, built on a Lexicon-based approach able to utilize multilingual lexicons beforehand established with a univocal correspondence. Essential requirements for this kind of structure are the flexibility, the opportunity to work within the analysis of real-time data and the inclination of its use on different application domains.

Then, a comparative evaluation phase follows, which is focused on the analysis of the performances on mobile app domain (using a dataset taken from Google Play) and on travel domain (using a dataset taken from TripAdvisor), proving its comparability with the monolingual instruments and at the same time covering seventeen different languages. Subject of the comparison are SentiStrenght results obtained for English, Italian, French, Portuguese and Greek, and those obtained by the commercial resource Dandelion for Italian

and English. It was also evaluated accuracy of the tool on the analysis of tweets in Italian, English and Spanish related to a sporting event.

Furthermore, a real case of application shows the operation of the Web service of Sentiment Analysis on data derived from Twitter in real-time. The result has allowed to monitor the evolution of positive and negative opinions from people regarding important events, brand, TV shows.

Indice dei contenuti

Indice delle figure

Indice delle tabelle

1. INTRODUZIONE

L'ascesa del Web 2.0 ha generato numerosi cambiamenti relativi al mondo di Internet e al suo utilizzo, dando la possibilità di creare applicazioni che in precedenza non era possibile nemmeno immaginare.

In passato le informazioni pubblicate sul Web, generate da utenti con competenze tecniche che avevano l'obiettivo di rendere pubblici determinati dati, erano solo di tipo informativo e gli utenti comuni potevano unicamente consultarle durante la navigazione. Con l'avvento di nuove tecnologie si é riusciti a rendere l'utente capace di condividere una miriade di contenuti in forme diverse chiamati User Generated Contents (UCG). Si tratta di elementi di qualunque genere come video, immagini, musica ma anche opinioni e impressioni condivisibili su specifiche piattaforme come YouTube, Vimeo, Flickr, Instagram, TripAdvisor, Yelp, Foursquare ma anche su store di applicazioni come Google Play o Apple Store. In questo modo si rendono disponibili informazioni create da un gran numero di persone provenienti da tutto il mondo e che non sono all'interno del bacino di conoscenza personale dell'utente.

Il pensiero degli altri rispetto a una determinata tematica é sempre stato un'informazione importante per la maggior parte degli esseri umani che devono prendere una decisione in merito ad ambiti a loro non sufficientemente conosciuti. Molto prima dell'avvento del World Wide Web si cercavano opinioni e raccomandazioni riguardo a qualsiasi tipo di argomento: dal meccanico migliore in cui poter riparare l'auto, al voto durante l'elezioni o anche solo al consiglio se acquistare o meno un particolare articolo. Attraverso il Web 2.0 questo meccanismo si é evoluto e lo si può notare da alcuni studi che mostrano come il comportamento degli utenti online si sia modificato negli ultimi tempi[1].

Queste ricerche hanno dimostrato che l'81% degli utenti su Internet hanno svolto almeno una volta un'indagine su un prodotto da acquistare; tra il 73% e l'87% dei lettori di recensioni di ristoranti, hotel o atri servizi, sono stati influenzati nella loro scelta. Inoltre, i consumatori

[1] http://www.comscore.com/Insights/Press-Releases/2007/11/Online-Consumer-Reviews-Impact-Offline-Purchasing-Behavior;
http://www.pewinternet.org/files/old-media/Files/Reports/2008/PIP_Online%20Shopping.pdf.pdf

riferiscono di essere disposti a pagare dal 20% al 99% in più per un elemento che possiede 5 stelle all'interno di una recensione rispetto a un elemento a 4 stelle e, il 32% degli utenti di Internet, ha fornito una valutazione riguardante un prodotto, un servizio o una persona.

Questi dati dimostrano come l'utente sia influenzato dall'opinione delle altre persone, espressa in questo caso tramite le stelle inserite dall'utente insieme al testo della recensione. La maggior parte degli UCG, come ad esempio i post provenienti dai social media, non hanno però nessun valore associato dall'utente che ne esprima l'opinione. Proprio per questo motivo molte organizzazioni decidono di utilizzare sempre più frequentemente tecniche di Sentiment Analysis, per analizzare e monitorare la reputazione online di cui gode il loro brand, focalizzando l'attenzione sia sugli aspetti quantitativi sia su quelli qualitativi che tracciano la positività o la negatività delle opinioni e il relativo grado di intensità emotiva.

Con Sentiment Analysis si intende il campo di applicazione che prova ad identificare in maniera automatica informazioni di tipo soggettivo provenienti da testi scritti o parlati. Queste tecniche si pongono come alternativa ai più classici metodi che permettono di svolgere ricerche di mercato basate su indagini di soddisfazione, questionari e altre operazioni. Infatti, nel caso in cui gli utenti che generano UCG costituiscano effettivamente la popolazione di riferimento per l'analisi da svolgere, si avrebbero risultati migliori con un ampio risparmio in termini di costi di realizzazione e tempistiche.

Utilizzando i risultati ottenuti nel processo di Sentiment Analysis é possibile generare specifiche politiche di marketing basate sulle opinioni dei consumatori. In questo modo, l'azienda in questione, ha la possibilità di confrontarsi con i propri competitori e l'opportunità di raggiungere tassi di successo elevati [32].

Gli utenti sul Web scrivono e commentano facendo uso di diverse lingue; per questa ragione risulta fondamentale differenziare le impressioni dei consumatori a seconda della nazione di appartenenza, in modo da poter applicare politiche di marketing pensate per ciascun paese.

Le tecniche di Sentiment Analysis posso essere utili non solo in ambito aziendale, ma in svariati campi di applicazione come l'istruzione, l'insegnamento a distanza [1], la sanità e, in particolar modo, l'ambito delle disabilità [15].

All'interno di questo innovativo scenario hanno particolare importanza i social media in tutte le loro forme: blog, wiki, social network, poiché sono considerati come una fonte inesauribile d'informazioni sensibili.

Un particolare servizio di social networking e microblogging é Twitter, che permette agli utenti iscritti alla piattaforma di condividere messaggi con una lunghezza massima di 140 caratteri. Le caratteristiche più importanti di questa piattaforma sono: il grande volume di informazioni che si muovo in tempo reale, la varietà di argomenti di cui si discute al suo interno e la dimensione ridotta dei suoi messaggi rispetto ad altre piattaforme come ad esempio Facebook.

All'interno di tutti i Social Media vengono messe a disposizione di sviluppatori esterni apposite API. Le API rappresentano un modo per esporre al mondo esterno le funzionalità della piattaforma in modo da poter creare nuovi prodotti e applicazioni. Di solito possono essere di tipo REST (Representational State Transfer) [31], il cui compito é quello di rispondere ad apposite richieste provenienti dal client o di tipo STREAM, che mettono a disposizione del client un continuo flusso d'informazioni in tempo reale provenienti dai social media.

Analizzando la letteratura [34] si può notare che la maggior parte dei lavori in campo di Sentiment Analysis sono sviluppati prettamente in lingua inglese. Inoltre, gli approcci utilizzati per questo processo sono due: Machine Learning e Lexicon-based (vedi sezione 3.3 di [34]), che hanno entrambi lo scopo di calcolare la connotazione di un testo scritto in termini di positività, negatività o neutralità.

L'approccio Machine Learning rappresenta una parte fondamentale nel campo dell'intelligenza artificiale e una soluzione al problema di classificazione. Si tratta di un processo a due fasi: in primo luogo viene addestrato un modello attraverso delle osservazioni e, secondariamente, viene utilizzato l'algoritmo per classificare i dati mai visti. Le prestazioni di questo tipo di algoritmo si basano sulla qualità e la quantità di dati utilizzati come osservazioni. In particolare per ottenere risultati più significativi é necessario focalizzare l'attenzione su un solo dominio di applicazione come i post di Twitter, o le recensioni di film, prodotti o applicazioni.

L'approccio Lexicon-based é basato sullo sfruttamento di un lessico composto da una lista di termini affiancati da un valore di polarità che ne indica la connotazione positiva, negativa o neutrale. Tramite un algoritmo chiamato rules-based si prende il testo da analizzare e si cercano le corrispondenze di parole all'interno del lessico estraendo i valori di polarità dei singoli termini e calcolando il valore totale di polarità. Questo approccio é utile per poter svolgere analisi in modo veloce e su grandi quantità di dati, oltre a non essere prettamente dipendente ad un solo dominio di applicazione. Per questi motivi all'interno di questo lavoro verrà approfondito l'approccio Lexicon-based.

I lessici utilizzati all'interno di questi approcci sono spesso ottenuti tramite algoritmi di propagazione semantica in cui, partendo da una risorsa monolingue contenente i valori di polarità dei singoli termini, si cerca di ottenere un nuovo lessico in un altra lingua, cercando di collegare le parole in base alla loro relazione semantica e propagandone quindi il valore di polarità.

Il primo obiettivo della tesi in questione é la costruzione di un algoritmo di propagazione, necessario alla creazione di un insieme di lessici specifici per il numero più alto di lingue possibile. Il lavoro parte dalle risorse esistenti per lingua inglese come ad esempio Princeton WordNet [25] e SentiWordNet [2], trasportandole su un dominio multilingue. All'interno del lessico devono inoltre essere contenute due categorie di

elementi fortemente utilizzati al giorno d'oggi in particolare nei social media, quali emoticon[2] ed emoji[3] che devono essere specificate con il loro valore di polarità.

Il secondo obiettivo é la realizzazione di uno strumento, che sfrutti l'approccio Lexicon-based, in grado di analizzare testi scritti nelle lingue di cui é stato costruito il lessico, cercando di mantenere le stesse prestazioni che si ottengono comunemente in letteratura per la lingua inglese. Lo strumento ottenuto, dato in ingresso un testo e la lingua in cui é espresso, computa un valore di polarità finale sfruttando le corrispondenze delle parole del testo con quelle all'interno dei lessici multilingue. Inoltre, durante il calcolo viene verificata la presenza di alcune forme sintattiche e grammaticali come negazioni, intensificatori, congiunzioni di contrasto e punteggiatura che possono modificare la polarità finale.

All'interno dei social media, si muovono un gran numero d'informazioni in tempo reale, per questo motivo, **il terzo obiettivo, riguarda il funzionamento dell'algoritmo all'interno di applicazioni che svolgono analisi di dati in real-time.** Per farlo é stato importante analizzare la struttura classica degli strumenti Lexicon-based, valutando quali fossero le parti necessarie e quelle superflue in modo da ottenere uno strumento che fosse il più leggero ed efficiente possibile.

Il quarto obiettivo é la realizzazione di un interfaccia utile per esporre un servizio Web di Sentiment Analysis in maniera del tutto trasparente all'utente. Per fare questo é stato necessario costruire un server Web sfruttando l'architettura REST che, ricevendo richieste basate sul protocollo Http contenenti i parametri lingua e testo, utilizzi l'algoritmo Lexicon-based restituendo il valore di polarità totale.

Il quinto obiettivo é legato alla creazione di un dataset multilingue costituito da un insieme di testi e dal relativo valore di polarità totale ottenuto da un'annotazione umana. Per fare questo si é scelto di sfruttare il meccanismo di rating utilizzato all'interno dei siti Web contenenti recensioni (Amazon, TripAdvisor, Google Play) in cui durante la valutazione di un elemento si argomenta la propria opinione sotto forma di testo e si indica un valore di soddisfazione numerico variabile tra uno e cinque. Tramite un apposito algoritmo di Web Scraping si ottiene una coppia di dataset provenienti da TripAdvisor e Google Play per ciascuna delle lingue utilizzate nell'algoritmo Lexicon Based.

I dataset ottenuti sono stati utilizzati all'interno di un procedimento di valutazione comparativa delle prestazione del servizio di Sentiment Analysis, in modo da dimostrare di essere comparabile con strumenti monolingue, ma allo stesso tempo, muovendosi contemporaneamente su diciassette lingue differenti. Oggetto della comparazione sono i risultati ottenuti per le lingue Inglese, Italiano, Francese, Portoghese e Greco dal progetto

[2] L'emoticon é una rappresentazione di un viso " : -) " attraverso la punteggiatura

[3] Gli emoji sono simboli pittografici, simili ad emoticon, divenuti popolari in Giappone e sono espresse utilizzando il formato Unicode (http://unicode.org/emoji/charts/full-emoji-list.html)

SentiStrenght, e quelli ottenuti dalla risorsa commerciale Dandelion[4] per Italiano e Inglese. Un ulteriore valutazione all'interno di questo lavoro é stata condotta sul dominio di applicazione dei Social Media, in particolare su tweet provenienti da un evento sportivo, con lo scopo di ottenere un valore di correttezza dello strumento per le lingue Italiano, Inglese e Spagnolo.

L'ultimo obbiettivo é la creazione di un'applicazione Web che mostri le possibilità offerte dal processo di Sentiment Analysis all'interno di un'architettura di analisi real-time di dati provenienti da Twitter. É stato sfruttato il framework SLD [3], che utilizza le principali tecnologie legate al Semantic Web, per la gestione di un flusso (STREAM) di tweet geolocalizzati e annotati con un valore di polarità. Il risultato é una interfaccia Web realizzata con le tecnologie HTML 5, CSS 3 e Javascript che mostra l'andamento degli hashtag nel tempo all'interno di una mappa.

La struttura di questo lavoro é la seguente:

- *Sezione 2:* in questa sezione viene presentato lo stato dell'arte relativo a due campi di studio fondamentali, quali il Sentiment Analysis e il Natural Language Processing (NLP) di cui vengono presentate le principali tecniche. Per la parte di NLP vengono introdotti i concetti di Tokenization, Part-Of-Speech e analisi semantica. Sono mostrate tre risorse importanti nello sviluppo di questo lavoro chiamate SentiWordNet, WordNet e Global WordNet. Per la parte di Sentiment Analysis vengono introdotti gli approcci Lexicon-based e Machine Learning. Successivamente sono presentate alcune tecnologie abilitanti, come Semantic Web, architetture REST e Web Scraping, necessarie per capire il significato e l'utilità dell'intero lavoro.
- *Sezione 3:* si presenta il dominio del problema, definendo in maniera più specifica obiettivi e requisiti.
- *Sezione 4:* si esplicita la soluzione del problema a livello concettuale, mostrando le scelte architetturali necessarie per la creazione di un algoritmo di propagazione, di un algoritmo Lexicon-based e di un server REST.
- *Sezione 5:* si ripercorre l'architettura esplicitata nella sezione 4 scendendo però a livello di esperienza implementativa, presentando le scelte relative ai framework utilizzati e mostrando alcuni frammenti di codice fondamentali.
- Sezione 6: si espone il metodo di valutazione seguito per il calcolo delle metriche relative alle prestazioni dello strumento. Si parte dalla creazione dei dataset contenenti il valore reale di polarità e si prosegue mostrando la distribuzione del dataset fino ad arrivare al calcolo vero e proprio delle metriche di valutazione.
- *Sezione 7:* si definisce il processo seguito per lo sviluppo di un caso reale di applicazione relativo alla visualizzazione dell'andamento del Sentiment degli hashtag all'interno di

[4] https://dandelion.eu/

Twitter. É presentata l'architettura generale e il formato dei dati utilizzati, oltre agli aspetti concettuali e implementativi.

- Sezione 8: si presentano le conclusioni relative al lavoro completo, con uno sguardo dettagliato sui possibili sviluppi futuri.
- Appendice|: si mostrano le tabelle complete costituite con i risultati ottenuti nella fase di valutazione.

2. STATO DELL'ARTE

Il presente capitolo analizza gli aspetti teorici che verranno ripresi con maggiore attenzione nelle prossime pagine di questo lavoro. La prima sezione é dedicata all'introduzione di un campo di studio chiamato Natural Language Processing approfondendo alcuni particolari concetti quali il POS tagging, l'analisi semantica e presentando alcune implementazioni di database lessicali come Princeton WordNet, SentiWordNet e Global WordNet. Successivamente, vengono fornite le basi per introdurre il processo di Sentiment Analysis, presentando gli algoritmi e gli strumenti più importanti citati in letteratura. Nella terza sezione sono espressi i concetti di Social Media, Social Media Analytics e in particolare viene focalizzata l'attenzione su Twitter. Nella quarta sezione vengono presentate le principali tecnologie abilitanti necessarie all'implementazione di alcune parti del lavoro.

2.1. Natural language Processing

Natural Language Processing (NLP) é un'area di ricerca e applicazione che studia come utilizzare i calcolatori per comprendere e manipolare testi scritti o parlati in “Linguaggio naturale” in modo da svolgere azioni utili. Con il termine “Linguaggio naturale” si intende un linguaggio che si sviluppa naturalmente nell’essere umano, in genere nei primi anni di vita, senza alcuna pianificazione o premeditazione della propria coscienza.

Lo scopo dei ricercatori NLP é quello di raccogliere e capire come gli esseri umani utilizzano il linguaggio naturale in modo da poter sviluppare strumenti automatici capaci di poterlo manipolare. L’NLP viene utilizzata in una grande varietà di discipline, come l’ingegneria informatica, la linguistica, la matematica, la psicologia, ecc. Alcuni esempi di applicazioni comuni nel campo dell’ingegneria informatica sono machine translation[5], summarization[6], Sentiment Analysis.

I software di analisi NLP operano su più livelli: si parte delle singole parole determinando una struttura morfologica con delle procedure quali ad esempio il part-of-speech, successivamente si muovono su una visione riguardante l’intera frase in cui si può analizzare la sintattica e l’ordine delle parole e infine si passa ad un analisi più generale che cerca di estrarre il dominio di applicazione attraverso l’utilizzo di regole semantiche. Infatti, una data parola o una frase può avere più di un significato specifico a seconda del contesto o dominio e può essere correlata a molte altre parole o frasi nel contesto dato.

Al fine di comprendere il linguaggio naturale, é importante essere in grado di distinguere tra i seguenti sette livelli interdipendenti [11], che le persone utilizzano per estrarre significato dal linguaggio scritto o parlato:

- *fonetico*, necessario per l’interpretazione dei suoni all’interno di una parola o di una frase;
- *morfologico*, analisi della parola come forma composta da prefisso, suffisso e radice;
- *lessicale*, l'attribuzione di significato della parola;
- *sintattico*, si occupa di considerare la struttura delle frasi;
- *semantico*, determina il possibile significato dell’intera frase;
- *discorso*, interpreta struttura e comportamento di testi molto più ampi di una frase;
- *pragmatico*, cerca di comprendere l’intenzione con cui viene utilizzato un linguaggio all’interno di un particolare contesto.

[5] Machine Translation (traduzione automatica) - disciplina che si occupa di studiare i metodi per tradurre un testo da una lingua ad un’altra mediante programmi informatici.

[6] Summarization - disciplina che si occupa di ccalcolare il riassunto di un testo attraverso programmi informatici

In questo lavoro verranno presentati solamente gli aspetti legati ai livelli lessicale e semantico che risultano particolarmente importanti per lo svolgimento di un processo di Sentiment Analysis. A livello lessicale vengono approfonditi gli aspetti di Tokenization e POS Tag per poi spostarsi sul livello semantico analizzando la semantica lessicale e alcuni database semantico-lessicali com WordNet SentiWordNet e Global WordNet.

2.1.1. Tokenization

Il primo passo nel processo di pre-elaborazione di un architettura NLP é detto Tokenization. Si tratta del processo di scomposizione del testo in unità chiamate token. I token possono essere parole, numeri o segni di punteggiatura o qualsiasi entità atomica che debba essere analizzata singolarmente. Per fare questo, si cercano i confini delle entità all'interno del testo scomponendolo in primo luogo in token semplici corrispondenti a spazi, parole, punteggiatura, frasi per poi cercare pattern più complessi. Questo task può sembrare semplice a prima vista, tuttavia presenta molte problematiche; ad esempio, se si considerasse il punto (.) sempre come elemento finale di una frase sarebbe un errore in quanto potrebbe riferirsi a una abbreviazione, a una data o a un link.

Uno dei più semplici algoritmi di tokenization é il "whitespace tokenizer" che scompone il testo supponendo che i confini di ogni token siano dati dagli spazi bianchi. Molto spesso risulta un approssimazione grossolana per la gran parte delle applicazioni in cui é necessario l'utilizzo di questo task; infatti, nel caso di lingue come l'inglese o l'italiano le parole sono separate nella maggior parte dei casi dallo spazio, mentre in altre lingue come il cinese e il giapponese non si hanno confini ben chiari ed é quindi necessario trovare regole morfologiche e sintattiche per svolgere questo compito.

Un altro algoritmo é il "Penn Treebank Tokenizer" che é un algoritmo deterministico che sfrutta il corpus annotato TreeBank[7] per scomporre la frase permettendo di intercettare elementi come le forme contratte della lingua inglese in modo da poter separare elementi come "i'm" in "i" e " 'm " riuscendo a dividere il soggetto dal verbo.

I più potenti algoritmi di tokenization sono quelli basati sulle Espressioni Regolari con cui si riescono a cercare pattern specifici all'interno delle frasi ed é perciò possibile implementare Tokenizer che si comportano come il Penn Treebank.

Un'espressione regolare o RegEx é una sequenza di caratteri che identifica un insieme di stringhe e permette quindi di focalizzarsi su uno specifico dominio lessicale in modo che solo le stringhe appartenenti all'insieme identificato rappresentino valori validi.

Ad esempio, se si volessero evidenziare solo indirizzi email bisognerà cercare le stringhe che rispettano caratteristiche come: cominciare con una sequenza di caratteri alfanumerici, seguiti dal simbolo chiocciola, seguiti da altri caratteri alfanumerici, seguiti dal punto, seguiti

[7] Definizione Treebank da Wikipedia - https://en.wikipedia.org/wiki/Treebank

da due o tre lettere. Per rappresentare questa regola come una RegEx si utilizza una sintassi ben precisa e riconosciuta da un programma in grado di analizzare le stringhe.

2.1.2. Part-of-speech Tagging

Associato al processo di Tokenization esiste il PartOfSpeech Tagging (POS Tagging) che permette l'assegnazione di un tag che identifica la parte del discorso (POS) a ciascun token. Con la parte del discorso si rappresentano le categorie di parole che hanno stesse proprietà grammaticali come ad esempio verbi, nomi, pronomi, avverbi, aggettivi, congiunzioni.

Parola: Giornale - Tag: Nome

Parola: Andare - Tag:Verbo

Parola: Famoso - Tag:Aggettivo

Molte parole possono avere più di un POS tag associato perché hanno diversi significati rispetto al contesto in cui sono utilizzate; ad esempio la parola "faccia" può essere considerata come sostantivo indicando il volto o le superfici di un poliedro ma può anche indicare una coniugazione del verbo "fare". Ci sono molti modi in cui si possono suddividere le categorie di POS e per poter effettuare la procedura di Tagging é necessario definire uno standard. Il più comune é rappresentato da una versione molto raffinata di 45 tag raccolti in un "Penn Treebank Tagset[8]". Per svolgere questo processo esistono due diversi approcci:

- *Rule-based tagging;*
- *Stochastic Tagging;*

Il primo metodo é l'approccio più vecchio, utilizza regole scritte a mano per il tagging. Si basa sulla presenza di un dizionario o lessico etichettato con i possibili tag per ogni parola. Le regole scritte a mano vengono utilizzate nel processo di assegnamento del tag corretto quando una parola ha più di un POS possibile. La disambiguazione viene fatta analizzando le caratteristiche linguistiche della parola, i termini che la precedono e che la seguono. Ad esempio, se la parola precedente é un articolo allora la parola considerata deve essere necessariamente un sostantivo.

Il secondo metodo é invece un approccio basato su modelli stocastici in cui l'ambiguità viene risolta utilizzando un corpus di addestramento per poter calcolare la probabilità che una parola possieda un determinato tag in uno specifico contesto. Per farlo vengono utilizzati modelli come ad esempio Markov Model, Hidden Markov Model e Maximum Entropy Markov Models [23].

[8] Penn Treebank Tagset - https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

2.1.3. Semantica Lessicale

Prima di proseguire é necessario definire alcuni concetti che verranno approfonditi in questo paragrafo e saranno utilizzati spesso all'interno dei capitoli successivi. Un concetto fondamentale é quello di lessema, che può essere pensato come l'abbinamento di una forma ortografica e fonologica con una forma di rappresentazione del significato simbolico. Un lessico é dunque una lista finita costituita da lessemi. Si userà il termine lemma per indicare la forma ortografica e la forma fonologica mentre con il termine sense si rappresenta la componente relativa al significato.

La semantica lessicale studia il significato dei lessemi e le connessioni che intercorrono fra loro. Tra i lessemi e i loro sense si definiscono alcune relazioni:

- *Omonimia* (*Homonymy*), é la relazione più semplice di tutte e si verifica quando due parole condividono la stessa forma ortografica ma hanno significati completamente diversi; ad esempio la parola vite identifica la pianta da cui si produce il vino ma rappresenta anche il plurale di vita;
- *Polisemia* (*Polysemy*), con polisemia indichiamo la proprietà di una parola di esprimere più significati: ad esempio "quadro" può essere usato sia per indicare l'oggetto fisico sia nella locuzione "fare il quadro di", con l'intenzione di dare una panoramica chiara di un dato argomento (es. "Facciamo il quadro del discorso"). In questo caso possiamo estendere la definizione di lessema indicandolo come l'abbinamento tra una forma ortografica e fonologica e un set (gruppo) di significati. Il primo problema che si verifica é come poter distinguere questa proprietà dall'omonimia. Nella polisemia si ha che i significati sono connessi sia etimologicamente sia semanticamente in quanto la parola é sempre stata la stessa ma ha subito un estensione del significato nel tempo, mentre nell'omonimia le due parole hanno assunto la stessa forma morfologica per una serie di controversie etimologiche. Un esempio di polisemia é la parola "navetta" il cui significato primario era "contenitore della spola" ma attraverso un evoluzione storica adesso viene anche utilizzata per indicare la navicella spaziale. La parola "lira" invece può rappresentare lo strumento musicale o la moneta un tempo in vigore in Italia. Questi due significati sono però completamente separati e non hanno nessun tipo di legame semantico.
- *Sinonimia* (*Synonymy*), rappresenta la relazione in cui due lessemi hanno differente forma ortografica e fonologica ma stesso significato; per poter esprimere in maniera corretta la definizione di "stesso significato" si applica un concetto di sostituibilità in cui due lessemi possono essere considerati sinonimi se possono essere sostituiti all'interno della stessa frase senza cambiarne il significato; un esempio può essere la parola abitazione e la parola casa; Il concetto inverso della sinonimia é detto *antonomia* (*antonym*) in cui due lessemi esprimo un concetto opposto;

- *Meronimia* (*Meronymy*), con questa proprietà si indica una relazione in cui un lessema rappresenta un costituente o una parte di un altro lessema come ad esempio la parola nave che sarà meronimo della parola flotta; la relazione opposta é l'*olonimia* (*holonymy*);
- *Iponimia* (*Hyponymy*), con il termine iponimia si indica una relazione tra due lessemi in cui uno può essere considerato come sottoclasse dell'altro. Un esempio é la parola sogliola che é iponima di pesce che rappresenta appunto un elemento più generale. La relazione inversa si definisce *iperonimia* (*hypernymy*). Questa relazione vale se applicata ai sostantivi mentre per i verbi prende il nome di *troponimia* (*Troponym*): un verbo V1 si dice troponimo se indica un caso particolare del più generico verbo V2 ad esempio la parola passeggiare é troponimo di camminare.
- Implicazione (entailment), é utilizzata solo per i verbi e indica una relazione di implicazione tra due lessemi. Il verbo V1 é un'implicazione del verbo V2 se nel fare V1 si deve per forza fare V2 (come russare rispetto a dormire).

2.1.4. Princeton WordNet

Princeton WordNet[9] (PWN) é uno dei più importanti database lessicali che collega verbi, nomi, aggettivi e avverbi della lingua inglese attraverso le relazioni tipiche della semantica lessicale. In PWN il lemma é rappresentato da una sequenza di caratteri ASCII, mentre il sense é rappresentato da un gruppo di uno o più sinonimi che hanno lo stesso significato. PWN contiene più di 118.000 Lemma differenti e più di 90.000 diversi Sense, e più in generale contiene 166.000 coppie (Lemma, Sense). Circa il 17% delle parole sono polisemiche e circa il 40% possiede uno o più sinonimi [25]. Al suo interno non sono contenute preposizioni, congiunzioni o pronomi in quanto non hanno una rappresentazione semantica importante. Per quanto riguarda le forme flesse invece, sono inserite come Lemma separati non esplicitando le connessioni morfologiche tra loro. PWN é sviluppato dal Cognitive science Laboratory presso l'università di Princeton. É usufruibile on-line insieme al materiale e alla documentazione relativa.

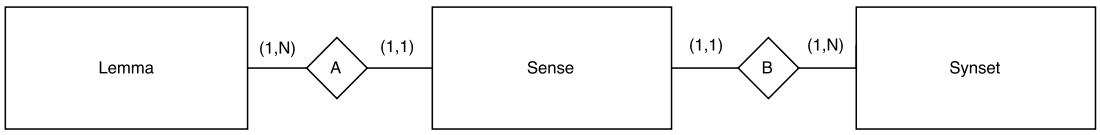


Figura 1: Schema ER Lemma-Sense-Synset

[9] http://www.cogsci.princetn.edu/~wn/

Il concetto fondamentale della struttura di WordNet é l'associazione tra Lemma (la forma scritta o il suono) e Sense (il concetto associato), rappresentata nella figura 1 con un apposito schema E/R. Esiste una relazione di tipo molti-a-molti che implementa i concetti di sinonimia e polisemia. Un gruppo di Lemma tra loro sinonimi é chiamato synset [26].
All'interno di PWN vengono inoltre rappresentate tutte le relazioni presentate nel paragrafo 2.1.3 che possono essere suddivise in due categorie:

- ***relazioni semantiche,*** che coinvolgono due synset e sono valide per tutti i lemmi ad esso collegati (ad esempio Iponimia/Meronimia);
- ***relazioni lessicali,*** che stabiliscono un nesso tra due singoli lemmi (ad esempio un contrario non é detto che sia valido per tutti i termini di un synset, ma solo per uno in particolare) .

Da questo momento in poi la parola WordNet verrà utilizzata per identificare un qualunque database lessicale generico. Indicheremo invece con Princeton WordNet (PWN) la versione più famosa relativa alla lingua inglese.

2.1.5. SentiWordNet

Lo scopo di SentiWordNet[10] é quello di fornire un'estensione per PWN, in modo da poter introdurre un elemento legato al sentimento che possa attribuire una connotazione di positività, negatività, oggettività a ciascun synset. SentiWordNet 3.0 é la versione migliorata del SentiWordNet 1.0 é gratuitamente disponibile per scopi di ricerca e usufruibile tramite apposita interfaccia Web.

Questa estensione etichetta ogni synset con tre valori, la cui somma è compresa tra 0 e 1, che rappresentano positività, negatività, oggettività; la somma dei tre valori é sempre 1.0, in modo che ogni synset possa avere un valore diverso da zero per ogni categoria. Il vantaggio di associare i valori ai synset invece che alle parole é quello di offrire diversi punteggi di sentimento per ogni sense di una parola, perché i tre elementi possono differenziarsi in una parola a seconda del suo significato.

Il metodo con cui é stato costruito SentiWordNet si basa sull'addestramento di una serie di classificatori ternari[11], ciascuno dei quali in grado di decidere se un synset é positivo, negativo o oggettivo. Ogni classificatore si differenzia dagli altri per il dataset utilizzato per addestrarlo, producendo quindi risultati diversi per ciascun synset di PWN. I punteggi finali di opinione per ogni synset sono determinati da una proporzione (normalizzata) dei valori calcolati dai classificatori ternari. Se tutti concordano nell'assegnare la stessa etichetta ad un

[10] http://sentiwordnet.isti.cnr.it/

[11] Un classificatore n-ario é un attribuisce a ciascun oggetto esattamente un'etichetta tra un insieme predefinito di n etichette.

synset, allora il punteggio sarà massimo, altrimenti ogni etichetta avrà un punteggio proporzionale al numero di classificatori che lo hanno assegnato [2].

2.1.6. Global WordNet

Global WordNet[12] é un progetto il cui scopo é quello di rendere possibile l'utilizzo della maggior parte dei WordNet disponibili in letteratura in maniera facile e intuitiva. É stato costruito prendendo in considerazione i numerosi WordNet disponibili per un gran numero di lingue straniere con l'obiettivo di arricchirli e convertirli in un formato comune. In generale esistono più di 60 lingue di cui esiste un WordNet in stato di sviluppo; chiaramente la qualità di ognuna di queste risorse dipende da come é stata realizzata. Sono inoltre presenti progetti che sviluppano risorse per gruppi di lingue come ad esempio EuroWordNet[13], AsianWordNet[14] o BalkaNet[15].

		Projects			Wiktionary			Merged (+CLDR)		
ISO	**Language**	**Synsets**	**Senses**	**Core**	**Synsets**	**Senses**	**Core**	**Synsets**	**Senses**	**Core**
eng	English	117,659	206,978	100	35,400	49,951	75	117,661	213,538	100
fin	Finnish	116,763	189,227	100	21,516	31,154	65	116,830	199,435	100
tha	Thai	73,350	95,517	81	2,560	3,193	17	73,595	97,390	81
fra	French	59,091	102,671	92	20,449	27,150	63	61,258	109,643	95
jpn	Japanese	57,179	158,064	95	12,685	19,479	52	59,112	166,617	96
ind	Indonesian	52,006	142,488	99	2,390	2,810	17	52,154	143,755	99
cat	Catalan	45,826	70,622	81	8,626	10,251	36	48,007	74,806	84
spa	Spanish	38,512	57,764	76	18,281	25,310	60	47,737	74,848	86
por	Portuguese	41,810	68,285	79	12,331	16,178	53	43,870	74,151	84
zsm	Standard Malay	42,766	119,152	99	2,833	3,744	19	43,079	120,686	99
ita	Italian	34,728	60,561	83	14,605	18,710	53	38,938	68,827	87
eus	Basque	29,413	48,934	71	1,693	1,943	11	29,965	49,945	72
pol	Polish	14,008	21,001	30	10,888	13,431	46	20,975	30,943	55
glg	Galician	19,312	27,138	36	2,492	2,871	15	20,772	29,136	42
fas	Persian	17,759	30,461	41	4,229	5,443	26	20,766	35,318	55
rus	Russian	0	0	0	19,983	33,716	64	20,138	34,009	64
deu	German	0	0	0	19,675	29,616	64	19,857	29,884	64
cmn	Mandarin Chinese	4,913	8,069	28	12,130	19,079	49	15,490	27,113	60
arb	Standard Arabic	10,165	21,751	48	6,892	9,337	38	14,861	31,337	63
nld	Dutch	0	0	0	13,741	19,709	56	13,950	20,003	56
ces	Czech	0	0	0	12,802	15,493	54	13,030	15,813	54
swe	Swedish	0	0	0	12,000	16,226	51	12,221	16,512	51
ell	Modern Greek	0	0	0	10,308	13,071	44	10,549	13,472	44
dan	Danish	4,476	5,859	81	7,290	8,931	35	10,328	13,551	85
nob	Norwegian Bokmål	4,455	5,586	79	7,262	9,170	35	10,322	13,612	83
hun	Hungarian	0	0	0	9,964	12,699	45	10,213	13,029	45

Core shows the percentage coverage of the 5,000 core concepts.

Figura 2: Risultati ottenuti dal progetto Global WordNet

Il primo WordNet costruito é stato PWN, tutti gli altri sono venuti successivamente e hanno cercato di seguire lo stesso processo di costruzione. In alcuni casi, come nel progetto MultiWordNet per l'italiano, è stato proprio utilizzato come punto di partenza per la

[12] http://globalWordNet.org/

[13] http://www.illc.uva.nl/EuroWordNet/

[14] http://www.asianWordNet.org/

[15] http://www.dblab.upatras.gr/balkanet/

creazione della risorsa nella nuova lingua attraverso un metodo chiamato Expand Model. Si tratta di un approccio basato sul presupposto che tra stessi concetti, in lingue differenti, esistono le stesse relazioni. Se due synset in un WordNet sono legati da una relazione, i due synset equivalenti in un altro WordNet saranno legati dalla stessa relazione. Per questo motivo il collegamento tra tutti i WordNet disponibili é stato effettuato utilizzando i synset di PWN come punto di partenza.

Successivamente i WordNet sono stati arricchiti sfruttando una risorsa strutturata come Wikitionary[16] che rappresenta un progetto gratuito il cui obiettivo é quello di produrre un dizionario online multilingue con significati, etimologie e pronunce.

Le risorse finali vengono poi raccolte in un database SQLite utilizzando lo schema prodotto dal progetto WordNet giapponese [19]. Il database si basa sulla struttura logica del Princeton WordNet, con l'aggiunta dell'attributo supplementare relativo alla lingua per lemma e sense. I risultati finali sono visualizzati in figura 2 [6].

2.2. Sentiment Analysis

"Sentiment Analysis" o "Opinion mining" é il campo di studio che si occupa di analizzare, sfruttando tecniche di Natural Language Processing opinioni, sentimenti, atteggiamenti ed emozioni espressi delle persone, verso entità come prodotti, servizi, eventi o argomenti. Si muove su un vasto spazio di applicazione che copre compiti differenti; é per questo motivo che vengono utilizzati molteplici nomi per identificarlo: sentiment analysis, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis. All'interno di questo lavoro verranno utilizzati solo i primi due in maniera totalmente analoga e intercambiabile.

Le opinioni sono al centro di quasi tutte le attività umane, perché influenzano in maniera decisiva i nostri comportamenti. Ogni volta che é necessario prendere una decisione, si cerca di conoscere le opinioni altrui. Nel mondo reale imprese e organizzazioni cercano sempre di trovare un modo per ottenere opinioni dai consumatori sui loro prodotti e servizi. In maniera analoga i consumatori vogliono conoscere le opinioni degli utenti che hanno già utilizzato un prodotto prima di acquistarlo, o le opinioni sui candidati politici prima di prendere una decisione di voto in una elezione politica. In passato, quando un'organizzazione o un business necessitavano di opinioni pubbliche o di consumo, venivano condotte indagini, sondaggi e focus group. L'acquisizione di informazioni riguardo l'opinione pubblica e dei consumatori é stata a lungo un enorme business per il marketing, le relazioni pubbliche, le aziende e le campagne politiche.

Con l'avvento del Web 2.0 la situazione ha subito un cambiamento drastico infatti, sono nati numerosi mezzi, attraverso i quali le persone possono esprimere opinioni riguardo a

[16] https://en.wiktionary.org/wiki/Wiktionary:Main_Page

qualsiasi cosa. Blog, wiki , forum e social network sono esempi di tali mezzi, attraverso i quali gli utenti connessi da tutto il mondo possono inserire informazioni ed esprimere pareri, ottenendo in tal modo un feedback da altri utenti. Di per sé, essi rappresentano collettivamente una ricca fonte di informazioni su argomenti diversi, che spaziano dalla politica alla salute, fino ad arrivare alle recensioni di prodotti, di hotel o ristoranti. In questo modo, si rende disponibile in formato digitale una grande mole di informazioni espresse in un numero svariato di lingue e idiomi che ha bisogno di essere correttamente analizzate. Risulta quindi importante riuscire ad identificare un'opinione soggettiva all'interno di un testo scritto etichettandola con un valore che possa esprimere la sua connotazione.

La gamma di emozioni umane é così vasta e articolata da rendere complessa l'individuazione di emozioni base. Esiste un tipo di rappresentazione che divide la gamma di emozioni in sei elementi base: rabbia, disgusto, paura, gioia, tristezza, sorpresa [12]. Questo tipo di classificazione é molto rappresentativa ma non abbastanza specifica. Infatti, con questa teoria non emerge con evidenza l'attribuzione di positività e negatività rispetto a ogni singola emozione.

Per questo motivo si é cercato di raggruppare le emozioni di base lungo quattro dimensioni: gioia e tristezza, accettazione e disgusto, attesa e sorpresa, paura e rabbia [36]. Tuttavia, una tale divisione richiede algoritmi molto complessi e non sempre applicabili a tutte le situazioni. Per rendere più immediata l'analisi la maggior parte dei ricercatori utilizza il concetto di polarità di un sentimento che rappresenta un punto su una scala di valutazione che corrisponde alla nostra idea di positività o negatività [29]. Il processo di Sentiment Analysis si può ridurre ad un processo di classificazione di un testo con un'etichetta che ne distingue l'opinione nei due caratteri: positivo e negativo. Molto spesso ai concetti di positività e negatività viene aggiunto quello di neutralità che ci permette di svolgere l'analisi di soggettività di un testo. Un testo neutrale é un testo in cui viene espressa un'opinione oggettiva e imparziale, quindi non etichettabile come positiva o negativa.

2.2.1. Tecniche Sentiment Analysis

Tramite la pubblicazione "Survey on Mining Subjective Data on the Web" [34] é stato possibile approfondire i metodi presenti in letteratura per lo svolgimento del processo di Sentiment Analysis, in modo da poter valutare le prestazioni per ciascuna tecnica al fine di espanderla su una visione multilingue.

Come possiamo vedere nell'immagine di figura 3 i due approcci fondamentali sono chiamati: Lexicon-based, basato su lessici automaticamente o manualmente costruiti e Machine Learning, fondato sull'addestramento di un modello attraverso l'utilizzo di appositi

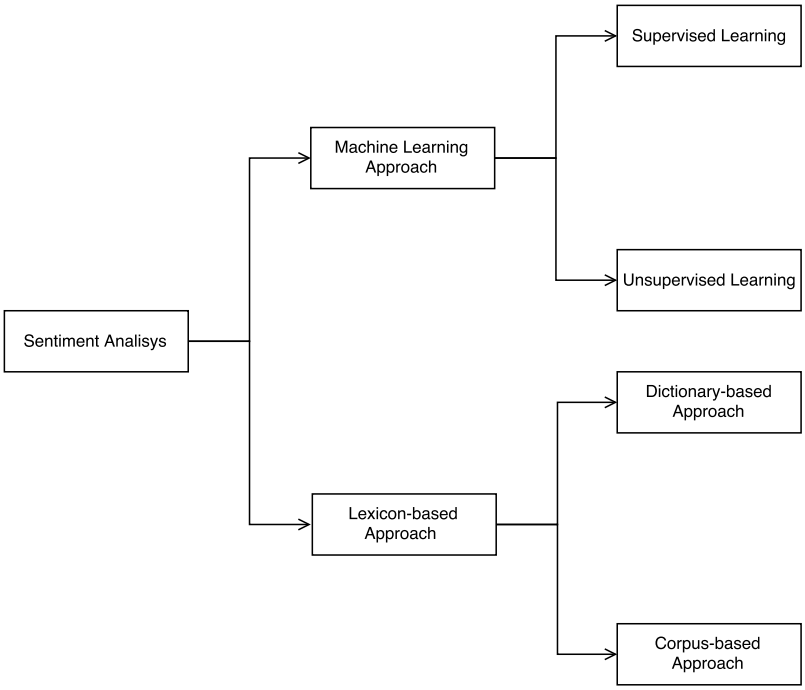


Figura 3: Tecniche Sentiment Analysis

corpus[17]. I corpus sono grandi raccolte di testi organizzati e selezionati in modo da poter esseri usati all'interno di analisi linguistiche.

L'approccio Machine Learning é una soluzione sofisticata e può essere a sua volta suddivisa nei metodi supervised e unsupervised, mentre per quanto riguarda l'approccio Lexicon-based si ha una suddivisione tra Dictionary-Based e Corpus-Based.

2.2.2. Approccio Machine Learning

"Machine Learning" (apprendimento automatico) rappresenta una delle aree fondamentali dell'intelligenza artificiale e si occupa della realizzazione di sistemi e algoritmi che utilizzano dati esistenti per la sintesi di nuovi modelli di conoscenza. L'apprendimento può avvenire catturando caratteristiche di interesse provenienti da esempi, strutture dati o sensori, per analizzarle e valutarne le relazioni tra le variabili osservate"[18].

Come é stato spiegato all'interno del paragrafo 2.2.1 l'approccio Machine Learning (ML), per svolgere il processo di Sentiment Analysis, si suddivide in due sottoclassi chiamate supervised learning (SL) e unsupervised learning (UL).

[17] Definizione da Wikipedia - https://en.wikipedia.org/wiki/Text_corpus

[18] Definizione da Wikipedia - https://en.wikipedia.org/wiki/Machine_learning

Nel primo caso si utilizzano una grande quantità di testi etichettati con valori di polarità per effettuare l'addestramento del modello che verrà poi utilizzato per la fase di classificazione. Le etichette presenti all'interno di questi testi sono spesso attribuite attraverso annotatori umani.

Nel secondo caso si tratta di algoritmi utilizzati quando non si dispone di risorse etichettate o risultano difficili da ottenere per il campo di applicazione desiderato. In tal caso l'addestramento del modello viene fatto solamente utilizzando i testi in cui verranno cercate similitudini e differenze in modo da poter successivamente effettuare la classificazione. Uno dei metodi più importanti si basa sulla ricerca di pattern sintattici che sono spesso utilizzati per esprimere un opinione; questi pattern sono composti utilizzando il POS Tagging [35]. Esistono anche dei metodi ibridi che utilizzano entrambi gli approcci sopra citati. Nella fase di apprendimento del metodo SL é molto importante la scelta delle features[19] (e.g., frequenza delle parole, bi-grams, etc.) con cui viene rappresentato il testo. Gli algoritmi che si utilizzano in questo tipo di approccio sono Naive Bayes, SVM Maximum Entropy (ME) e Support Vector Machines (SVM) [30].

Il problema principale di questi metodi é dato dal fatto che le performance finali dipendono dalla qualità dei testi utilizzati per gli algoritmi di apprendimento oltre che dalla velocità di esecuzione che per queste operazioni é spesso molto lungha. Inoltre, questi algoritmi si adattano al dominio di applicazione nei quali vengono addestrati degradando nelle prestazioni se applicate in altri domini. Un esempio può essere un algoritmo Naive Bayes addestrato utilizzando delle review di Prodotti e utilizzato per la classificazione di testi provenienti da Twitter. In questo caso chiaramente ci sarà una perdita di accuratezza che invece non si avrebbe se la classificazione venisse effettuata sullo stesso dominio dell'addestramento.

2.2.3. Approccio Lexicon Based

In questo tipo di approccio risulta molto importante identificare particolari classi di parole chiamate opinion words che indicano opinioni positive o negative che, insieme al loro valore di polarità, andranno a costituire un lessico. Infatti, il problema fondamentale di questi metodi sta nella costruzione del lessico che verrà utilizzato successivamente per la fase di analisi. Ci sono tre metodi per poter costruire un lessico e sono: approccio dictionary-based, approccio corpus-base e approccio manuale.

Nell'approccio dictionary-based la costruzione del lessico può essere eseguita partendo da una lista di parole dette seed (semi), raccolte manualmente e annotate con il proprio valore di polarità e successivamente viene ampliata tramite l'utilizzo di risorse semantiche come ad esempio PWN sfruttando le relazioni di sinonimia e antonimia rispetto alle parole seed,

[19] https://en.wikipedia.org/wiki/Feature_(machine_learning)

questo processo viene continuato iterativamente fino a che non vengono più trovate nuove parole [17]. In particolare questo tipo di approccio prende il nome di Semantic-Based. Ci sono altri metodi per la costruzione dei lessici che sfruttano metodi di supervised o unsupervised machine learning come ad esempio nel caso di SentiWordNet. In questo tipo di approcci si possono anche utilizzare dizionari già presenti in letteratura tra cui i principali per la lingua inglese dizionari General Inquirer[20], Dictionary of Affect of Language[21], the WordNet-Affect[22], e appunto SentiWordNet. Il problema all'interno di questo tipo di lessici risulta essere il loro carattere generale, che molto spesso non riesce a cogliere le differenze di polarità legate alle parole utilizzate in contesti diversi.

Per cercare di risolvere questo problema viene utilizzato un approccio corpus-based che utilizza un insieme di corpus molto ampi insieme all'utilizzo di modelli statistici per costruire i lessici finali. In generale, si parte anche in questo caso da un insieme di parole seed (con carattere generale) che vengono ampliate attraverso corpus provenienti da domini specifici. Sono utilizzate regole per espandere la lista dei seed tramite la ricerca di pattern statistici all'interno dei corpus come ad esempio la co-occorrenza. Infatti, secondo questo metodo se un aggettivo compare in una frase accanto ad un aggettivo positivo (seed) allora anch'esso sarà positivo. Un altra regola esprime il concetto per cui due parole che vengono utilizzate spesso all'interno dello stesso contesto allora, con un alta probabilità, condivideranno la stessa polarità. In particolare questo metodo prende il nome di Statistical-based Approach.

Il terzo metodo chiamato approccio manuale consiste nel costruire il lessico attraverso annotatori umani che attribuiscono un valore di polarità a un elenco di parole. Questa tecnica risulta però molto intensiva e per questo viene spesso combinata con i primi due.

Una volta costruito il lessico sfruttando i metodi sopra citati é necessario utilizzarli nel processo di Sentiment Analysis. Nella maggior parte dei casi il calcolo della polarità totale viene effettuato computando la media delle polarità delle singole parole tramite algoritmi chiamati rules-based. La formula generale é:

$$\frac{\sum_{w \in t} S_w * WEIGHT(w) * MODIFIER(w)}{\sum WEIGHT(w)}$$

Dove con t indichiamo il testo in considerazione, con w indichiamo le parole che compongono il testo e con Sw indichiamo la polarità di una singola parola.

Vengono poi utilizzate due funzioni *weight()* e *modifier()* necessarie ad effettuare una procedura di peso rispetto a particolari parole che definiscono ad esempio negazioni o

[20] http://www.wjh.harvard.edu/~inquirer/

[21] http://www.hdcus.com/

[22] http://wndomains.fbk.eu/wnaffect.html

intensificatori. Un esempio potrebbe essere una funzione di *weight()* uguale a 1 nelle vicinanze di parole che identificano uno specifico argomento mentre uguale a zero altrove. Esistono numerosi tool presenti in letteratura che svolgono il processo di Sentiment Analysis sfruttando questo approccio. Si é deciso di focalizzare l'attenzione su Pattern[23], Vader[24] e SentiStrenght[25] .

Pattern é un modulo Python utile per lo svolgimento di operazioni di web mining, in particolare contiene un sottomodulo Pattern.en dedicato all'ambito del NLP tra cui Sentiment Analysis sfruttando un lessico composto secondo il metodo semantic-based utilizzando PWN come risorsa semantica [8]. Insieme a Pattern.en vengono fornite anche le versioni per francese, italiano, tedesco spagnolo e olandese anche se attualmente non é ancora stato implementato il modulo relativo al Sentiment Analysis.

Vader é anch'esso un modulo sviluppato per Python il cui obiettivo é lo svolgimento del processo di Sentiment Analysis su testi in lingua inglese provenienti dai Social Media. In questo caso il lessico é stato costruito utilizzando tre dizionari annotati con la polarità per la lingua inglese LIWC[26], ANEW[27], e General Inquier. Partendo da qui, é stato arricchito il dizionario inserendo parole ed espressioni comuni all'interno dei Social Media che sono state annotate manualmente. Il calcolo finale della polarità viene effettuato applicando un insieme di regole sintattiche e grammaticali utilizzate spesso dagli esseri umani per intensificare un'opinione [18].

Anche nel caso di SentiStrenght (SS), come per Vader, il lessico per la lingua inglese é costruito sfruttando delle risorse già esistenti ed in particolare Linguistic Inquiry and Word Count (LIWC). Partendo da questo dizionario sono state estratte le parole ed il loro tema; successivamente il lessico é stato arricchito attraverso modi di dire tipici della lingua inglese annotati manualmente. Il lessico contiene 2130 parole e ogni parola possiede sia un valore positivo che un valore negativo [33]. SS nasce solamente per la lingua inglese ma é stato successivamente esteso ad altre lingue calcolandone i lessici e utilizzandoli all'interno dello stesso algoritmo. In particolare vengono resi disponibili sul sito risorse per 9 lingue oltre all'inglese Arabo, Francese, Greco, Italiano, Portoghese, Svedese, Persiano, Polacco, Gallese.

[23] http://www.clips.ua.ac.be/pages/pattern

[24] https://github.com/cjhutto/vaderSentiment

[25] http://sentistrength.wlv.ac.uk/

[26] http://liwc.wpengine.com/

[27] http://csea.phhp.ufl.edu/media/anewmessage.html

2.3. Social Media

Prima di approfondire il concetto di social media é importante presentare il Web 2.0 e come i social media siano parte integrante di questo mondo.

Con Web 2.0 si intende uno stato di evoluzione del World Wide Web, rispetto alla condizione precedente legata appunto al concetto primordiale di web. Con questo termine vengono identificate tutte quelle applicazioni online che permettono uno spiccato livello di interazione sito-utente (blog, forum, chat, sistemi quali Wikipedia, Youtube, Facebook, Myspace, Twitter, Gmail, WordPress, TripAdvisor ecc.). In principio all'interno del Web 1.0 i contenuti erano di tipo prevalentemente informativo ed erano pubblicati da una piccola parte di utenti esperti in tecnologie web, mentre venivano solamente consultati dal resto delle persone. Tramite il Web 2.0 si vuole indicare il cambiamento in cui ciascun utente può allo stesso tempo consultare e creare contenuti (non più solo informativi ma anche discussioni, scambi di opinioni, contenuti multimediali) condividendoli con altri utenti, senza nessuna competenza particolare. Le infrastrutture HTTP e TCP/IP sono sempre rimaste le stesse, a evolversi sono state le tecnologie di programmazione web come ad esempio Ajax o l'avvento dei CMS (Content Management System)[28].

I social media sono molto importanti all'interno di questa nuova visione di web e possono essere definiti come:

> "gruppo di applicazioni costruite sui fondamenti ideologici e tecnologici del Web 2.0, che permettono la creazione e lo scambio di contenuti generati dall'utente all'interno di comunità virtuali"

L'universo dei social media ha ormai assunto un'importanza e una dimensione tale nella vita di tutti i giorni da non poter più essere trascurato, anche da un punto di vista sociologico. Nel 2016 gli utenti connessi al Web sono circa 3.4 miliardi ovvero il 46% della popolazione e gli utenti che possiedono almeno un account all'interno dei social media sono più di 2.3 miliardi[29]. Questi dati possono dimostrare quanto é importante trovare metodi per lo sfruttamento delle informazioni nel web 2.0 e in particolare sui social media per poter studiare opinioni e comportamenti delle masse su scala globale.

Con il termine Social Media Analytics viene rappresentato proprio l'insieme delle operazioni che permettono la raccolta dei dati dai principali social network come Facebook, Twitter, Flickr, Instagram, LinkedIn, per poi essere analizzati e valutati in modo da poter prendere delle decisioni a livello di business. Esistono numerose applicazioni che sfruttano le tecniche di Social Media Analytics abbinate con concetti di Sentiment Analysis: il progetto

[28] https://en.wikipedia.org/wiki/Content_management_system

[29] We Are Social - http://wearesocial.com/it/

Pulse of the Nation[30] ad esempio prende in considerazione 300 milioni di Tweet provenienti dagli Stati Uniti e ne traccia un andamento relativo al sentimento focalizzandosi su molteplici aspetti come ad esempio le differenze di umore degli utenti della costa est rispetto alla costa ovest.

Un particolare esempio di Social media é Twitter. Si tratta di un social network gratuito che offre un servizio di microblogging[31] e risulta uno dei maggiormente utilizzati in questo ambito. Gli utenti hanno la possibilità di interagire sia accedendo direttamente al sito web sia sfruttando le numerose applicazioni mobile e tablet. Il concetto chiave di Twitter risiede nella sua regola più semplice: vengono pubblicati testi la cui dimensione massima non deve superare i 140 caratteri. Su Twitter attualmente ci sono 320 milioni di utenti attivi e il 79% degli account é creato in un paese fuori dagli Stati Uniti, infatti al suo interno vengono supportate più di 35 lingue diverse[32]. Col passare del tempo Twitter ha assunto un ruolo di rilievo nel campo della diffusione di notizie provenienti da qualunque ambito e molto spesso risulta molto più attendibile e veloce dei media tradizionali. Un esempio può essere il terremoto in Abruzzo dell'aprile 2009, durante il quale ci furono le prime segnalazioni proprio su Twitter. Qui di seguito vengono spiegati alcuni concetti importanti utilizzati all'interno di Twitter:

- *following*: con questo meccanismo gli utenti possono decidere di seguire un utente su Twitter e ricevere aggiornamenti riguardo ai suoi contenuti;
- *hashtag "#" :* é un tipo di etichetta (tag) che rende più facile la ricerca di messaggi con un tema o un contenuto specifico;
- *reply*: un tweet in risposta ad uno precedentemente pubblicato;
- *retweet*: permette di ripubblicare un tweet di un altro utente;
- *like:* permette di dimostrare la propria approvazione riguardo un contenuto pubblicato da un altro utente e lo salva nei preferiti del proprio profilo;
- *geolocalizzazione*: sfruttando i sistemi GPS di cui sono dotati i dispositivi smartphone e tablet di ultima generazione si può inserire all'interno dei contenuti le coordinate geografiche del luogo da cui é stato effettuata la pubblicazione.

Twitter mette inoltre a disposizione un insieme di API[33] che permettono di ottenere i contenuti presenti sulla piattaforma in modo da essere sfruttati da sviluppatori in ambito

---

[30] Pulse of the Nation - http://www.ccs.neu.edu/home/amislove/twittermood/

[31] Un microblog é un luogo in cui gli utenti si scambiano brevi contenuti, come pensieri, opinioni, immagini personali, link o contenuti multimediali deriva dal blogging, dal quale differisce per la dimensione dei contenuti pubblicati, tipicamente limitati a 100-200 caratteri

[32] Twitter (December 15, 2015) - https://about.twitter.com/it/company

[33] API - https://it.wikipedia.org/wiki/Application_programming_interface

aziendale o di ricerca. Questo permette la nascita e la crescita di numerose realtà legate a Twitter che utilizzano queste informazioni ad esempio per svolgere processi di Sentiment Analysis e per trarne benefici economici. Esistono delle Politiche (Policy) create per regolamentare l'utilizzo di questi dati e definire cosa é possibile fare o non fare. Le API sono raggruppate in due categorie:

- *REST API*, permettono l'estrazione delle informazioni presenti su Twitter tramite richieste HTTP. Vengono imposte delle limitazioni al numero di richieste effettuabili che possono cambiare se si tratta di un utente anonimo o un utente registrato come sviluppatore. Le informazioni che si estraggono possono essere ad esempio, la timeline di un utente, oppure si possono effettuare apposite ricerche per reperire i trend[34].
- *stream API,* in questo caso vengono messi a disposizione un flusso di informazioni estratte in tempo reale dalla piattaforma. Rappresentano una parte dei Tweet pubblici e possono essere appositamente filtrati impostando ad esempio la lingua o una finestra geografica o una serie di termini che devono essere contenuti.

Proprio sfruttando le potenzialità di queste risorse é possibile effettuare procedure di Sentiment Analysis. In particolare, esistono molte dimostrazioni di come questo ambito associato ai social media ed in particolare a Twitter abbia avuto una grande espansione; alcuni esempi sono "Automatic detection of political opinions in Tweets" [24], in cui gli autori cercano di determinare l'orientamento politico degli utenti di Twitter nel periodo precedente alle elezioni inglesi del 2010; mentre in "Twitter Sentiment Classification using Distant Supervision" [14] si cerca di determinare l'opinione espressa dai tweets attraverso l'uso di algoritmi di apprendimento supervisionato associati alla rilevazione delle emoticon, ottenendo risultati incoraggianti. Risultati molto simili sono riscontrabili anche in "Twitter as a Corpus for Sentiment Analysis and Opinion Mining" [28].

2.4. Tecnologie abilitanti

All'interno di questo paragrafo vengono presentate le principali tecnologie necessarie per lo svolgimento di alcuni passaggi fondamentali che verranno illustrati nel prosieguo della tesi ed importanti a capirne il contesto di applicazione. In particolare viene introdotto il concetto di Semantic Web e ne viene mostrato un particolare framework chiamato Resource Description Framework (RDF). Si mostra inoltre una particolare architettura Web chiamata REST che verrà utilizzata in diversi punti del lavoro. Per finire viene mostrata una tecnica di estrazione di dati non strutturati chiamata Scraping Web.

[34] Trend - Una tendenza o trend su Twitter si riferisce ad un argomento associato ad un hashtag che diventa popolare in un dato periodo temporale

2.4.1. Semantic Web

Come é stato visto nei paragrafi precedenti il web é diventato un luogo in cui poter condividere e creare contenuti di qualunque tipo in modo da essere utilizzati attraverso i browser da utenti umani. Con il termine web semantico, utilizzato per la prima volta dal suo ideatore, Tim Berners-Lee, si intende un ulteriore evoluzione del World Wide Web in un luogo dove i contenuti pubblicati (pagine HTML, file, immagini, e così via) sono associati ad informazioni che ne evidenziano la semantica in modo da poter essere compresi e interrogati anche da un elaboratore automatico. Con questo approccio l'obiettivo é quello di creare un insieme di linguaggi per esprimere le informazioni all'interno del web in un modo che siano comprensibili dalle macchine. Ciò permetterebbe ai computer di poter combinare la conoscenza proveniente dalle diverse fonti in modo da derivarne una nuova. Un esempio potrebbe essere la combinazione delle informazioni provenienti da biblioteche diverse, in modo da poter trovare la versione originale di un libro, le informazioni sul suo autore, o su altri libri dello stesso semplicemente partendo dal suo identificatore univoco (ISBN) e percorrendo le associazioni rappresentate nelle varie fonti informative.

La rappresentazione dell'architettura generale del Semantic Web é rappresentata in figura 4. Come possiamo vedere il livello base é rappresentato dal Uniform Resource Identifier (URI) che rappresenta una stringa in un formato standard che identifica in maniera univoca una risorsa nel Web che può essere un indirizzo, un documento, un'immagine o un file. Una sua estensione é l'Internationalized Resource Identifier (IRI) che permette l'utilizzo di caratteri e formati che sono utili alla rappresentazione delle lingue diverse dall'inglese, in particolare lo standard utilizzato prende il nome di Unicode[35].

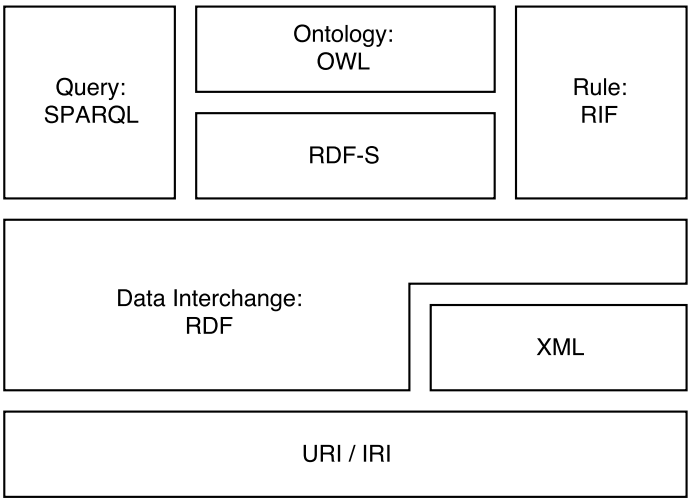


Figura 4: Architettura Semantic Web

[35] https://en.wikipedia.org/wiki/Unicode

Il livello successivo é occupato dal Extensible Markup Language (XML) che definisce una sintassi comune da utilizzare all'interno del semantic web. XML é un linguaggio di markup[36] per i documenti che contengono informazioni strutturate. La parte fondamentale per la modellazzione dei dati é fornita dal Resource Description Framework (RDF) che permette di rappresentare le risorse e le relazioni che intercorrono tra loro. Un modello RDF può essere espresso utilizzando sintassi XML. Per permettere uno standard nella descrizione delle tassonomie e nella creazione di piccole ontologie é stato creato un vocabolario chiamato RDF-S (RDF Schema).

Un'ontologia descrive parole comuni e concetti utilizzati per descrivere e rappresentare un'area di conoscenza (dominio). Può essere utilizzata da persone, applicazioni, database per condividere concetti riguardo ad un certo dominio in un modo usabile dal computer ma anche comprensibile agli umani. Ontologie più dettagliate possono essere create attraverso l'utilizzo di una famiglia di linguaggi chiamata Web Ontologia Language (OWL) che estende le potenzialità del modello RDF e permette di mappare in maniera più specifica le relazioni che intercorrono tra le risorse. L'ultimo livello é composto da SPARQL che rappresenta un termine generico che sta ad indicare sia un protocollo sia un linguaggio per interrogare modelli RDF. Il protocollo fornisce un'interfaccia utente per effettuare query in remoto verso un endpoint sfruttando HTTP e SOAP. Utilizzato invece come linguaggio permette di definire interrogazioni su modelli di dati non relazionali, ed in particolare a grafo, utilizzando parole chiave simile ad SQL (Structured Query Language) [37].

2.4.2. Resource Description Framework (RDF)

La specifica di RDF é costituita da due componenti: RDF Model and Syntax e RDF Schema. Attraverso RDF Model and Syntax viene definito un modello di rappresentazione dei dati e una sintassi espressa in formato XML per specificare i dati. Attraverso l'RDF Schema invece é possibile definire il significato e le caratteristiche delle proprietà e delle relazioni che esistono tra le risorse descritte nel modello dei dati. Una risorsa viene identificata con un URI univoco ed é descritta da un modello di dati basato su tre entità:

- *Resource* (*risorsa*): indica l'oggetto che si vuole descrivere mediante RDF e che può essere una risorsa Web (ad esempio una pagina HTML, un documento XML o parti di esso) o anche una risorsa esterna al Web (ad esempio un libro, un quadro, etc.);

[36] Un linguaggio di markup é un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici) di un testo che, utilizzando convenzioni standardizzate, sono utilizzabili su più supporti

[37] https://en.wikipedia.org/wiki/SQL

- *Property* (*proprietà*): identifica una proprietà, un attributo o una relazione utilizzata per descrivere una risorsa. Il significato e le caratteristiche di questa componente vengono definite tramite RDF Schema;
- *Statement* (*espressione*): é l'elemento che descrive la risorsa ed é costituito da un soggetto (che rappresenta la Resource), un predicato (che esprime la Property) e da un oggetto (chiamato Value) che indica il valore della proprietà.

Lo statement viene rappresentato dalla figura 5 in cui possiamo immaginare il concetto di risorsa che possiede una proprietà con uno specifico valore. L'oggetto di uno statement RDF può essere a sua volta una risorsa, consentendo in questo modo di descrivere in maniera più approfondita il valore della proprietà.

Un problema di RDF consiste nel fatto che esso si limita a descrivere le proprietà di un oggetto senza fornire una struttura gerarchica che possa permettere di evidenziare le relazioni fra di esse. Per questo motivo é stato necessario trovare un linguaggio che descrivesse queste informazioni, chiamato RDF Schema. Attraverso RDF-S é possibile creare veri e propri vocabolari che descrivono classi e proprietà utilizzate all'interno di un'applicazione fornendo quindi una visione semantica tipica delle ontologie.

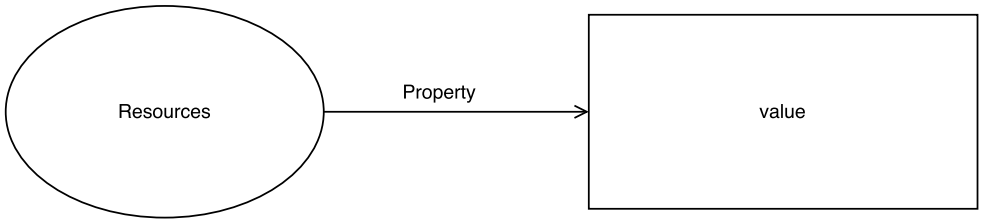


Figura 5: Rappresentazione modello di dati RDF

2.4.3. Representational State Transfer (REST)

Representational State Transfer (REST) é uno stile architetturale utilizzato all'interno del World Wide Web e fondato su un insieme di principi che descrivono come le risorse all'interno di una rete devono essere definite e indirizzate. Un applicazione definita REST-ful o REST-style deve essere:

- *Client-Server* ovvero uno stile architetturale che permette la suddivisione netta tra il server che implementa i servizi e gestisce la persistenza dei dati e il client, il cui compito é quello di fornire un interfaccia all'utente con cui poter effettuare richieste al server. Tramite questa suddivisione si rende il server molto più semplice migliorandone la scalabilità e si ottiene la portabilità del client su piattaforme eterogenee, permettendo a tutti e due i componenti di progredire indipendentemente l'uno dall'altro;

- *Stateless* questo principio definisce la comunicazione che deve essere effettuata tra client e server. Ogni richiesta dal client al server deve infatti contenere tutte le informazioni necessarie per essere interpretata, non deve cioé avere bisogno di informazioni memorizzate sul server. Lo stato della sessione é quindi mantenuto interamente sul client;
- *Cacheable/Non-Cacheable* tramite questo caratteristica si richiede che i dati all'interno di una risposta a una richiesta siano implicitamente o esplicitamente etichettati come “cacheable” e “non-cacheable”. Se una risposta é “cacheable” sarà possibile memorizzarla nella cache del client in modo da essere riutilizzata successivamente ad una richiesta equivalente. Questo permette di aumentare le performance limitando le richieste su cui bisogna attendere la risposta del server;
- *Layered* si intende la realizzazione di un’architettura gerarchica su diversi livelli partendo dal client e arrivando al server finale senza che i componenti di ciascun livello riescano a vedere questa stratificazione. Questo permette la scalabilità del server oltre che la possibilità di creare layer specifici per la gestione della sicurezza o per il bilanciamento del traffico di rete.

Un elemento fondamentale in questa architettura é l’URI Resources con cui si definisce il metodo di identificazione delle informazioni. Qualsiasi informazione viene infatti modellata come una risorsa che possiede un identificativo univoco URI; con risorsa intendiamo un documento, un immagine, un servizio o qualunque altro dato. Questo stile architetturale si basa sul protocollo HTTP con i suoi relativi metodi standard GET, POST, PUT, o DELETE. Inoltre, per le risposte da parte del server, vengono supportati tutti i maggior formati web quali ad esempio XML o JSON[13].

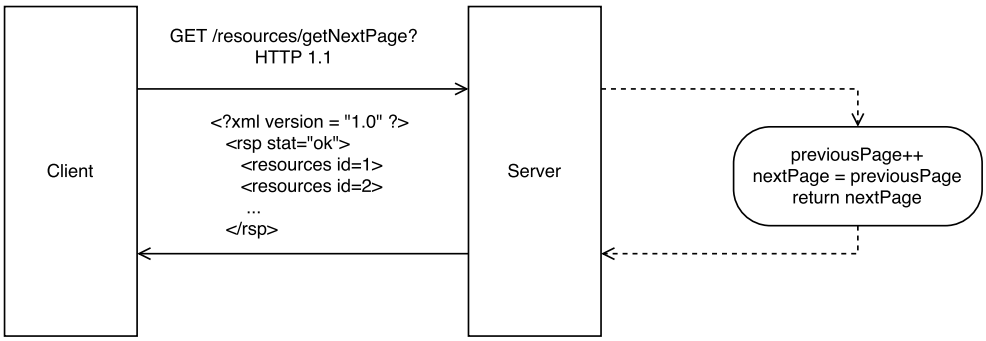


Figura 6: Esempio di richiesta-risposta tra client e server

2.4.4. Web Scraping

Il Web scraping é una tecnica utilizzata per estrarre informazioni non strutturate dalle pagine web sulla base di un procedimento automatico. Le pagine web sono documenti scritti in Hypertext Markup Language (HTML) e sono rappresentate da un albero strutturato chiamato Document Object Model (DOM). L'obiettivo dell'HTML é quello di specificare il formato di testo visualizzato dai browser Web.

Dal punto di vista operazionale, lo scraping web assomiglia a un copia e incolla manuale del contenuto delle pagine web, la differenza é che questo lavoro é fatto con un procedimento organizzato e automatico da programmi chiamati spider. Quando un spider segue i link all'interno di una pagina web sta effettuando delle richieste HTTP. Il risultato delle richieste effettuate sarà appunto una pagina HTML. A questo punto sarà necessario effettuare un analisi dell'albero HTML attraverso appositi linguaggi come XPATH. Questi linguaggi permettono di utilizzare un percorso (path) assoluto o relativo per identificare le parti del codice web in modo da selezionarle e di conseguenza estrarle, ad esempio, tramite l'utilizzo di espressioni regolari. Il web scraping si può utilizzare per svolgere operazioni come ad esempio confrontare prezzi online, monitorare dati meteorologici, rilevare modifiche in un sito internet.

3. DEFINIZIONE DEL PROBLEMA

Alla base della realizzazione di questa tesi vi é la volontà di esplorare un campo che si é rivelato altamente innovativo negli ultimi tempi, ovvero quello del “Sentiment Analysis” o “Opinion Mining”. L'intento é di osservare differenti domini di applicazione e focalizzare l’attenzione sul mondo dei social media e, in particolare, su Twitter. Tre aspetti fondamentali da considerare all’interno di questo ambito sono quindi la velocità con cui i dati si muovono, la loro forma, ovvero il gran numero di lingue in cui possono essere scritti i tweet e la mancanza di un dominio specifico (applicazioni mobile, turismo, sport, etc.).

Come é stato spiegato all’interno del paragrafo 2.2.1, molti dei lavori di ricerca più avanzati svolti fino ad oggi si sono concentrati sulla lingua inglese, sfruttando le principali tecniche di Sentiment Analysis su precisi domini di applicazione, come ad esempio: recensioni di prodotti [17] film [18], ristoranti e hotel [20] o su post di social media [14] . Tuttavia, l’uso di una sola lingua risulta altamente limitante, in quanto solo il 29,4% degli utenti di Internet scrive in inglese[38]. Per questo motivo é importante riuscire ad ampliare il raggio di applicazione sul numero più ampio possibile di lingue e senza un fuoco preciso in termini di dominio. Infatti, in generale all’interno di Twitter non vi é un argomento specifico anzi, come dimostra uno recente studio condotto da Brandwatch[39], si può vedere che gli argomenti più citati sono televisione, sport, musica, celebrità dimostrando quindi la variabilità e la diversità dei domini di utilizzo[40].

Il primo obiettivo di questo lavoro consiste quindi nella realizzazione di un **strumento che riesca a mantenere le stesse prestazioni offerte dai singoli strumenti per l’inglese anche su un numero più ampio di lingue**.

Attualmente, all’interno di grandi aziende é molto importante poter analizzare i dati provenienti dal Web 2.0 per poter scoprire informazione utili a effettuare decisioni di business importanti. In particolare i social media attraverso apposite API, come quelle di

[38] http://2www.internetworldstats.com/stats.htm

[39] https://www.brandwatch.com/

[40] http://bloggerjet.com/what-do-people-tweet-about/

Twitter citate nel paragrafo 2.3, mettono a disposizione un flusso (stream) d'informazioni che si muovono in tempo reale. Risulta importante riuscire a esaminare queste informazioni all'interno di un processo di analisi real-time di dati, che permette di utilizzare i dati nello stesso istante in cui vengono resi disponibili al sistema.

Il secondo obiettivo é la **creazione di uno servizio che renda possibile lo svolgimento del processo Sentiment Analysis multilingue su stream di dati in real-time**, come ad esempio lo stream di Twitter. Per tali motivi l'approccio più adeguato é quello Lexicon-based in quanto non richiede una fase di addestramento con periodi di esecuzione molto lunghi e risulta, al contrario dell'approccio Machine Learning, indipendente dal dominio di applicazione. Come spiegato nel paragrafo 2.2.3 questo metodo si basa su due concetti fondamentali: il primo, é la presenza di un lessico per ciascuna lingua contenente le parole e il relativo valore di polarità; il secondo, é la presenza di un metodo per la ricerca e la combinazione delle corrispondenze tra le parole del testo da analizzare e il lessico al fine di ottenere un valore di polarità totale.

In letteratura esistono alcuni lavori, ancora in via di sviluppo, legati all'ambito del Multilingual Sentiment Analysis [16] [4] , che utilizzano due differenti metodi associati all'approccio Lexicon-based. In un primo caso [9] viene sfruttata la procedura di traduzione automatica[41] in modo da poter passare da una qualsiasi lingua al dominio della lingua inglese, sfruttandone le risorse già presenti in letteratura. Questo metodo pur essendo il più immediato possiede delle controindicazioni, in quanto non é sempre specifico; infatti, delle volte, un termine della lingua di partenza non é perfettamente traducibile con uno della lingua di destinazione. Per esempio, si può osservare come in Svedese la generalizzazione "nonno" non esiste ma esiste solamente il concetto specifico "nonno paterno"/"nonno materno".

In un secondo approccio [5], si mira alla creazione di un lessico per ciascuna delle lingue di analisi sfruttando le caratteristiche semantiche delle singole lingue. Per fare questo vengono utilizzati metodi di propagazione semantica [22] che permettono di collegare un termine della lingua di partenza con la relativa rappresentazione concettuale della lingua di destinazione.

Il terzo obiettivo di questo lavoro é legato quindi al concetto utilizzato in entrambi i metodi appena citati , ovvero l**o sfruttamento delle risorse disponibili per la lingua inglese ai fini di muoversi su un dominio di Sentiment Analysis multilingue tramite algoritmo Lexicon-based**.

Durante il procedimento di Multilingual Sentiment Analysis attraverso l'approccio Lexicon-based si aspira a valutare alcune categorie di parole che portano alla modifica della polarità dell'intera frase. In particolare sono:

---

[41] Traduzione automatica : (dall'inglese Machine Translation) é un'area della linguistica computazionale e della scienza che studia la traduzione di testi da una lingua naturale ad un'altra mediante programmi informatici.

- *negazioni*, rappresentate da un gruppo di parole la cui presenza all'interno di una frase ne capovolge la polarità; ad esempio, la frase "Io non lo odio" non é negativa nonostante al suo interno abbia una parola con una polarità fortemente negativa come "odio";
- *booster*, sono termini particolari che intensificano in maniera positiva o negativa la parola che li segue, ad esempio, la frase "Il servizio qui é estremamente buono" avrà un connotato sicuramente più positivo rispetto alla frase "Il servizio qui é buono";
- *congiunzioni di contrasto*, ovvero parole che segnalano la presenza di due opinioni contrastanti all'interno della stessa frase; in particolare, l'opinione espressa dopo la congiunzione sarà dominante. Ad esempio nella frase "Il cibo qui é buon, ma il servizio é terribile" si può notare come la seconda parte della frase sia quella che vuole essere enfatizzata dall'autore;
- *punteggiatura*, la presenza della punteggiatura porta a enfatizzare l'opinione di una frase, ad esempio "Il cibo qui é buono!!!" sarà più espressivo di "Il cibo qui é buono";
- *maiuscole*, anch'esse come la punteggiatura portano all'intensificazione del concetto generale, ad esempio "Il cibo qui é BUONO!!!" avrà un valore fortemente più positivo rispetto a "Il cibo qui é buono".

Volendosi focalizzare sul dominio di applicazione dei social media é fondamentale valutare inoltre la presenza di due elementi che accentuano il valore dell'opinione: le emoticon e le emoji. Le emoticon sono rappresentazioni di possibili espressioni facciali realizzate attraverso l'utilizzo della punteggiatura, ad esempio " :-) " . Le emoji sono invece un'evoluzione del concetto di emoticon e sono utilizzate ormai in qualsiasi forma di comunicazione moderna, in quanto facilitano l'espressività. In particolare, le emoji sono simboli grafici, ideogrammi che non rappresentano solo espressioni facciali ma un insieme di elementi della vita quotidiana di ognuno di noi come ad esempio cibo e bevande, costruzioni, veicoli, celebrazioni, animali, piante o semplicemente emozioni attività o stati d'animo. Sono infatti molto importanti nell'ambito del Sentiment Analysis multilingue in quanto sono utilizzate in modo simile in tutte le lingue [27].

Dovendo creare uno strumento che funzioni real-time é importante studiare la frequenza con cui i dati vengono messi a disposizione per l'analisi. É necessario quindi effettuare il calcolo nel modo più veloce possibile. Per tale motivo, é essenziale mantenere all'interno dell'algoritmo solo le componenti che sono strettamente necessarie e che danno un apporto concreto alla valutazione finale della polarità.

Un esempio può essere dato dalla procedura di POS Tagging che é di solito uno standard all'interno degli algoritmi "Lexicon Based". Questa procedura, utilizzando gli algoritmi comuni citati nel paragrafo 2.1.2, all'interno dei social media difficilmente funziona, in quanto i testi sono scritti in un linguaggio informale oltre a contenere elementi come ad esempio emoticon, emoji, URL, username caratteristici dei soli social media [10].

Al giorno d'oggi si può notate la presenza di un gran numero di piattaforme che sfruttano tecnologie differenti ma il cui obiettivo é fornire uno stesso servizio all'utente . Un esempio intuitivo può essere dato dalle applicazioni mobile che vengono fornite per tutte le piattaforme disponibili sul mercato (Android, iOs, ecc.) e che utilizzano tecnologie differenti (Java, Objective C, ecc.). Durante la fase di creazione di un qualsiasi strumento é quindi importante considerare il concetto di interoperabilità in cui un unico elemento possa essere utilizzato nello stesso modo su piattaforme differenti. É perciò quarto obiettivo **rendere disponibile un'interfaccia che esponga il servizio di Multilingual Sentiment Analysis sfruttando i protocolli open del Web e mantenendo un elevato livello di sicurezza e di integrità**. Il servizio deve inoltre essere trasparente all'utente che deve poterlo utilizzare in maniera semplice e veloce senza effettuare procedure preliminari di messa a punto e solamente sottoponendo il testo da valutare e la lingua in considerazione.

Come é stato detto in precedenza, uno degli obiettivi della ricerca é quello di creare uno strumento ottimizzato per i social media e, quindi, in grado di funzionare su domini differenti legati ad esempio a recensioni di prodotti, di hotel o applicazioni senza degradare troppo nelle prestazioni. Per poter effettuare il calcolo delle metriche necessarie alla valutazione di questo tipo di strumento é però indispensabile avere a disposizione molteplici dataset provenienti da domini eterogenei per ciascuna lingua da esaminare. Il problema relativo a questi dataset é dato dal fatto che, oltre a dover contenere i testi da analizzare, devono essere affiancati da un valore di polarità proveniente da un essere umano e che rappresenti un valore reale osservato nel testo. Per questo motivo il quinto obiettivo é quello di **ottenere un dataset multilingue annotato manualmente per poter effettuare una procedura di valutazione che fornisca degli indici di qualità del lavoro**.

Riprendendo brevemente i concetti sopra citati, lo scopo di tutto il lavoro é la costruzione di un architettura software che permetta di svolgere una procedura di Sentiment Analysis che utilizzi il più alto numero di lingue possibile sfruttando un metodo Lexicon-based ottimizzato per poter funzionare all'interno di processi real-time e che sia consumabile da qualunque applicazione sottoforma di servizio Web. Ai requisiti funzionali descritti, si aggiungono requisiti basilari in termini di efficacia ed usabilità degli strumenti realizzati, oltre alla valutabilità del lavoro attraverso dataset annotati da esseri umani e provenienti da domini differenti.

4. SOLUZIONE DEL PROBLEMA

Tutta l'infrastruttura software realizzata parte dalla necessita di analizzare testi multilingue provenienti da domini differenti e di etichettarli con una rappresentazione dell'opinione data dal valore di polarità. É importante considerare la velocità di esecuzione in modo da permetterne il funzionamento per l'analisi in tempo reale. I macro componenti fondamentali sono quattro:

- *Propagation Algorithm*, il compito di questo componente é quello di propagare un lessico contenente termini in lingua inglese su un dominio multilingue in modo completamente automatico, tenendo conto delle regole sintattiche e semantiche utilizzate nella linguistica e considerando le caratteristiche dei testi prodotti sui social media;
- *Lexicon and Rules Based Algorithm*, il compito di questo componente é l'analisi di un testo in ingresso verificando le corrispondenze all'interno del lessico della lingua in considerazione e, successivamente, applicando regole di enfatizzazione, di tipo sintattico e grammaticale, utilizzate nel linguaggio comune;
- *Server Web*, il compito di questo componente é quello di rendere raggiungibile da remoto lo strumento di Sentiment Analysis in maniera completamente trasparente al client offrendo un interfaccia REST;

In figura 7 viene mostrata la struttura completa dell'architettura evidenziando le relazioni tra i singoli componenti e le risorse coinvolte. In particolare il propagation algorithm necessita di avere in ingresso alcuni elementi provenienti dalla lingua inglese indispensabili alla propagazione sulle altre lingue.

Queste risorse possono essere suddivise in tre categorie:

- *database lessicali,* come PWN, Global WordNet e SentiWordNet, questi tre elementi sono necessari per costruire il lessico principale costituito dalle parole fondamentali di ogni lingua;

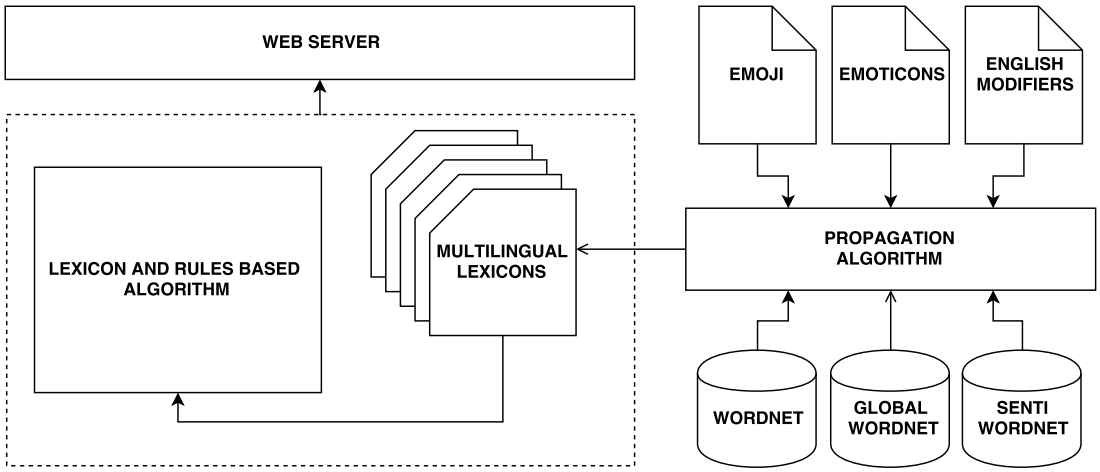


Figura 7: Architettura generale

- *emoji ed emoticon,* rappresentano la lista di emoji ed emoticon maggiormente utilizzate che deve essere costruita ed annotata con un valore di polarità per ogni simbolo, risulta molto importante perché sarà propagata all'interno di tutti i lessici in egual maniera;
- *english modifiers,* lista di parole provenienti dalla lingua inglese suddivise per categoria c con c ∈ {booster, negazioni, congiunzioni di contrasto} in questo caso l'algoritmo di propagazione si occupa della traduzione della lista di parole per tutte le lingue di destinazione.

Una volta eseguito l'algoritmo produce una lista di lessici, una per ogni lingua di destinazione, che vengono successivamente utilizzati dal "Lexicon and Rules Based Algorithm" per effettuare la procedura di Sentiment Analysis multilingue.

Infine l'ultima parte é costituita dal Web Server che espone un'interfaccia mediante la quale il Lexicon and Rules Based Alghoritm é accessibile da remoto. Dopo aver dato una visione fondamentale dei componenti principali dell'infrastruttura si può procedere con un'analisi più dettagliata di ciascun componente.

4.1. Propagation Algorithm

Questo macro componente é suddiviso in tre parti fondamentali i cui compiti rispettivamente sono:

- raccogliere i WordNet provenienti da Global WordNet e associarli con SentiWordNet ottenendo un lessico formato dalle coppie univoche <parola, polarità>;
- raccogliere le emoticon e le emoji in modo da ottenere anche in questo caso una lista di valori composta dalla coppia univoca <emoticon, polarità> e <emoji, polarità>;

- partendo da una lista di modifiers provenienti dalla lingua inglese associare la traduzione sulle altre lingue.

Il risultato di questo algoritmo determina un lessico per ognuna delle lingue di cui si é raccolto il WordNet contenente parole, emoji, emoticon, modifiers.

4.1.1. Estrazione WordNet multilingue

Nell'implementazione dell'algoritmo di propagazione il primo passo da compiere é la raccolta e la successiva organizzazione dei database lessicali per le singole lingue. Per fare questo é stato sfruttato il progetto Open Source Global WordNet che fornisce l'accesso a un ampia varietà di WordNet relativi a numerose lingue internazionali, tutti collegati alla lingua inglese tramite il Princeton WordNet (PWN); infatti, la struttura innovativa di PWN é rappresentata dalla sua possibilità di estensione da utilizzare come punto di partenza per introdurre il concetto di multilinguismo. I singoli WordNet provengono da molti progetti eterogenei e variano notevolmente in termini di dimensioni e precisione. L'approccio di propagazione del database Global WordNet si basa sul concetto di collegamento tra i synset della lingua inglese con i synset di altre lingue: in questo modo le relazioni semantiche rimangono le stesse per tutte le lingue. Parlando di ontologie, se due synset in un WordNet sono legati da una relazione, i due synset equivalenti in un altro WordNet saranno legati dalla stessa relazione.

I dati di ogni singolo WordNet sono forniti in formato Lexical Markup Framework (LMF) [7] che é uno standard ISO 639-2 per la codifica in linguaggio naturale di lessici leggibile facilmente in modo automatico. L'intenzione di LMF é quella di fornire un modello comune per la creazione e l'utilizzo delle risorse lessicali, per gestire lo scambio di dati e consentire la fusione di un gran numero di singole risorse per formare risorse globali.

Attraverso lo schema ER in figura 8, si evidenzia come l'entità Lexical Resource rappresenti il contenitore dell'intero lessico ed é in una relazione 1 a 1 con l'entità Global Information che contiene un insieme di informazioni generali che identificano il lessico. Ogni lessico monolingue é un'istanza dell'entità standard Lexicon che, a sua volta, rappresenta il contenitore per le parole della data lingua. La classe LexicalEntry rappresenta un'unità astratta di vocabolario: in prima approssimazione, può essere intesa come una parola. Le LexicalEntry rappresentano il legame tra Form (Lemma), classe astratta che rappresenta il modo in cui una parola é scritta (o parlata), e il relativo Sense, che ne rappresenta invece i significato. Ogni singolo Sense possiede un campo ID preceduto dal codice ISO 639-2 della lingua in cui é rappresentato, questo ID sarà lo stesso del corrispondente Synset di PWN. Nei Lessici basati su WordNet, la triade LexicalEntry-Lemma-Sense consente di gestire separatamente ogni synset possibile. L'implementazione di Wordnet in LMF consente inoltre di esprimere i rapporti semantici tra specifici significati piuttosto che tra soli synset. Nel caso

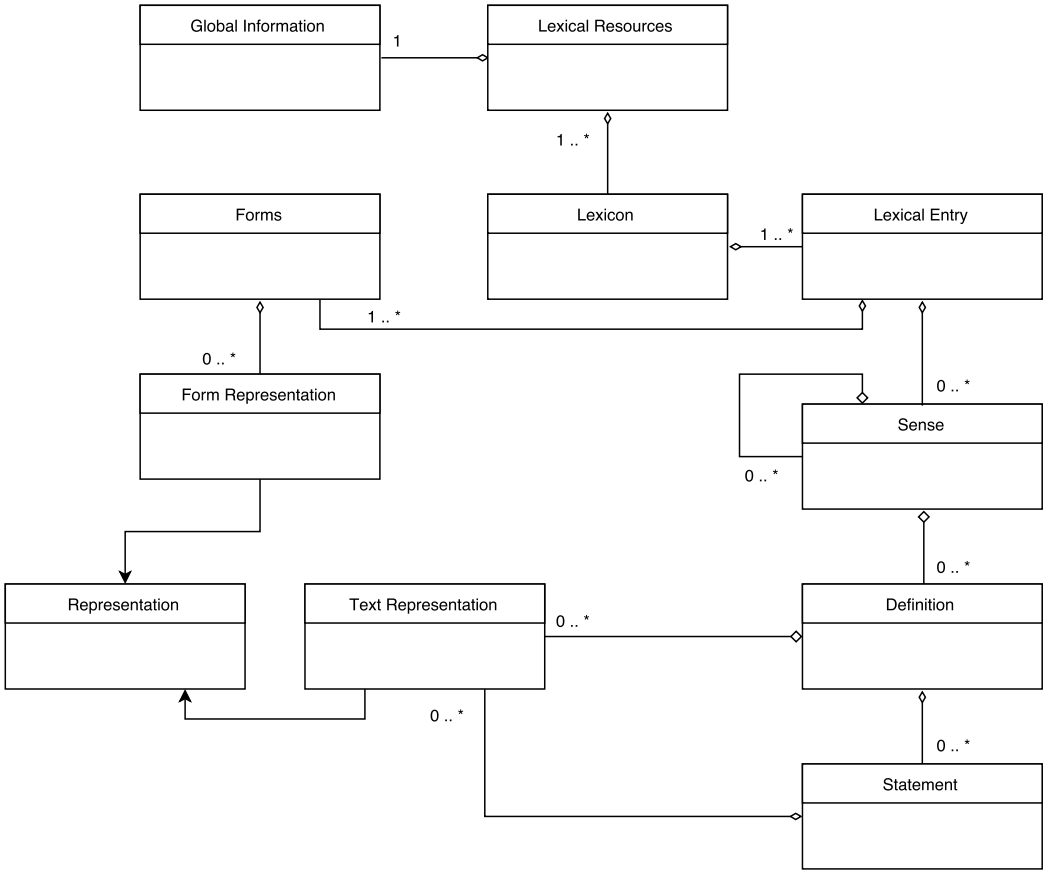


Figura 8: Schema ER Lexical Markup Framework [7]

in esame, ogni Wordnet rappresentato con LMF incorpora una singola lingua e le lingue in considerazione che sono state scelte per la propagazione sono rappresentate in tabella 1.

Dato ogni singolo lessico in formato LMF, é stato opportuno riorganizzare le LexicalEntry in un formato più immediato e veloce, utile nella procedura di Sentiment Analysis. Come si può vedere dalla rappresentazione UML in figura 9 ogni singola LexicalEntry per ogni Sense viene individuata da cinque campi:

- *idSynset,* che individua la corrispondenza con PWN ed é necessario al calcolo della polarità di ogni singola LexicalEntry;
- *writtenForm,* che individua la forma scritta di ogni LexicalEntry;
- *language,* che rappresenta la lingua;
- *polarity,* che si calcola successivamente e rappresenta la polarità associata alla data LexicalEntry;

Name	Language	# Synset
Princeton WN	Inglese	155,000
FinnWordNet	Finlandese	117,700
Thai Wordnet	Thailandese	73,593
DanNet	Danese	65,000
Spanish WN	Spagnolo	38,512
Arabic WN	Arabo	11,269
Greek Wordnet	Greco	18,049
Croatian Wordnet	Croato	23,120
MultiWordNet	Italiano	35,001
Open Dutch WordNet	Olandese	30,177
Norwegian Wordnet	Norvegese	3,671
OpenWN-PT	Portoghese	43,895
Romanian Wordnet	Rumeno	56,026
Lithuanian WordNet	Lituano	9,462
Slovak WordNet	Slovacco	18,507
sloWNet	Sloveno	42,583
WOLF	Francese	59,091

Tabella 1: WordNet provenienti da Global WordNet

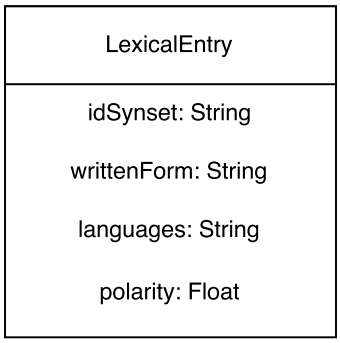


Figura 9: Class Diagram LexicalEntry

4.1.2. Polarità LexicalEntry

Una volta ottenute le singole LexicalEntry per tutte le lingue sopra citate il passo successivo é l'attribuzione del valore di polarità che ne rappresenti la connotazione Positiva, Negativa, Oggettiva. Per effettuare questa procedura é stato utilizzato il modello di SentiWordNet che é rappresentato in figura 10. Come si ha già avuto modo di spiegare, quest'ultimo si basa su PWN e associa ad ogni synset la sua connotazione di polarità con tre valori Positive, Negative, Objective tutti compresi tra 0.0 e 1.0. La somma dei tre valori é sempre 1.0, quindi ogni synset può avere un valore diverso da zero per ogni sentimento in quanto alcuni synset possono essere positivi, negativi o oggettivi a seconda del contesto in cui sono utilizzati. Come si vede dalla figura 10 ci sono due dimensioni principali:

1. *SO-polarity,* con questa dimensione si vuole rappresentare con un valore la soggettività di un testo, come nel verificare se il contenuto descriva una situazione o un evento, senza esprimere un parere positivo o un parere negativo su di esso. Ciò equivale a eseguire una classificazione binaria del testo nelle categorie soggettivo e oggettivo;
2. *PN-polarity,* con questa dimensione si vuole rappresentare la caratteristica di un testo soggettivo. Ciò equivale a eseguire una classificazione binaria del testo nelle categorie positivo e negativo;

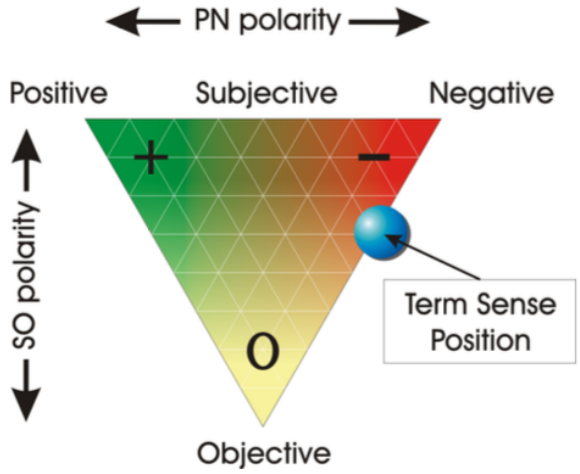


Figura 10: Modello SentiWordNet [2]

Chiaramente, con questo tipo di rappresentazione si può anche indicare la forza con cui una parola esprime positività o negatività infatti, un termine può essere debolmente o fortemente positivo o negativo. A questo punto si può associare ad ogni LexicalEntry i tre valori estratti da SentiWordNet utilizzando l'idSynset, per poi combinarli al fine di ottenere un valore unico.

La tripla associata ad ogni valore sarà combinata sottraendo il valore di positività con il valore di negatività:

$$Polarity = Pos - Neg$$

Come si può notare il valore di oggettività viene ignorato perché é intrinseco nel risultato in quanto, tanto più il valore di polarità finale sarà basso tanto più il significato sarà di avere una parola con alta connotazione di oggettività.

Successivamente bisogna procedere ad uniformare i valori di LexicalEntry il cui campo writtenForm risulta essere uguale; infatti, come é stato già ampiamente descritto, una stessa parola può appartenere a più synset. Bisognerà quindi unificare questi valori per ottenere un valore univoco di writtenForm associato alla sua polarità.
Per fare questo si calcola la media di tutti i valori di polarità che compaiono per quella determinata writtenForm. Avremo quindi:

$$polarity_{total} = \frac{1}{n} \sum_{j=1}^{n} polarity_j$$

Il valore finale é un valore che andrà da -1.0 a 1.0. Per il termine “good” ad esempio, SentiWordNet contiene 33 synset differenti. Sommando e dividendo ogni valore di positività e negatività otteniamo pos = 0.55, neg = 0.03 che dà un risultato finale di $polarity_{total}$ = 0.52. Il risultato finale di questa operazione ci permette di ottenere una lista di valori rappresentati dalla coppia univoca <writtenForm, polarity> per ognuna delle diciassette lingue sopra citate.

4.1.3. Emoji ed Emoticon

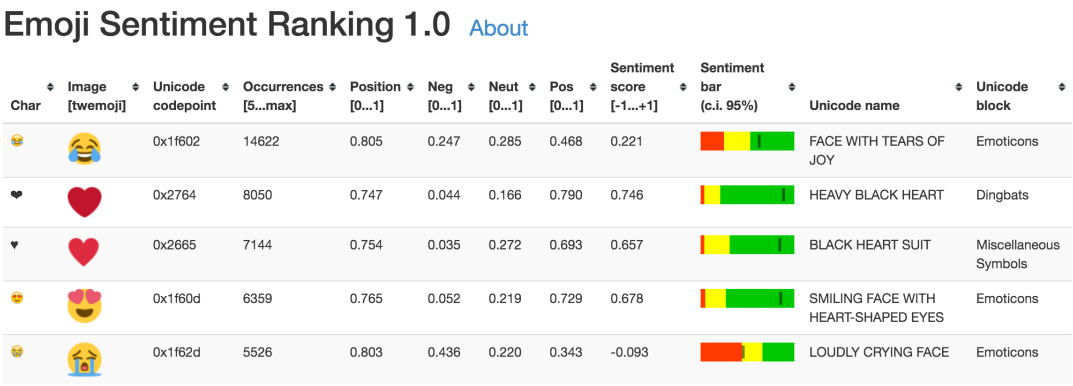
Ai fini della costruzione di un lessico da utilizzare nel processo di Sentiment Analysis risulta molto importante considerare due elementi che al giorno d’oggi vengono utilizzati dalla maggior parte delle persone nella scrittura di testi in digitale (sms, mail, chat, micropost). Tali elementi sono le emoji e le emoticon. Assumono particolare importanza all’interno di una valutazione multilingue in quanto rappresentano un alfabeto unico con cui vengono veicolati concetti di emozione o stati d’animo attraverso un linguaggio non verbale.

Per capirne l’importanza si può pensare al fatto che a Marzo 2015 Instagram, uno dei social network più famosi al mondo basato sulla condivisione di fotografie, ha rivelato che circa la metà dei testi al suo interno contengono emoji[42]. Inoltre, sono ormai contenuti su qualsiasi tastiera di dispositivi mobile e tablet basati su Android, iOS e Windows. A partire dal 2010 sono stati inseriti nella codifica Unicode versione 6.0 che rappresenta il principale standard nell’indicizzazione di caratteri.

[42] http://italianeography.com/instagram-analizza-uso-emoji-statistiche/

Per questo motivo sarà importante la costruzione di un lessico composto da queste due categorie di elementi e dal loro rispettivo valore di polarità. Nel caso delle emticon, questo problema risulta facilmente superabile in quanto la loro numerosità non é molto ampia, e questo ci permette di poter annotare manualmente ogni singola emoticon con un valore di polarità compreso tra (-1,1) . I due valori estremi sono stati scelti per poter rimanere coerenti con il lessico creato con SentiWordNet e in generale per essere quindi valutati nello stesso modo nel momento in cui si effettuerà la combinazione delle polarità contenute all'interno dei testi.

Per quanto riguarda le emoji il problema risulta essere invece un po più complesso in quanto la loro numerosità si aggira intorno alle migliaia di elementi. Per poter creare quindi un lessico che abbia le stesse sembianze di quello costruito per le LexicalEntry é stata utilizzata una risorsa esistente Emoji Sentiment Ranking 1.0 [27].



Emoji Sentiment Ranking 1.0 About

Char	Image [twemoji]	Unicode codepoint	Occurrences [5...max]	Position [0...1]	Neg [0...1]	Neut [0...1]	Pos [0...1]	Sentiment score [-1...+1]	Sentiment bar (c.i. 95%)	Unicode name	Unicode block
😂	😂	0x1f602	14622	0.805	0.247	0.285	0.468	0.221		FACE WITH TEARS OF JOY	Emoticons
❤	❤️	0x2764	8050	0.747	0.044	0.166	0.790	0.746		HEAVY BLACK HEART	Dingbats
♥	❤️	0x2665	7144	0.754	0.035	0.272	0.693	0.657		BLACK HEART SUIT	Miscellaneous Symbols
😍	😍	0x1f60d	6359	0.765	0.052	0.219	0.729	0.678		SMILING FACE WITH HEART-SHAPED EYES	Emoticons
😭	😭	0x1f62d	5526	0.803	0.436	0.220	0.343	-0.093		LOUDLY CRYING FACE	Emoticons

Figura 11: Esempio di risultati provenienti da Emoji Sentiment Ranking 1.0 [27]

L'elemento interessante di questa risorsa é il modo in cui sono state trovate e annotate le emoji. Infatti, per l'attribuzione del valore di polarità il primo passo é stato quello di raccogliere e annotare un dataset di tweet con un valore discreto {-1, 0, +1} assegnato da un gruppo di 83 annotatori madrelingua. Successivamente, ad ogni Emoji é stato assegnato il corrispondente valore di annotazione del tweet in cui era contenuta in modo da formare una distribuzione di probabilità discreta (p−, p0, p+). Il valore corrispondente alla media della distribuzione rappresenta il valore finale di polarità. Tramite questa risorsa é stato possibile ottenere un lessico formato da una lista di valori <emoji, polarity> nello stesso formato del lessico composto con le LexicalEntry.

4.1.4. Multilingual Modifiers

L'ultima parte di ciascun lessico multilingue é composta dai Modifiers, ovvero elementi che permettono di invertire o alterare la polarità delle parole che li segue. É importante isolare per ogni lingua questo gruppo di parole suddividendole nelle tre categorie: booster, negazioni e congiunzioni di contrasto. Anche in questo caso per poter ottenere queste risorse per ciascuna delle diciassette lingue raccolte é stato necessario partire dalla lingua inglese di cui si possedevano già questi elementi per poi propagarli sulle diverse lingue.

In questo caso il metodo utilizzato per la propagazione é la traduzione attraverso le API di Google Translate, mediante un componente che, per ogni termine in inglese ricevuto in ingresso, interrogando le API determina la traduzione nella lingua di destinazione.

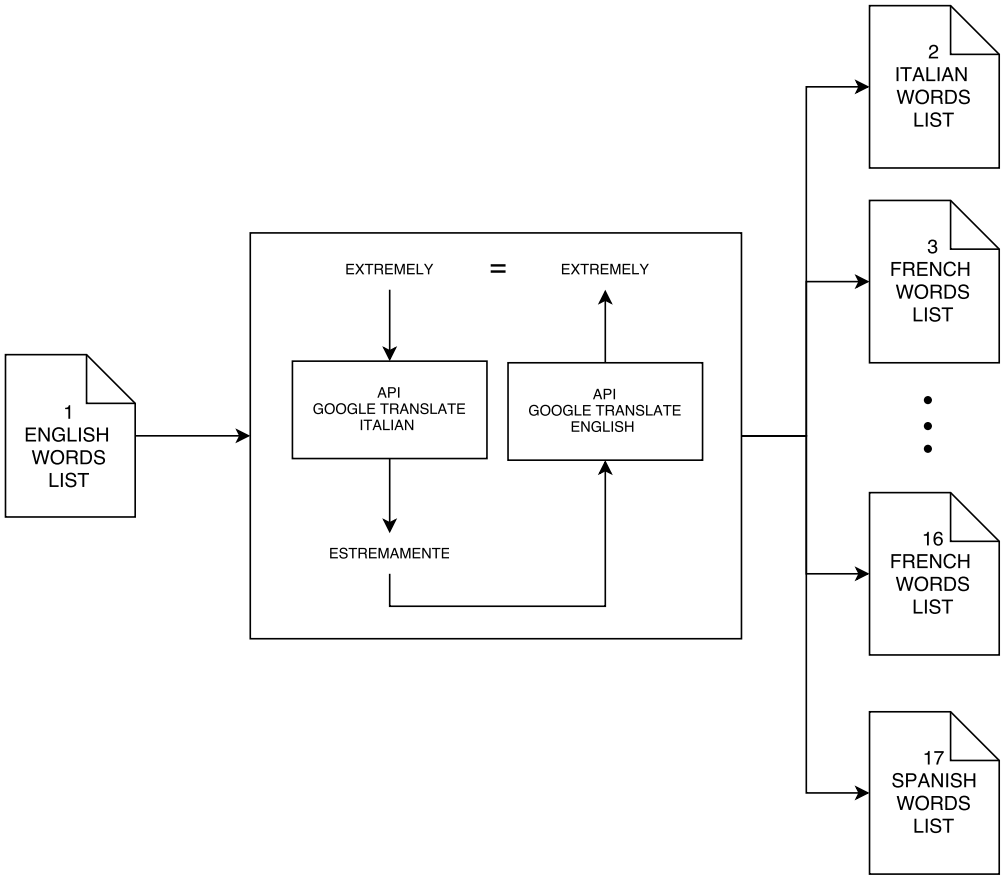


Figura 12: Metodo di traduzione english modifiers

Come si può vedere in figura 12 il componente in questione utilizza un metodo molto intuitivo. Si parte prendendo la parola in ingresso e se ne traduce il valore nella lingua di destinazione. Successivamente il risultato ottenuto viene riconvertito nella lingua sorgente e

confrontato con il termine di partenza. Solo nel caso in cui i due elementi sono uguali allora il processo di traduzione viene considerato corretto altrimenti no. Questo tipo di procedimento ci permette di effettuare un filtraggio che porta ad eliminare gli elementi che vengono tradotti in maniera errata.

Ad esempio, se si prende la parola “extremely” appartenente alla categoria dei booster il componente effettuerà una traduzione (in questo caso) verso l’italiano ottenendo la parola “estremamente” e successivamente ritraducendo verso la lingua di partenza (cioè l’inglese) riottenendo la parola “extremely”. Nel caso la parola ottenuta sia uguale alla parola di partenza si procede ad aggiungerla alla lista di modifiers per la destinazione, altrimenti verrà scartata.

4.2. Lexicon and Rules Based Algorithm

A seguito della costruzione dei lessici multilingue, ci si é occupati della progettazione di un componente che sia in grado, dato un testo di input, di etichettare con un valore discreto p compreso in P = {Positivo, Negativo, Neutrale} tale testo sfruttando le caratteristiche di polarità contenute all’interno di ogni singolo lessico.

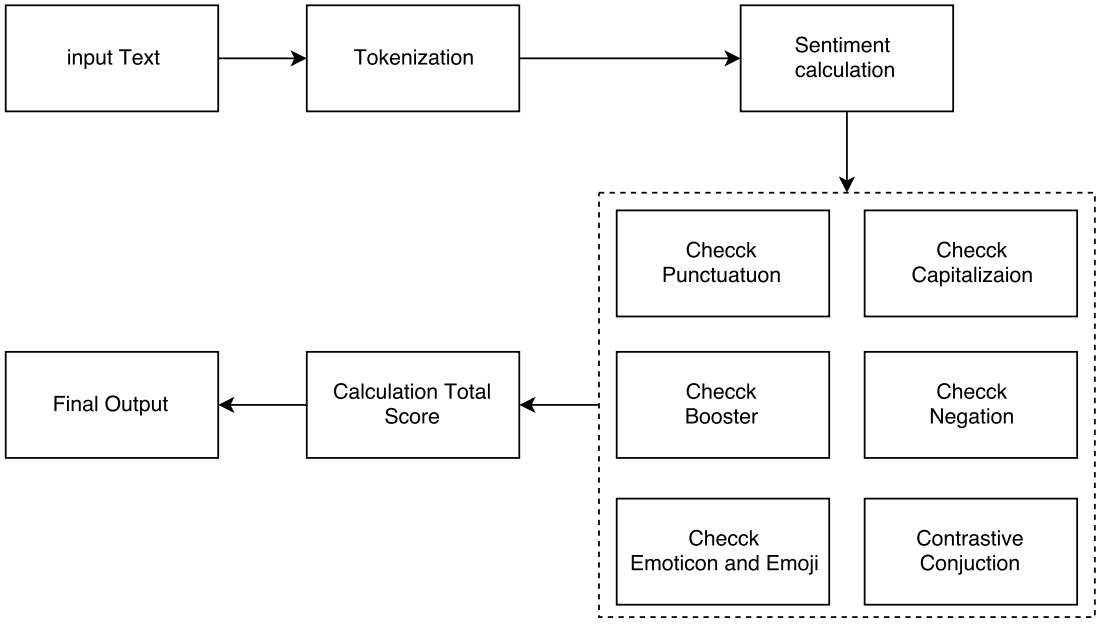


Figura 13: Struttura Lexicon Based and Rules Algorithm

La figura 13 descrive ad alto livello il funzionamento dell’algoritmo, la cui descrizione nel dettaglio verrà trattata in seguito.
Le funzioni principali svolte sono:

- scomporre in token il testo fornito in input all'algoritmo. I token rappresentano le singole entità indipendenti che costituiscono la frase, si possono intendere banalmente come parole, ma in realtà possono rappresentare la punteggiatura, le emoticon, gli url o, nel caso dei social Network, gli hashtag;
- assegnare ad ogni token un valore di polarità recuperato dal lessico relativo alla lingua fornita in input insieme al testo;
- controllare la presenza della punteggiatura che funge da intensificatore di polarità mantenendo invariata la semantica;
- valutare l'utilizzo delle maiuscole che, anche in questo caso, hanno funzione di intensificatore;
- considerare la presenza di congiunzioni di contrasto come ad esempio "ma" che accentuano la polarità dell'intera frase che le segue;
- evidenziare la presenza delle forme di negazione che portano ad avere un capovolgimento della polarità di una singola parola;
- verificare la presenza di aggettivi o avverbi che modifichino la polarità della parola che li segue in maniera positiva o negativa, questo tipo di avverbi sarà chiamato booster.
- calcolare e aggregare i valori di ogni singolo token;
- convertire il valore finale numerico, tramite un valore soglia, in un'etichetta $p \in P = \{\text{Positivo, Negativo, Neutrale}\}$;

4.2.1. Tokenization

Il processo di estrazione di token, che consiste nel dividere una stringa nelle sue varie componenti indipendenti, é fondamentale per tutte le attività di Natural Language Processing. Ci sono molti modi in cui svolgere questo task e nessuno é corretto a priori in quanto la massimizzazione delle prestazioni é data dal contesto di applicazione. Nel processo di Sentiment Analysis secondo un metodo Lexicon and Rules Based questa operazione diventa ancora più importante poiché dalla corretta individuazione dei singoli token dipende il recupero della polarità corrispondente dal lessico.

Occorre evidenziare come questo processo ha subito delle variazioni nel corso del tempo e, con l'avvento dell'uso delle emoticon e successivamente delle emoji, ha portato a dare a particolari sequenze di segni di punteggiatura (come ad esempio ":-(") significati specifici, soprattutto nel campo dei Social Media. Qui di seguito possiamo notare come, ad esempio nel caso di Twitter, sia altamente influente tenere conto di alcuni aspetti.

> @CheffyPaul Tutti pronti per la #vigilia con #MasterChefIt??? MANCA POCO!!:) http://www.sky.it

Svolgendo una procedura di Tokenization basata solamente sugli spazi vuoti otterremo una lista di token (suddivisi dalla virgola) così composta:

> @CheffyPaul,Tutti, pronti, per,la,#vigilia,con,#MasterChefIt???, MANCA,POCO!!:) , http://www.sky.it

É considerevole osservare come ad esempio il token [POCO!!:)], che in questo caso verrebbe considerato come singolo token, contenga invece tre parti molto importanti analizzate singolarmente: la parola [POCO], l'emoticon [:-)] e la punteggiarura [!!].

L'esempio, inoltre, é utile per mostrare altri aspetti su cui é necessario focalizzarsi per poter svolgere una procedura di tokenization che massimizzi l'utilizzo del lessico precedentemente costruito. Si può vedere come sia comune all'interno di Twitter ma in generale nella maggior parte dei social network l'utilizzo degli hashtag (#). Per poter attribuire un valore di polarità agli hashtag utilizzati all'interno dei micropost é necessario che questi siano presi senza il carattere "#" in modo da poter trovare una corrispondenza all'interno del lessico.

Un'altra particolare caratteristica di Twitter é l'utilizzo della "@" per rappresentare gli username degli utenti coinvolti nel post; é fondamentale riuscire ad estrarli correttamente per poi poter svolgere procedure di analisi successive, che non sono oggetto di questo lavoro, come ad esempio NER[43] (Named Entity Relationship).

Come é stato già sottolineato, un altro punto a cui rivolgere l'attenzione sono le emoticon e le emoji, molto comuni all'interno dei social media, le quali svolgono un ruolo fondamentale nel valore totale di polarità del testo. Risulta dunque necessario poterle estrarre. Attraverso l'utilizzo di apposite espressioni regolari é possibile catturare la maggior parte delle emoticon testuali e delle emoji che circolano in Twitter. Una criticità legata alle emoticon é la gestione della punteggiatura; infatti il processo di tokenization richiede di eliminare dal testo la punteggiatura come ad esempio [",", ".", ":", ";"], tuttavia i singoli segni di punteggiatura devono essere mantenuti nel caso in cui compongano un'emoticon [" :) ", ": ("].

Url o elementi HTML sono fattori molto comuni all'interno dei testi in digitale. Nonostante la loro presenza non sia determinante ai fini del processo di Sentiment Analysis seguito all'interno di questo lavoro é utile poterli isolare correttamente, in modo da non causare problemi nelle successive procedure di analisi. É sostanziale, inoltre poter preservare

[43] Named Entity Recognition : Tale termine é definito come la fase in cui si cerca di classificare gli elementi atomici di un testo in categorie predefinite, quali i nomi delle persone o delle organizzazioni, posizioni, espressioni dei periodi, quantità, valori, monetari, percentuali, etc…

le maiuscole perché successivamente aiuteranno a dare un valore di intensificazione alle singole parole.

Per svolgere correttamente tutte queste procedure é necessario propagare il testo all'interno di opportuni filtri che ne estraggono correttamente le componenti fondamentali. Riprendendo l'esempio sopra citato, una corretta procedura di Tokenization all'interno di Twitter porterà a un output finale dato da:

@CheffyPaul, Tutti, pronti, per, la, vigilia, con, MasterChefIt, ???,
MANCA, POCO, !!, ,:), http://www.sky.it

Figura 14: Sequenza attività tokenization

4.2.2. Attribuzione Polarità

Il risultato dell'algoritmo di tokenization consiste in una lista di token che rappresentato le entità fondamentali del testo in input come parole, punteggiatura, hashtag, emoticon e emoji. Il processo di attribuzione della polarità si occupa di assegnare ad ogni entità un valore numerico estratto dai lessici precedentemente costruiti ed effettuando un'aggregazione di tali valori. Per effettuare questo procedimento é fondamentale ricevere, insieme al testo in input, l'indicazione della lingua in cui é espresso.

Per prima cosa vengono controllati gli elementi che non hanno bisogno di un'analisi sintattica, ovvero emoji, emoticon e hashtag. Ne viene cercata una corrispondenza all'interno del lessico e, nel caso ci sia, vengono prelevati e assegnati i valori di polarità ad essi associati. Successivamente é possibile procedere all'analisi del testo vero e proprio.

Il primo passo verifica se tutte le parole sono scritte in maiuscolo, l'enfatizzazione pari a 0.733 verrà assegnata solo nel caso in cui la condizione sia falsa e quindi le maiuscole siano usate solo in alcune parti del testo. I valori costanti che permettono di invertire, intensificare o diminuire la polarità in particolari situazioni (come la presenza di maiuscole, punteggiatura, booster) sono stati presi da una risorsa per l'inglese chiamata Vader in cui venivano calcolati empiricamente valutando la modifica media apportata nel punteggio di polarità all'interno di un elenco di Tweet [18].

Successivamente si procede con l'analisi del testo principale focalizzandosi, oltre che sulla ricerca delle corrispondenze delle parole all'interno del lessico, anche sulla verifica delle regole sintattiche necessarie ad effettuare una procedura di intensificazione o di inversione della polarità. Ad ogni parola viene in primo luogo assegnato il valore v corrispondente (se contenuto nel lessico) quindi si procede con il controllo della presenza di negazioni e/o

booster ispezionando il bi-gram e il tri-gram[44]. Nel caso in cui venga verificata la presenza di una negazione si moltiplica v con un valore fissato costante -0.74, invertendone cosi la polarità.
Stessa cosa verrà fatta anche per i booster che sono suddivisi in due famiglie:

- *booster positivi,* che hanno associato un valore di incremento di 0.293;
- *booster negativi,* con un valore di -0.293.

Anche in questo caso vengono cercate corrispondenze nell'analisi del bi-gram e del tri-gram e, una volta verificata la presenza, viene controllato il segno della polarità della parola presa in considerazione; se é negativa, il valore del booster viene invertito, altrimenti viene lasciato cosi com'é. Solo successivamente il valore viene sommato a v. Sia nel caso delle negazioni che dei booster si parte dal presupposto che la presenza di due negazione porti ad una affermazione positiva.

Una volta ottenuto l'elenco di polarità delle parole del testo si esegue un controllo sulla presenza delle congiunzioni di contrasto e, nel caso in cui ce ne siano, vengono amplificati tutti i valori successivi a tali parole e decrementati quelli precedenti. In seguito, si effettua la somma di tutti i valori e si controlla la presenza della punteggiatura introducendo un incremento di 0.291 nel caso si trovino "!" e di 0.96 nel caso di "?".Dopo una procedura di normalizzazione tra -1 e 1 otterremo quindi il valore finale di polarità.

Per poter effettuare in maniera più agevole la procedura di Sentiment Analysis si é deciso di convertire il valore numerico, trovato con il calcolo sopra descritto, avvalendosi di un valore discreto tra {"positive", "neutral", "negative"}. Per fare questo sono state fissate delle soglie con cui poter convertire il valore di polarità totale in modo da poter classificare nella maniera corretta le tre categorie.

Si é scelta una soglia intuitiva fissata a 0 che assegna l'etichetta "positive" a tutti i valori maggiori di zero, l'etichetta "negative" a tutti i valori minori di zero e l'etichetta "neutral" ai valori esattamente uguali a zero. Questa scelta consente di ottenere delle performance, in termini di precision e recall, che saranno maggiori nella ricerca di "positive" e "negative" con un corrispettivo degrado dei "neutral". Per poter aumentare le prestazioni dei "neutral" si è scelta quindi un'altra soglia compresa tra -0.25 e 0.25 corrispondente al valore massimo di F1 del classificatore che enfatizza le prestazioni sui "neutral". Questa scelta porta però a perdere di qualità su "positive" e "negative".

4.3. Rest Web Server

L'ultima parte dell'architettura principale é rappresentata dal Rest Web Server, il cui compito fondamentale é quello di esporre delle API accessibili da remoto in qualunque

[44] n-gram: si intende una sequenza di n token adiacenti nella frase generale

momento e da qualunque dispositivo sfruttando il protocollo di comunicazione HTTP. Nella creazione di questo servizio si é scelto di accettare richieste POST le cui caratteristiche fondamentali sono quelle di:

- non essere mai inserite nella cache del browser;
- non rimanere nella cronologia del browser;
- non avere restrizioni sulla lunghezza dei dati.

La richiesta deve contenere due parametri obbligatori:

- *text,* che rappresenta il testo da analizzare;
- *language,* che rappresenta la lingua in cui é espresso il testo;

La risposta del Web Server é inviata in formato JSON (JavaScript Object Notation) che risulta essere molto intuitivo da leggere e scrivere sia per gli esseri umani che per le macchine. Utilizza una serie di convenzioni che sono comuni ai più famosi linguaggi di programmazione come ad esempio C, C++, C#, Java, JavaScript, Perl, Python, e molti altri. La struttura del messaggio di risposta ha la seguente forma:

```
{
    "timestamp": “Data e tempo di generazione della risposta",
    "time": “Tempo impiegato per la generazione della risposta",
    "lang": “Lingua utilizzata per analizzare il testo",
    "sentiment": {
        "score": “polarità del testo compreso -1.0 to 1.0",
        "type": “etichetta ‘positive’ o ‘neutral’ o ‘negative’ ”
    }
}
```

5. ESPERIENZA IMPLEMENTATIVA

In questo capitolo vengono discusse le scelte implementative adottate nella progettazione e nella creazione dell'intera architettura. Come nel capitolo precedente, si giustificano le scelte fatte partendo da una visione generale dell'applicazione, fino ad arrivare alla presentazione dei framework utilizzati.

La prima scelta da fare in fase di progettazione é il *linguaggio di programmazione* da utilizzare per l'implementazione dell'intera architettura. La scelta é caduta su Python, in quanto presenta una molteplicità di vantaggi in tutti i campi di applicazione in cui si muove il progetto.

In primo luogo, Python risulta essere molto utile sia come linguaggio di scripting che come linguaggio orientato agli oggetti. É caratterizzato da una sintassi molto leggibile ed intuitiva e questo ne facilità l'utilizzo e la fase di debugging. Inoltre, nell'architettura presentata é necessario avere una strumento che sia il più possibile ottimizzato per il processo di data analysis, dovendo interagire con dataset provenienti da domini differenti. Python possiede infatti delle librerie integrate per operare con differenti formati di dati come ad esempio JSON, Xml, csv. Sono inoltre presenti delle estensioni molto interessanti per la misurazione di metriche applicate ai dati come Precision, Recall, Accuracy, F1.

Un altro punto a favore della scelta di Python é la grande propensione al mondo del Natural Language Processing, con molte librerie che aiutano lo svolgimento di questo task come ad esempio la famosa NLTK[45]. Oltre a queste caratteristiche che lo rendono particolarmente adatto al progetto, Python é anche un linguaggio di programmazione orientato agli oggetti, il che ha permesso la progettazione di un sistema che sfruttasse le caratteristiche di modularità fornite da questo paradigma di programmazione. Inoltre, Python possiede un ampia scelta di estensioni per la realizzazione di applicazioni web e, per questo motivo, é stato utilissimo anche nell'implementazione del *Rest Web Server* necessario per accedere al servizio di Sentiment Analysis da remoto.

[45] http://www.nltk.org/

5.1. Architettura generale

Nell'architettura sono presenti cinque *package* che implementano i componenti fondamentali e sono:

- *Propagation*, al suo interno si trovano le classi utilizzate per la creazione dei lessici multilingue partendo dai formati LMF di Global WordNet. Il risultato di questo processo é un file in formato txt contenete la lista di valori <wirttenForm, polarity>;
- *Sentiment_Tool*, contiene le classi necessarie all'implementazione del Lexicon and Rules Based Algorithm. Riceve in input un testo da analizzare e la lingua desiderata per l'analisi;
- *Creating_Corpus_Reviews*, in questo package si trovano le classi necessarie allo scraping di TripAdvisor e Google Play sfruttando il framework Python Scrapy;
- *Web_API*, sono presenti le classi utili all'implementazione del Rest Web Server utilizzando due framework Python: Flask e Flask-Restful;
- *Services,* implementa le interfacce ai servizi esterni come ad esempio le Api di Google Translate e le classi necessarie al calcolo delle metriche per la fase di valutazione dei dataset.

5.2. Propagation Algorithm

Il primo elemento di cui devono occuparsi le classi all'interno di questo *package* é la lettura e l'interpretazione di un file in formato LMF che identifica il singolo WordNet di ciascuna lingua e la mappatura su un modello basato su due classi chiamate Lemma e Synset. Il formato LMF (come si vede in figura 15) é molto simile al XML. Come é stato visto nella rappresentazione tramite il diagramma ER in figura 8 a pagina 44, ogni singola parola all'interno di un WordNet é rappresentata come una *LexicalEntry* che possiede il proprio *ID* univoco e che al suo interno é composta da:

- *Lemma***,** rappresentato dalla forma scritta *writtenForm* e dal valore di *PartOfSpeech*;
- Sense, costituito da un ID univoco e dal valore di synset corrispondente con PWN; inoltre, si trova la presenza di un prefisso che identifica la lingua.

Le classi presenti all'interno del package *Propagation* sono quattro e sono *PropagationAlgorithm*, *XmlReader*, *Lemma* e *Synset*. Le ultime due classi *Lemma* e *Synset,* rappresentate in figura 16, hanno il compito di mappare il modello LMF con un modello ad oggetti. La classe *Lemma* possiede gli attributi *writtenForm*, *idLemma* che corrisponde all'*ID* del *LexicalEntry*, *partOfSpeech* e un array di S*ynset* che ne rappresentano i *Sense* del formato LMF.

```
<LexicalEntry id ='w1533128'>
      <Lemma writtenForm='buono' partOfSpeech='a'/>
      <Sense id='w1533128_01983162-a' synset='ita-10-01983162-a'/>
      <Sense id='w1533128_01800349-a' synset='ita-10-01800349-a'/>
      <Sense id='w1533128_00106020-a' synset='ita-10-00106020-a'/>
      <Sense id='w1533128_01123148-a' synset='ita-10-01123148-a'/>
      <Sense id='w1533128_01372049-a' synset='ita-10-01372049-a'/>
      <Sense id='w1533128_00633410-a' synset='ita-10-00633410-a'/>
      <Sense id='w1533128_00631391-a' synset='ita-10-00631391-a'/>
      <Sense id='w1533128_01129977-a' synset='ita-10-01129977-a'/>
</LexicalEntry>
```

Figura 15: LexicalEntry in formato LMF

Con la classe *Synset* si rappresenta il singolo *Sense,* a cui vengono impostati gli attributi *neg_polarity*, *pos_polarity*, *obj_polarity* tramite l'estrazione di tali valori da SentiWordNet. L'attributo *label* é il valore discreto che viene assegnato al *Synset* e che é compreso tra {"positive", "negative", "neutral" }. L'attributo *idSynset* infine, corrisponde al valore synset presente nel formato LMF.

Il modello di dati é utilizzato dalla classe *PropagationAlgorithm* per svolgere la procedura competa di costruzione del lessico. Il punto di partenza di tutto il package é la funzione *CreateLexicon()* che riceve come parametro *path,* che indica il percorso in cui si trova il WordNet da analizzare. Questo metodo ha il compito di chiamare il servizio *XmlReader,* che al suo interno possiede la funzione extractLexicalEntry(), che si occupa di fare da interfaccia con il file LMF permettendone la lettura e la conversione in un'array che può quindi essere letto in maniera più agevole all'interno della nostra classe.

Successivamente, all'interno della funzione *createAllLemmaObj()* si procede alla fase di mappatura dell'array estratto dal file LMF e la relativa associazione sulle due classi *Lemma* e *Synset*. All'interno di questa funzione si procede all'eliminazione della parte relativa alla lingua contenuta nell'*idSynset* e, utilizzando la libreria NLTK contenente l'estensione per l'utilizzo di SentiWordNet, vengono recuperati i tre valori associati a ciascun *idSynset*.

Come si può notare nella rappresentazione del modello di dati utilizzato si é voluto mantenere tutti i valori presenti all'interno del WordNet come ad esempio il *partOfSpeech,* anche se in realtà non saranno inseriti nel lessico finale. Tale scelta é stata fatta in previsione di una futura estensione del lessico che potrebbe portare ad avere bisogno di alcuni valori come ad esempio appunto il *partOfSpeech*.

L'ultimo passo per la creazione del lessico si occupa di prendere la lista di *Lemma,* precedentemente ottenuta, e di convertirla in una rappresentazione finale che porti ad avere una lista di valori formata da una serie di coppie <*writtenForm, polarity*>. Per fare questo si é partiti dal *Lemma* prendendo l'attributo *writtenForm* e si sono creati tanti valori quanti

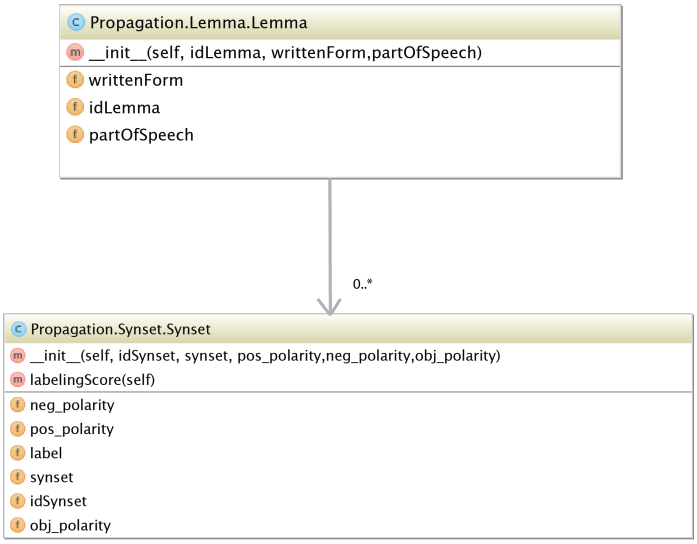


Figura 16: Class Diagram Lemma Synset

sono i *Synset* associati al *Lemma* ottenendo così una lista di coppia <*writtenForm, polarity*>. Il campo polarity é stato ottenuto sottraendo il valore di *neg_polarity* al valore di *pos_polarity* e mantenendo uguale *writtenForm*.

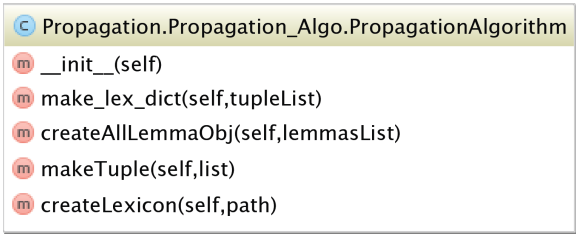


Figura 17: Class Diagram Propagation Algorithm

Successivamente all'interno della funzione *makeTuple()* si prendono tutti i valori con lo stesso writtenForm e se ne calcola la media ottenendo quindi il valore finale per ciascun Lemma. La funzione *make_lex_dict()* si occupa di creare il file finale con il path contenente il codice ISO 639-2 che ne rappresenta la lingua.

5.3. Lexicon and Rules Based Algorithm

L'implementazione del *Lexicon and Rules Based Algorithm* viene effettuata all'interno del package *Sentiment_Tool,* il quale sfrutta le risorse ottenute dalla propagazione dei Global WordNet. Queste risorse sono una raccolta di 17 lessici, formati da un file principale contenente le parole fondamentali di ciascuna lingua e le emoticon, un file contenente le

emoji e un file contenente i modifiers, tutti con relativi valori di polarità.
Partendo da queste risorse si può illustrare la struttura interna del package in considerazione elencando le tre classi fondamentali:

- *SentimentAnalysis,* il cui compito é quello di prendere il testo di input e la lingua e restituire come output un valore di polarità;
- *LoadLexicon,* che ha il ruolo di caricare in memoria il lessico corrispondente alla lingua in input. Viene utilizzata all'interno della classe *SentimentAnalysis;*
- *Tokenizer,* che si occupa di dividere il testo di input in entità autonome pronte ad essere analizzate. Anche in questo caso é la classe principale *SentimentAnalysis* ad utilizzarla.

5.3.1. Caricamento Lessico

La prima classe ad essere analizzata é *LoadLexicon,* che al momento dell'inizializzazione riceve come parametro la lingua di cui si vuole caricare il lessico. Come é stato esplicitato nei paragrafi precedenti, ciascun lessico é composto da tre parti, una con le parole fondamentali e le emoticon, una con le emoji e una con i modifiers. Il path di ciascuno di questi file viene costruito con l'utilizzo del codice ISO 639-2 della lingua e salvato all'interno delle variabili *path_modifier*, *emoji_path*, *path_lexicon* al momento dell'inizializzazione dell'oggetto.

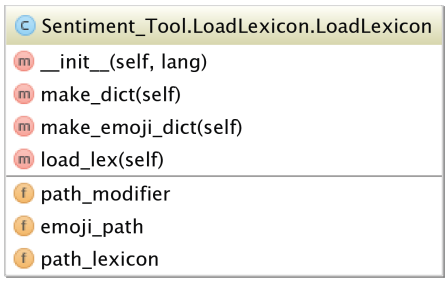


Figura 18: Class Diagram LoadLexicon

Per l'utilizzo dei lessici contenuti all'interno di questi file per la fase di Sentiment Analysis si é scelto di utilizzare una particolare struttura dati messa a disposizione da *Python* chiamata *dizionario*. I *dizionari* possono essere pensati come un insieme non ordinato di coppie *{chiave : valore}* in cui la chiave é unica e immutabile ed é utilizzata per indicizzare gli elementi. In questo caso la chiave é rappresentata dalla parola, dalla emoji o dalla emoticon, mentre il valore é la polarità.

Le funzioni *make_dict()* e *make_emoji_dict()* hanno il compito di leggere i file e creare i dizionari che conterranno rispettivamente parole, emoticon, modifiers ed emoji.

5.3.2. Tokenization

Per la procedura di *Tokenization* vengono utilizzate le espressioni regolari o *RegEx* attraverso un apposita estensione fornita da Python chiamata *re*. All'interno di questa classe, rappresentata in figura 19, si trova una prima funzione molto importante chiamata *tokenize()* che, applicando una concatenazione di espressioni regolari, ha il compito di estrarre le entità fondamentali contenute nel testo individuando in particolare alcuni pattern utilizzati solo all'interno di Twitter. Nello specifico, vengono cercate parole con e senza apostrofo, emoticon, numeri di telefono, url, html tag**,** Twitter username e Twitter hashtag. Una volta individuati gli hashtag, su di essi viene effettuata una semplicissima procedura di eliminazione del carattere "#" in modo che il termine corrispondente possa essere cercato all'interno del lessico..

Successivamente si trova una funzione chiamata *extract_emoji_token()* che ha il compito di applicare un'espressione regolare in formato UNICODE necessaria a individuare le emoji presenti all'interno del testo di input. Entrambe le funzioni verranno chiamate in momenti differenti dalla classe principale *SentimentAnalysis* che verrà analizzata in seguito.

All'interno delle operazioni di tokenization si attribuisce particolare importanza alla gestione delle maiuscole che devono essere preservate per permetterne l'analisi successiva. Entrambe le funzioni ricevono come input un testo e restituiscono un'array di elementi che corrisponde all'elenco di entità da analizzare.

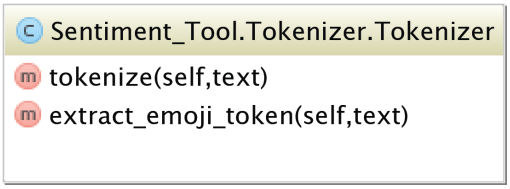


Figura 19: Class Diagram Tokenizer

```
def tokenize(self, s):
    """
    Argument: s -- any string or unicode object
    Value: a tokenize list of strings;
    """
    # Try to ensure unicode:
    try:
        s = unicode(s)
    except UnicodeDecodeError:
        s = str(s).encode('string_escape')
        s = unicode(s)

    # Tokenize:
    words = word_re.findall(s)

    return words
```

Figura 20: Codice Python funzione *tokenize()*

```
REGEXPS = (
    URLS,
    # Phone numbers:
    r"""
    (?:
      (?:            # (international)
        \+?[01]
        [\-\s.]*
      )?
      (?:            # (area code)
        [\(]?
        \d{3}
        [\-\s.\)]*
      )?
      \d{3}          # exchange
      [\-\s.]*
      \d{4}          # base
    )"""
    ,
    # ASCII Emoticons
    EMOTICONS
    ,
    # HTML tags:
    r"""<[^>\s]+>"""
    ,
    # ASCII Arrows
    r"""[\-]+>|<[\-]+"""
    ,
    # Twitter username:
    r"""(?:@[\w_]+)"""
    ,
    # Twitter hashtags:
    r"""(?:\#+[\w_]+[\w\'_\-]*[\w_]+)"""
    ,

    # Remaining word types:
    r"""
    (?:[a-z][a-z'\-_]+[a-z])       # Words with apostrophes or dashes.
    |
    (?:[+\-]?\d+[,/.:-]\d+[+\-]?)  # Numbers, including fractions, decimals.
    |
    (?:[\w_]+)                     # Words without apostrophes or dashes.
    |
    (?:\.(?:\s*\.){1,})            # Ellipsis dots.
    |
    (?:\S)                         # Everything else that isn't whitespace.
    """
    )

######################################################################
# This is the core tokenizing regex:

WORD_RE = re.compile(r"""(%s)""" % "|".join(REGEXPS), re.VERBOSE | re.I | re.UNICODE)
```

Figura 21: Codice Python per RegEx e funzione tokenize

5.3.3. Calcolo Polarità

La procedura di calcolo della polarità vera e proprio avviene all'interno della classe *SentimentAnalysis*. Al momento della sua inizializzazione viene chiamata la classe *LoadLexicon* che, come precedentemente descritto, carica i lessici dai rispettivi file e restituisce i dizionari che vengono salvati nelle variabili di stato*:*

- *negate,* array di stringhe contenente una lista di parole che esprimono negazione (*mai, nessuna, no, né, non, ecc.);*

- *booster_dict,* dizionario formato dalle coppie chiave valore dove il valore di polarità é -0.293, nel caso sia una booster negativo, o 0.293 nel caso sia un booster positivo (<*considerevolmente, 0.293*> , <*leggermente, -0.293* >*);*
- *contrastiveConj,* array di stringhe contenente le congiunzioni di contrasto (*ma, però, tuttavia*);
- *word_valence_dict,* dizionario principale contenente la coppia chiave valore in cui la chiave é rappresentata dalle parole del lessico e il valore é la polarità corrispondente;
- *emoji_valence,* dizionario costituito dal valore Unicode delle emoji e dal rispettivo valore di polarità associato.

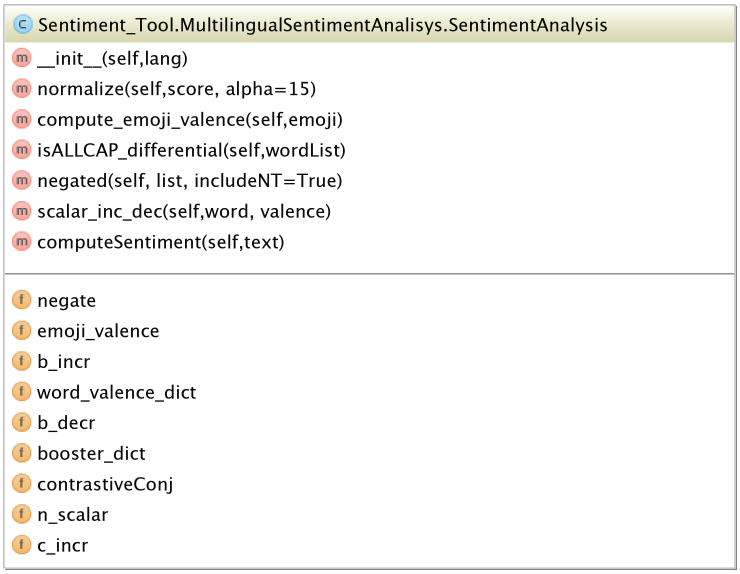


Figura 22: Class Diagram SentimentAnalysis

In questa fase vengono inoltre inizializzate le costanti *c_incr* con un valore di 0.733 che viene utilizzato per apportare una modifica in base alla presenza di parole maiuscole e *n_scalar* con un valore -0.74 impiegato per addurre una modifica alla polarità dovuta alla presenza di negazioni.

Successivamente all'inizializzazione del lessico, per effettuare l'analisi di un testo é necessario invocare la funzione *computeSentiment()* che prende come parametro di input appunto il testo da analizzare e restituisce un valore numerico di polarità. La prima parte di questa funzione si realizza attraverso la scomposizione del testo nelle sue componenti principali. Tramite la chiamata alle funzioni di *Tokenizer* otterremo quindi due array di stringhe chiamati *wordsAndEmoticons* ed *emoji_tokens* contenenti, rispettivamente, parole, emoticon e hashtag nel primo ed emoji unicode nel secondo.

In seguito, attraverso la funzione *isALLCAP_differential()* si calcola un valore booleano che controlla le maiuscole all'interno del testo verificando se le parole contenute nell'array

wordsAndEmoticons sono tutte minuscole o tutte maiuscole; in questi due casi la funzione restituisce *False* altrimenti torna *True*.

```
def isALLCAP_differential(self,wordList):
        countALLCAPS= 0
        # count number of word in wordlist that are CAPS
        for w in wordList:
            if str(w).isupper():
                countALLCAPS += 1

        #subtracts lenght of wordlist with number of CAPS word
        cap_differential = len(wordList) - countALLCAPS

        #check that not all word are CAPS and return TRUE
        if cap_differential > 0 and cap_differential < len(wordList):
            isDiff = True
        else: isDiff = False
        return isDiff
```

Figura 23: Codice controllo maiuscole

Come si può evincere dal frammento di codice in figura 23, la funzione riceve in input una lista di parole e restituisce un valore booleano che all'interno della funzione *computeSentiment()* viene immagazzinato nella variabile locale *isCap_diff* che verrà usata nel resto del calcolo della polarità totale.

A questo punto, per ciascuna delle parole contenute all'interno dell'array *wordsAndEmoticons* viene eseguita un'analisi che porta all'associazione di un valore di polarità. Per prima cosa si esamina se la parola in considerazione é contenuta negli array dei *modifiers:* in tal caso si passa immediatamente alla parola successiva associando zero come valore di polarità, poiché il loro compito é di modificare la connotazione delle parole che li seguono.

Nel caso non siano contenute negli array di modifiers, si controlla la corrispondenza all'interno del lessico e si estrae il valore di polarità *v*. Da questo momento in poi vengono effettuate le operazioni che portano delle modifiche al valore estratto dal lessico e riguardanti le regole sintattiche ampiamente discusse nei capitoli precedenti. Il primo controllo si occupa di verificare se la parola corrente é in maiuscolo e se la variabile *isCap_diff* é impostata a *True;* in tal caso viene verificato se *"v"* é positivo o negativo e in base al risultato viene ad esso sommato/sottratto il valore *c_incr*. Successivamente, verificando la posizione all'interno dell'array e quindi all'interno del testo, si procede controllando *bi-gram* e *tri-gram*. Questo permette di controllare la presenza di forme composte all'interno del dizionario principale *word_valence_dict* e all'interno di *booster_dict*.

Nel frammento di codice di figura 24 "*i*" fornisce l'indice alla posizione corrente della parola nel testo. Si controlla se esiste una parola all'interno del dizionario principale formata dalla parola corrente di posizione *i* e dalla parola di posto precedente *i-1*. Nel caso in cui non viene trovata una corrispondenza vengono controllate negazioni e booster attraverso le funzioni *negated()* e *scalar_inc_dec(),* ottenendo quindi un valore finale di polarità "*v*" per la parola analizzata. Nello stesso modo si procede per il controllo del tri-gram.

```
if i > 0:
    onezero = "{} {}".format(str(wordsAndEmoticons[i-1]), str(wordsAndEmoticons[i]))
    if onezero in self.word_valence_dict:
        v = float(self.word_valence_dict[onezero])
    else:
        s1 = self.scalar_inc_dec(wordsAndEmoticons[i-1], v)
        v = v+s1
        if self.negated([wordsAndEmoticons[i-1]]): v = v*self.n_scalar
```

Figura 24: Codice controllo bi-gram

Una volta effettuata l'analisi per tutte le parole dell'array si ottiene una lista di egual lunghezza contenente i valori di polarità delle singole parole, che sono estesi con la lista di valori associati all'emoji unicode calcolata tramite la funzione *compute_emoji_valence()*. A tal punto viene effettuato un controllo per la presenza di congiunzioni di contrasto.

L'analisi delle congiunzioni di contrasto, come é possibile vedere dall'immagine di figura 25, comincia cercando se una qualsiasi *contrastiveConj* sia contenuta all'interno di *wordsAndEmoticons*. Se viene scoperta una corrispondenza si estrae il valore che indica la posizione della congiunzione nel testo; si procede così a enfatizzare tutte le parole successive moltiplicando il valore di polarità per 1.5 e smorzando quelle precedenti moltiplicando per 0.5. Una volta eseguite queste operazioni si sommano tutti i valori ottenuti per poi effettuare il controllo finale sulla punteggiatura e infine il calcolo del risultato finale.

Per controllare la presenza di "!" e "?" si cercano le occorrenze e si contano fino a un massimo di quattro, moltiplicando la loro numerosità per un amplificatore uguale a 0.291 nel caso dei punti esclamativi e 0.96 per i punti di domanda. Successivamente, il valore viene aggiunto alla somma calcolata in precedenza. Il valore ottenuto infine, viene normalizzato per essere compreso tra -1 e 1 tramite la funzione *normalize()*.

```
for c in self.contrastiveConj:
    if c.lower() in wordsAndEmoticons or c.upper() in wordsAndEmoticons:
        try:
            bi = wordsAndEmoticons.index(c.lower())
        except:
            bi = wordsAndEmoticons.index(c.upper())
        for s in valence:
            si = valence.index(s)
            if si < bi:
                valence.pop(si)
                valence.insert(si, s*0.5)
            elif si > bi:
                valence.pop(si)
                valence.insert(si, s*1.5)
        break
```

Figura 25: Codice controllo congiunzioni di contrasto

5.3.4. Rest Web Server

Il Web Server é implementato all'interno del package *Web_API* utilizzando un framework chiamato Flask e, in particolare, una sua estensione utilizzata per la creazione di servizi REST in modo facile e veloce che prende il nome di *Flask-RESTful*.

La parte fondamentale di questa estensione é rappresentata dalle *Resouces* (risorse) che sono alla base di un servizio REST. Le risorse sono costruite su un oggetto *View* proveniente da Flask che offre la gestione di tutte le richieste effettuate sul protocollo HTTP implementando semplici metodi associati alla risorsa.

Nell'implementazione del Rest Web Server é stata creata una classe *Sentiment* che eredita da *Resources* e che possiede un metodo *post()* pronto a ricevere le richieste POST effettuate da un qualsiasi dispositivo presente in rete. Per gestire i dati ricevuti dalle richieste esistono molti moduli; nel caso in esame si é scelto di utilizzare il modulo *reqparse* che offre la possibilità di effettuare in modo automatico un processo di data validation. Infatti, nel caso in cui i parametri della richiesta non passino la validazione *Flask-RESTful r*isponde automaticamente con un messaggio di errore con stato http 400. In questo caso i parametri che possono essere accettati vengono impostati come stringhe essendo un testo e la lingua in formato ISO 639-2 (it, en, es., ecc…).

```
parser = reqparse.RequestParser()
parser.add_argument('lang', type=str, help='Language of text message')
parser.add_argument('text', type=str, help='Text of message')
args = parser.parse_args()
```

Figura 26: Codice data validation

Flask supporta una grande varietà di valori di ritorno all'interno delle funzioni che gestiscono le richieste. In questo caso la risposta viene fatta costruendo un JSON contenente i parametri calcolati durante l'esecuzione del metodo *post()*.

Per poter correttamente accedere alla *Resources* costruita é necessaria la dichiarazione di un *endpoint* associato ad un *URL* attraverso la funzione *add_resources(Sentiment, "/sent")* che riceve due parametri: il primo é la classe che rappresenta la risorsa e il secondo é invece *l'URL*. Il servizio sarà quindi raggiungibile ad un URL formato dall'indirizzo della macchina su cui risiede il server, seguito dalla porta su cui opera e dal path univoco che ne identifica la risorsa e che in questo caso sarà "/sent". In realtà *Flask-RESTful* permette di associare alla stessa risorsa una molteplicità di *URL* che devono essere elencati successivamente al nome della risorsa come ad esempio *add_resources(Sentiment, "/sent" , "/", "/s")*.

La fase di inizializzazione dei lessici, mostrata in figura 27, rappresenta la procedura più dispendiosa a livello di tempistiche, per questo motivo si é scelto di svolgerla in fase di avvio del server. Questo permette una risposta molto più rapida alle singole richieste. Insieme all'inizializzazione dei lessici vengono impostate anche le soglie che, confrontate con il valore

numerico, restituito dalla classe *SentimentAnalysis* permettono di annotare il testo analizzato con un valore discreto compreso tra {“positive, “negative”, “neutral”}.

```
min_neutral_tool = -0.25
max_neutral_tool = 0.25

it = m14.SimpleSentimentAnalisys("it")
en = m14.SimpleSentimentAnalisys("en")
fr = m14.SimpleSentimentAnalisys("fr")
es = m14.SimpleSentimentAnalisys("es")
pt = m14.SimpleSentimentAnalisys("pt")

class Sentiment(Resource):
    def post(self):
        ...
```

Figura 27: Codice inizializzazione variabili

6. VALUTAZIONI

Il presente capitolo analizza gli aspetti legati alla fase di valutazione dell'architettura creata nel corso di questo lavoro. La prima sezione mostra la progettazione e l'implementazione di un algoritmo di Web Scraping necessario a costruire le Ground Truth per la valutazione. Successivamente, nella seconda sezione vengono introdotte e spiegate le metriche che serviranno come indice di prestazione dello strumento e in particolare chiamate Precision, Recall e F1. Verrà poi mostrato il metodo seguito nella valutazione e la distribuzione delle Ground Truth. Infine, nelle ultime due sezioni, verranno spiegati i risultati ottenuti sui domini di applicazione Google Play, TripAdvisor e Twitter.

6.1. Creazione Ground Truth

Per la valutazione dello strumento descritto nei capitoli precedenti, capace di svolgere il processo di Sentiment Analysis su diciassette lingue diverse, é necessario trovare un metodo per ottenere un insieme di testi annotati da un essere umano attraverso un valore di opinione. Questi testi prendono il nome di Ground Truth (GT).

Una GT é utile per poter confrontare gli elementi provenienti da un'osservazione diretta di un utente umano con quelli calcolati attraverso un classificatore. Nei siti Web in cui é possibile recensire prodotti, hotel applicazione, ecc. la recensione sarà composta da un testo e da un numero di stelle compreso tra uno e cinque che ne rappresenta il grado di soddisfazione dell'utente. Il valore delle stelle é inserito dall'utente nel momento in cui scrive il testo della recensione ed é per questo motivo che viene considerata un'osservazione diretta. Sarà proprio confrontando il valore della stelle con la polarità calcolata attraverso il nostro classificatore che potremo misurare alcune metriche che ne definiscono le prestazioni. Per costruire una GT é quindi necessario estrarre le recensioni da appositi domini. In particolare per questo lavoro sono stati scelti TripAdvisor[46] e Google Play[47]. Il procedimento utilizzato per effettuare questa operazione utilizza il metodo di scraping Web di cui abbiamo parlato nel capitolo 2.

Per la procedura di scraping delle due risorse si é partiti visitando il portale di TripAdvisor di ogni Paese di cui si voleva ottenere il dataset e si é scelto per ognuno di questi Paesi 5 città in modo casuale. Successivamente, é stata avviata la procedura di estrazione delle recensioni riguardante tutte le città in questione.

Per Google Play il procedimento é stato leggermente differente ed é iniziato con la raccolta degli URL di tutti gli store Google Play di ogni Paese; solo successivamente si sono estratte le recensioni per tutte le applicazioni appartenenti a tutte le categorie ottenendo così un dataset molto ampio per ciascuna delle lingue ottenute. In tutti e due i casi lo scraper aveva il compito per ogni città di estrarre una coppia di valori per ciascuna recensione e rappresentata da:

< TESTO, STELLA>

Per poter valutare la GT così costruita con lo strumento di sentiment analysis realizzato in questa tesi, che restituisce come valore di output un valore discreto compreso tra { "positive", "neutral", "negative" }, é stato necessario convertire la valutazione assegnata dal numero di stelle in un valore anch'esso discreto. Come per il valore di polarità numerico utilizzato dal nostro strumento si é deciso di normalizzare il valore della stella tra (-1,1) e successivamente

[46] https://www.tripadvisor.it/

[47] https://play.google.com/store?hl=it

utilizzando la soglia T = ± 0.25 é stato convertito nella categoria coerente. Il risultato finale porta ad avere:

- *1,2 stella*: negative;
- *3 stella*: neutral;
- *4,5 stella*: positive.

Per effettuare la procedura di scraping di questi due domini di applicazione é stato scelto un framework Python chiamato Scrapy che può essere utilizzato anche per l'estrazione di dati strutturati usando API come ad esempio (Amazon Associates Web Services).

Gli oggetti necessari per la procedura di scraping di un qualunque dominio prendono il nome di Spider e per poterli definire devono implementare la classe con l'omonimo nome. Gli Spider definiscono come un particolare sito deve essere ispezionato, definendo quali link devono essere seguiti e come estrarre i dati strutturati che si desidera ottenere. La classe Spider non ha nessuna particolare funzione, se non quella di fornire alcuni metodi tra cui start_requests(), che all'avvio dell'esecuzione effettua le richieste agli url contenuti nell'array start_urls.

Un altro metodo messo a disposizione é parse(), che gestisce la risposta per ciascuna richiesta iniziale effettuata. Le risposte sono in formato HTML e possono essere ispezionate attraverso dei Selettori che permettono di muoversi al suo interno in modo agevole estraendone i dati o effettuando nuove richieste tramite i link che si incontrano. I Selettori sono così definiti perché selezionano determinate parti dell'HTML specificandole attraverso espressioni XPath[48].

All'interno del package Creating_Corpus_Reviews sono stati creati due spider chiamati GPlaySpider e TripAdvisorSpider che hanno il compito rispettivamente di estrarre testo e numero di stelle dalle recensioni di Google Play e TripAdvisor. Per poter iniziare con lo scraping é stato necessario inizializzare due array chiamati allowed_domains e start_urls. Il primo contiene l'elenco dei domini ammessi e che, nel caso di TripAdvisor, saranno molteplici e corrispondenti ai domini internazionali come ".com", ".fr", ".es" ecc. Per Google Play invece, il dominio per tutte le nazioni risulta essere sempre ".com".

Il secondo parametro deve contenere tutti gli URL da cui iniziare l'ispezione e che nel caso di TripAdvisor sono gli URL delle città scelte per ogni paese, ad esempio:

- *TripAdvisor Italia città Milano:* http://www.tripadvisor.it/Restaurants-g187849-Milan_Lombardy.html;

Per quanto riguarda invece Google Play l'array contiene l'elenco delle home page degli store internazionali come:

- *Google Play Store Stati Uniti:* https://play.google.com/store/apps?hl=en-US;

[48] XPath - linguaggio che permette di individuare i nodi all'interno di un documento XML, HTML o XHTML

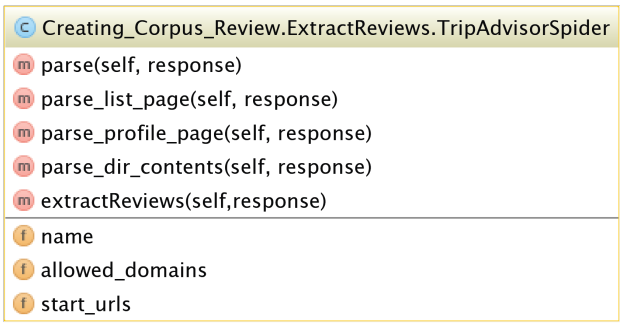


Figura 28: Class Diagram TripAdvisorSPider

- *Google Play Store Arabia Saudita :* https://play.google.com/store/apps?hl=ar;
- *Google Play Store Bulgaria:* https://play.google.com/store/apps?hl=bg.

In seguito verrà analizzato solo il procedimento svolto per l'estrazione delle recensioni da TripAdvisor, in quanto per Google Play é stata svolta la stessa procedura cambiando solo i Selettori utilizzati per estrarre gli elementi dalle pagine HTML.

Come si può vedere attraverso il class diagram in figura 28, oltre all'implementazione della funzione parse(), presente all'interno di qualunque Spider, é stato necessario creare dei metodi che gestissero ulteriori richieste. Infatti, attraverso i link inseriti nell'array start_urls viene restituito un elenco di hotel e ristoranti presenti all'interno della città selezionata; il problema risulta essere la presenza della paginazione che visualizza solamente una parte delle strutture presenti in città. Per questo motivo all'interno della funzione parse() vengono estratti i link della paginazione ed effettuate le richieste impostando come funzione di callback parse_list_page(). All'interno di questa funzione vengono estratti i link dell'elenco di hotel presenti all'interno nella pagina e analizzate le richieste che restituiranno la pagina profilo di ciascuna struttura che viene gestita dalla funzione parse_profile_page(). Anche in questo caso le recensioni presenti nella pagina profilo sono gestite attraverso un meccanismo di paginazione ed é per questo che prima di poter estrarre i dati bisogna recuperare tutti i link delle pagine. Le richieste di tali link sono gestite da parse_dir_contents() che, insieme al metodo extractReviews(), ha il compito di estrarre testo e numero di stelle da ciascuna recensione e salvarlo nell'apposito file il cui nome é rappresentato dal codice ISO 639-2 della lingua di cui si sta effettuando l'estrazione. In figura 29 si può osservare il frammento di codice che tramite un espressione XPath estrae le singole recensioni.

Una volta ottenuti i file di testo contenenti le recensioni provenienti sia da TripAdvisor che da Google Play, attraverso la classe BalanceReviews, si é effettuata una procedura di filtraggio per poter ottenere un dataset bilanciato contenente per ciascun numero di stelle la stessa quantità di recensioni. Per ogni file sono state contate le recensioni corrispondenti a una stella, due stelle e così via fino a cinque. Una volta ottenute le numerosità per ciascuna

```
###### XPATH: Select and Extract Text and Star ############
for review in hxs.select('//div[@class="innerBubble"]/div'):
    score = review.select('./div[@class="rating reviewItemInline"]/span[@class="rate sprite-rating_s rating_s"]/img/@alt').extract()
    reviewText = review.select('./div[@class="entry"]/p[@class="partial_entry"]/text()').extract()

    score = score[0][:1].encode('utf-8')
    reviewText = reviewText[0].encode('utf-8').replace("\n","").replace("\t","")
    out_file.write (reviewText+"\t"+score+"\n\n")
###########################################################
```

Figura 29: XPath estrazione testo, numero stelle

stella é stato preso il valore minimo *x* e successivamente sono state selezionate in modo casuale *x* recensioni per ciascuna stella, ottenendo così un dataset bilanciato.

All'interno della stessa classe si esercita un controllo sfruttando le API di Google Traslate per verificare che ciascuna delle recensioni sia realmente espressa nella lingua di cui si sta raccogliendo il dataset. Nel caso contrario si elimina. I valori finali estratti sono rappresentati in tabella 2, in cui per ogni lingua é mostrato il numero totale di recensioni e quello per ciascuna delle 5 categorie di stelle. Nel caso di TripAdvisor in alcuni paesi non é attivo per questo non é stato possibile ottenere le recensioni.

Figura 30: HTML recensioni TripAdvisor

Lingua	TripAdvisor	Google Play
Inglese	1395 (279 per stella)	525 (105 per stella)
Finlandese	-	3255 (651 per stella)
Danese	140 (28 per stella)	1935 (387 per stella)
Spagnolo	970 (194 per stella)	12295 (2459 per stella)
Arabo	435 (87 per stella)	2850 (570 per stella)
Greco	205 (41 per stella)	2860 (572 per stella)
Italiano	1670 (334 per stella)	14970 (2994 per stella)
Olandese	715 (143 per stella)	4375 (875 per stella)
Norvegese	295 (59 per stella)	2080 (416 per stella)
Portoghese	765 (153 per stella)	7655 (1531 per stella)
Rumeno	-	2165 (433 per stella
Lituano	-	1035 (207 per stella)
Sloveno	-	620 (124 per stella)
Belga	-	1160 (232 per stella)
Francese	1415 (283 per stella)	12270 (2454 per stella)
Tedesco	2735 (547 per stella)	14985 (2997 per stella)
Polacco	405 (81 per stella)	5730 (1146 per stella)

Tabella 2: Recensioni estratte

6.2. Metriche di valutazione

La valutazione sperimentale di un classificatore solitamente misura la sua efficacia, ovvero l'abilità di prendere la giusta decisione durante il processo di classificazione. Questa viene solitamente misurata in termini di *precision* (P) e *recall* (R). Per poter effettuare questa valutazione vengono quindi usate le GT estratte da Google Play e TripAdvisor associate con la loro etichetta compresa tra {positive, negative, neutral} precedentemente calcolata e confrontate con il valore computato dal classificatore.

Con *precision* (P) si intende la probabilità che un elemento classificato come appartenente alla classe c_i appartenga effettivamente alla classe c_i .

Con *recall* invece viene espressa la probabilità che un elemento appartenente alla classe c_i venga effettivamente classificato come appartenente alla classe c_i.
I risultati della classificazione di un dataset forniscono le seguenti quantità:

- *TP (true positives)*: documenti classificati c_i e appartenenti a c_i *;*
- *FP (false positives)*: documenti classificati c_i, ma non appartenenti a c_i *;*
- *TN (true negatives)*: documenti non appartenenti a c_i e non classificati c_i *;*
- *FN (false negatives)*: documenti appartenenti a ci, ma non classificati come ci .

Tramite queste quantità viene calcolato il valore di precision e recall attraverso le seguenti formule:

$$precision_i = \frac{TP_i}{TP_i + FP_i}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i}$$

Essendo il classificatore di tipo ternario é necessario calcolare valori di precision e recall per ciascuna delle tre classi {positive, negative, neutral}. Per poter ottenere un valore finale é stato utilizzato il metodo chiamato macroaveraging che calcola la media dei valori di precision e recall delle classi *C* utilizzate dal classificatore. In particolare:

$$precision^{\mu} = \frac{\sum_{i=1}^{3} precision_i}{|C|} \qquad recall^{\mu} = \frac{\sum_{i=1}^{3} recall_i}{|C|} \qquad F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Esiste inoltre un altro indice chiamato F1 che mette in relazione i concetti di precision e recall. Con l'F1 (F1-measure o F1-score) si vuole misurare l'accuratezza di un test. Può essere visto come la media pesata tra precision e recall ed é considerata con connotazione positiva quando tende a 1 e connotazione negativa quando tende a 0.

6.3. Metodo di valutazione

Il valore di polarità calcolato dal classificatore é in formato numerico, esso viene successivamente convertito in un valore discreto compreso tra {positive, negative, neutral}. Per la conversione, come é stato detto nel paragrafo 6.1, vengono utilizzate due differenti soglie impostate a {0} e {+0.25; -0.25} ottenendo di fatto un classificatore che può comportarsi in maniera differente in base al valore di soglia. Per questo motivo, nel calcolo dei valori di precision e recall si tiene conto di questa distinzione computando le metriche per entrambi i casi su tutte le GT.

Oltre a calcolare i valori di precision e recall per lo strumento creato é stato importante potersi confrontare con valori provenienti da altri strumenti presenti in letteratura e citati nel capitolo 2. Si é scelto come strumento di paragone SentiStrenght in quanto risulta essere il più popolare e in continua espansione su nuove lingue. Viene inoltre preso in considerazione anche uno strumento commerciale che offre API per il processo di Sentiment Analysis per la lingue inglese e italiana chiamato Dandelion API[49] di proprietà di SpazioDati[50]. Per questo strumento vengono utilizzate le GT in inglese e italiano mentre per SS vengono utilizzate in inglese, italiano,francese, portoghese e greco. Per tutte le altre GT avremo solamente i risultati provenienti dal classificatore oggetto del lavoro.

Per effettuare inoltre una valutazione che mettesse in risalto le caratteristiche dei componenti fondamentali dell'algoritmo rules-based, relative a booster, emoji, emoticon e

[49] https://dandelion.eu/

[50] http://spaziodati.eu/it/

punteggiatura é stato necessario svolgere un particolare procedimento. Per prima cosa si é estratta la parte di GT che contiene gli elementi analizzati dal componente in questione. Ad esempio, nel caso si voglia misurare l'incidenza del componente relativo alle emoji bisogna estrarre dalle GT tutte le recensioni contenenti le emoji e focalizzarsi solo su di esse.

Successivamente, essendo un algoritmo modulare, composto cioé da un gruppo di moduli funzionanti in maniera indipendente l'uno dall'altro, é stato effettuato il calcolo di precision e recall per questo sottogruppo di recensioni utilizzando l'algoritmo con il componente disattivato e con il componente attivo. In questa modo é possibile ottenere una visualizzazione dell'incidenza di ogni componente per ciascuna delle lingue e per entrambe le soglie di discretizzazione.

6.4. Distribuzione Ground Truth

Prima di mostrare i risultati ottenuti in termini di precision e recall é importante soffermarsi sulla descrizione delle GT mostrando le caratteristiche e le differenze tra Google Play (GP) e TripAdvisor (TA). In particolare si mostra come le differenti entità che vengono analizzate tramite l'algoritmo rules-based siano distribuite all'interno delle principali lingue europee ed in particolare per Inglese, Italiano, Francese, Portoghese, Spagnolo. Con entità si intendono emoticon, emoji, booster e punteggiatura. Questo permetterà di poter comprendere in maniera più chiara le metriche che verrano esposte successivamente.

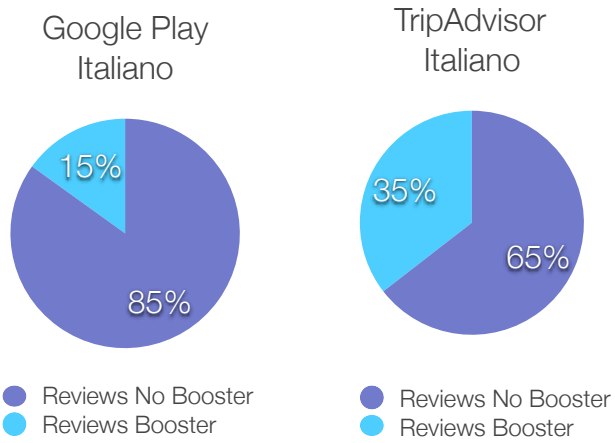


Figura 31: Diagramma a torta numerosità booster

La prima categoria analizzata nello specifico é stata quella dei booster. Con booster in questo specifico caso vengono intese tutte le categorie di modificatori di polarità e cioè booster positivi, booster negativi, negazioni e congiunzioni di contrasto.

Esiste una netta differenza tra GP e TA infatti, nel primo caso essendo le recensioni molto brevi (spesso composte da due o tre parole) questi elementi non sono molto presenti, in particolare li troviamo solo in una piccola parte di recensioni. Al contrario all'interno di TA

essendo i testi molto articolati la percentuale di presenza aumenta significativamente. Come possiamo vedere nei grafici in figura 31, in cui viene preso come esempio la lingua italiana, la percentuale di booster in GP risulta meno della metà rispetto a quella presente in TA. La stessa cosa vale per tutte le altre lingue prese in esame. Rispetto alle valutazioni assegnate dall'utente, la percentuale di booster tende a crescere all'aumentare del numero di stelle, tranne nel caso in cui si raggiunge il massimo valore in cui decresce leggermente. In questo caso il comportamento come si può vedere in figura 32 é il medesimo sulle due GT.

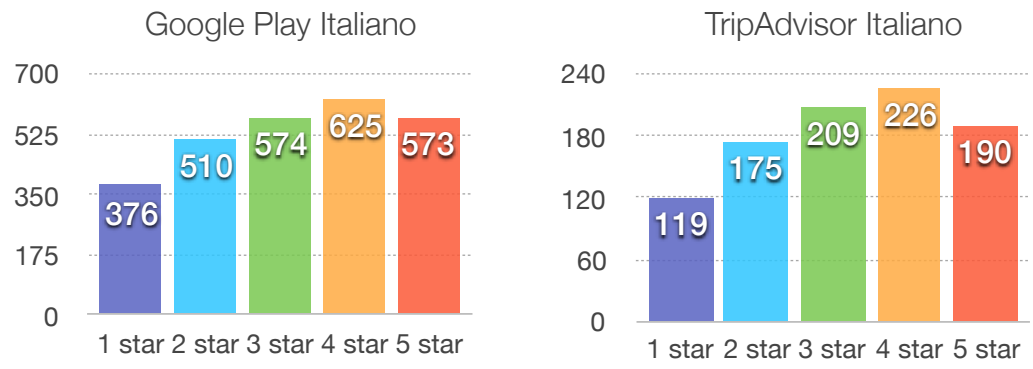


Figura 32: Istogramma distribuzione booster

La seconda categoria ad essere analizzata risulta essere la punteggiatura. Con punteggiatura intendiamo i punti esclamativi e i punti interrogativi che vengono solitamente utilizzati per rafforzare un opinione all'interno di un testo scritto.

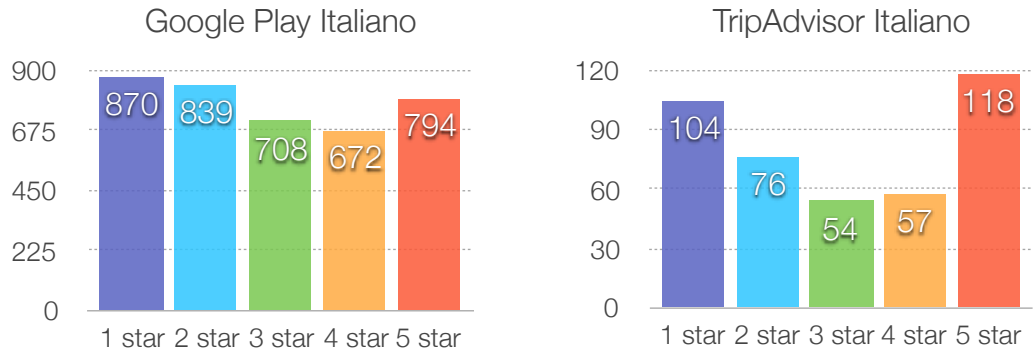


Figura 33: Istogramma distribuzione punteggiatura

In questo caso la numerosità concorda con le due tipologie di GT prese in esame ed in particolare corrisponde a più o meno il 20% dell'intero dataset. Risulta quindi molto interessante soffermarsi sull'analisi delle distribuzioni sui 5 valori possibili di stelle. Possiamo notare infatti (figura 33), come la punteggiatura venga usata maggiormente agli estremi quindi in recensioni con 1 e 5 stelle che possiamo immaginare corrispondano ad una opinione ottima o ad una opinione pessima. Tende invece a diminuire verso il centro

corrispondente alle 3 stelle, che rappresenta un'opinione neutra vicina quindi al concetto di oggettività. Questo comportamento é netto all'interno di TA ma risulta visibile anche in GP.

L'ultimo punto analizzato riguarda le emoticon e le emoji che, come abbiamo detto nei capitoli precedenti, risultano un ottimo elemento per poter capire l'opinione espressa dalle persone. La prima cosa evidente é che all'interno di TA entrambe le categorie sono praticamente assenti, questo si verifica in quanto il linguaggio con cui si scrive é prevalentemente formale. É all'interno di GP che possiamo vederne l'incidenza in quanto il linguaggio utilizzato é comune e molto simile, ad esempio, a quello utilizzato all'interno dei social media come Twitter. In particolare, si può vedere in figura 34 come le emoticon, che sono state le prime ad essere nate, siano attualmente inferiori all'emoji. Questo é dovuto al fatto che attraverso le emoji é possibile esprimere un'opinione molto più specifica e dettagliata oltre alla presenza di tastiere dedicate all'interno di qualsiasi dispositivo mobile e tablet e all'interno delle più famose applicazioni di messaggistica.

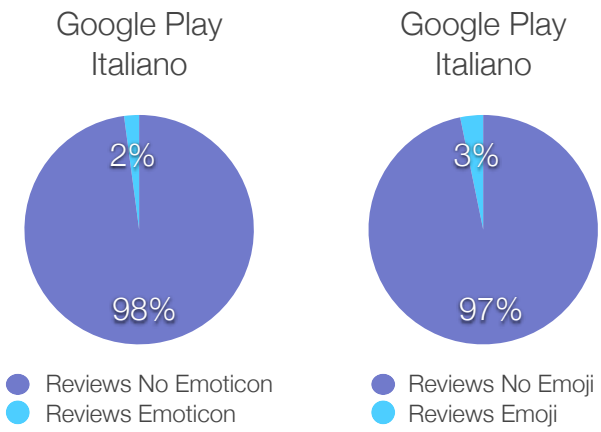


Figura 34: Diagramma a torta numerosità emoticon ed emoji

Nella distribuzione tra le categorie di stelle si può notare come l'utilizzo sia nettamente maggiore all'interno di recensioni che vogliono comunicare la soddisfazione dell'utente mentre risultano meno utilizzate all'interno delle recensioni con carattere negativo. In generale, per le emoji questo fenomeno si verifica in maniera meno netta in quanto essendo una vasta gamma possono spesso rappresentare in maniera molto espressiva opinioni negative.

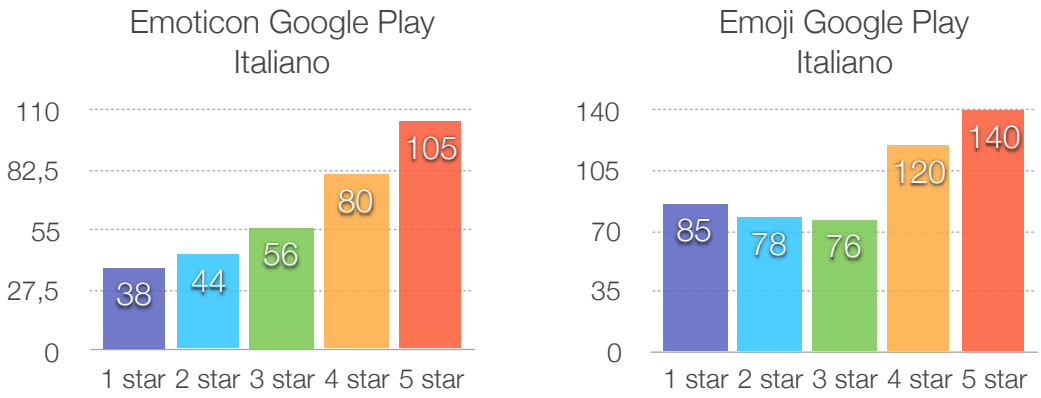


Figura 35: Istogramma distribuzione emoticon ed emoji

6.5. Valutazione delle prestazioni

L'analisi delle caratteristiche del dataset é utile per comprendere in maniera corretta i risultati ottenuti dal calcolo delle metriche di valutazione sulle classi positive, negative e neutral, ottenute dallo strumento di Sentiment Analysis e confrontate con il valore di polarità contenuto nelle GT. Tutti i risultati sono raccolti in Appendice| in apposite tabelle contenenti le differenti lingue e gli strumenti utilizzati su di esse con i corrispondenti valori di precision e recall. In questo paragrafo verranno mostrati i risultati più rappresentativi lasciando al lettore la possibilità di consultare i risultati completi.

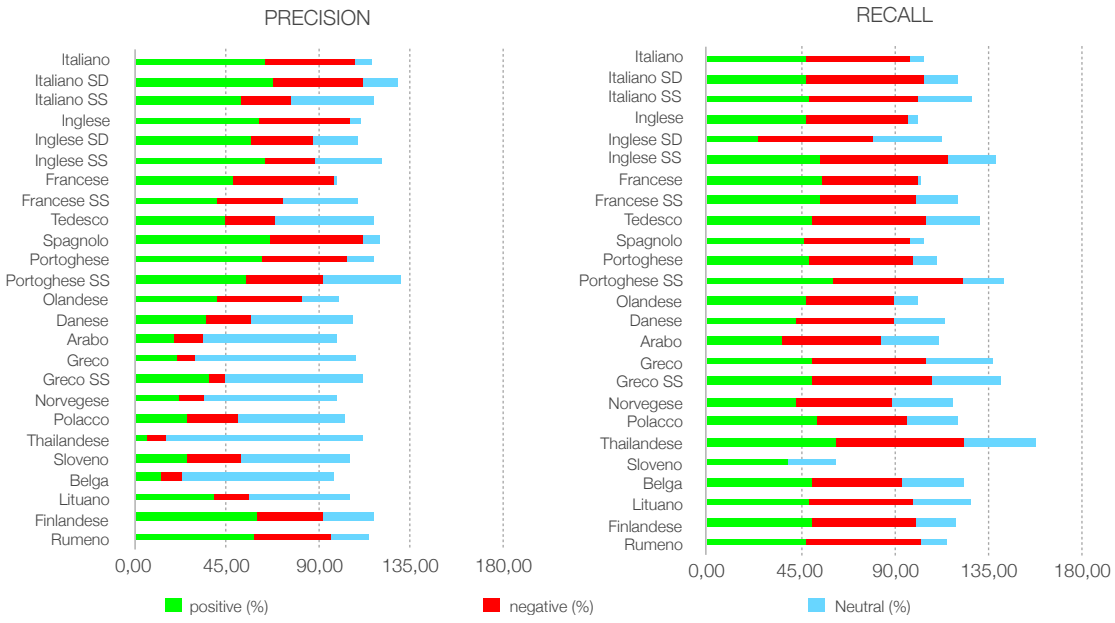


Figura 36: Precision e Recall Google Play (o) con SpazioDati (SD) e SentiStrength(SS)

Con soglia impostata a 0, come possiamo vedere in figura 36, i valori calcolati sulle GT di Google Play evidenziano come le lingue Italiano, Inglese, Spagnolo, Francese, Tedesco, Portoghese, Olandese, Finlandese e Rumeno possiedono valori di precision sui neutral molto bassi, mentre aumentano su positive e negative, in quanto la soglia risulta molto restrittiva. Le altre lingue possiedono invece valori più alti di precision sui neutral in quanto la qualità dei loro lessici é inferiore rispetto ai precedenti. Questo si verifica in quanto se all'interno di un testo non vengono trovate corrispondenze nel lessico allora il valore di polarità finale rimane a zero portando ad un'etichetta neutrale. Chiaramente un lessico povero di termini comporta un'alta probabilità di non trovare corrispondenze durante l'analisi del testo. Come si può vedere nelle tabelle A5, A6 anche su TripAdvisor mantenendo la soglia a 0 si hanno i valori di precision molto più alti sui positive e negative, mentre diminuiscono sensibilmente a favore delle precision sui neutral spostando la soglia a 0.25 (tabella A7, A8).

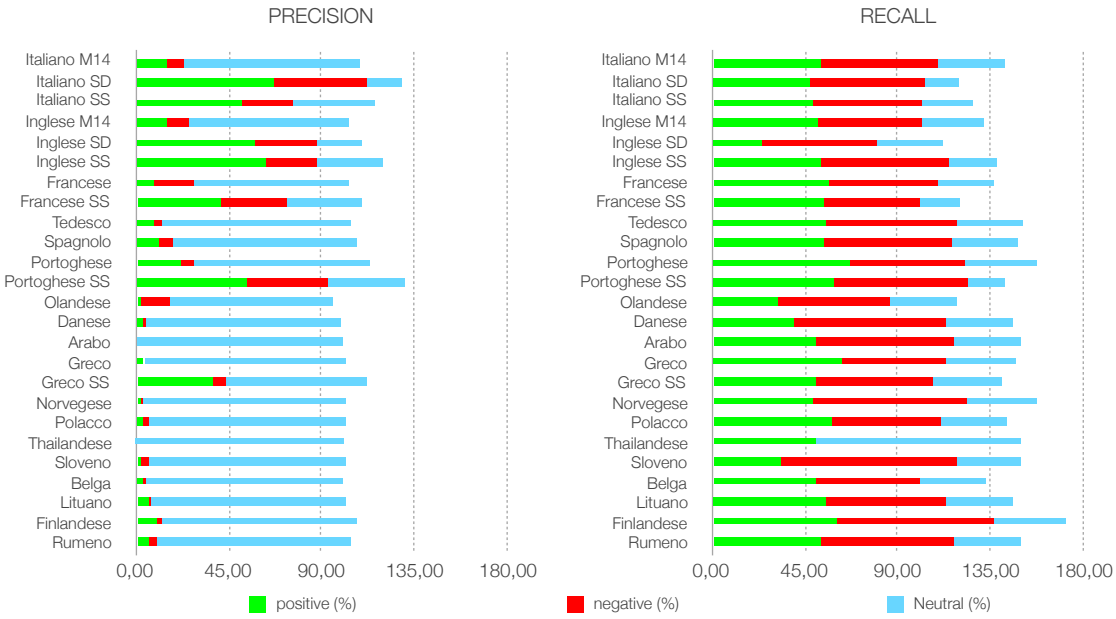


Figura 37: Precision e Recall Google Play (0.25) con SpazioDati (SD) e SentiStrength(SS)

In generale sia su Google Play che su TripAdvisor con soglia a 0 sia ha precisione più alta su positive e negative, perdendo nettamente sui neutral; viceversa, con soglia a 0.25 le precision su negative e positive diminuiscono e si incrementano quelle sui neutral. Per quanto riguarda i valori di recall, essi rimangono simili sulle classi positive e negative, sia con soglia a 0 che a 0.25, mentre subiscono un leggero incremento sulla classe neutral spostando la soglia da 0 a 0.25. Le differenze sono mostrate nelle figure 36 e 37 che si riferiscono alle tabelle A1, A2, A3 e A4 in Appendice|. Si possono vedere anche i valori calcolati con SentiStrenght (SS) e SpazioDati (SD) in cui i risultati migliori si trovano sulle classi positive e negative e risultano paragonabili allo strumento con soglia impostata 0.

Come é stato già spiegato nel paragrafo 6.3 una volta analizzate le GT in generale ci si é focalizzati sull'analisi di booster, emoji ed emoticon e punteggiatura. Per quanto riguarda la punteggiatura i risultati, che sono rappresentati nelle tabelle che vanno da A9 a A16, mostrano un piccolissimo incremento con soglia a 0 nelle precision sui positive rimanendo invariate su negative e neutral. L'aumento di precision sui positive si ha anche spostando la soglia a 0.25 ma in questo caso si verifica la diminuzione di precision e recall sulle classi negative e neutral.

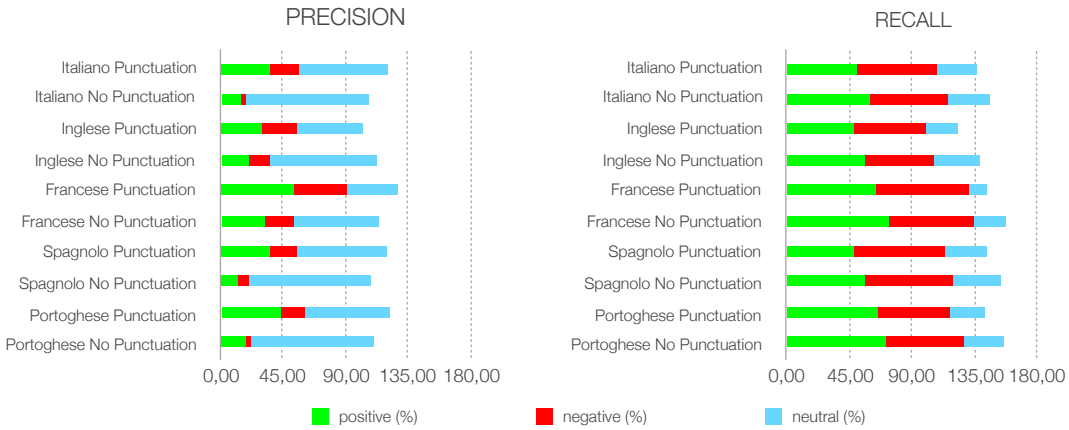


Figura 38: Precision e Recall puntegiatura Google Play (0.25)

Risultati molto interessanti si possono vedere in figura 39, che contenente i valori ottenuti nell'analisi della punteggiatura. Come mostrato nella distribuzione nelle GT la punteggiatura viene usata maggiormente su recensioni positive e negative. Guardando i valori di precision e recall si può vedere infatti un grosso incremento delle prestazioni soprattutto in termini di precision sulle classi negative e positive.

L'ultima analisi relativa alle emoji mostra un incremento netto sia su precision e recall che sulle classi positive e negative avendo però una diminuzione sui neutrali. Questo fa capire come l'utilizzo delle emoji e delle emoticon all'interno di un testo scritto lo renda soggettivo a priori. Tra queste due classi é però la positive a subire un aumento netto di performance.

Volendo ricapitolare i risultati ottenuti si può osservare che con soglia impostata a 0.25 lo strumento ha buone prestazioni nella classificazione di un testo come oggettivo o soggettivo, spostando la soglia a 0 questa capacità viene meno; tuttavia, aumentano sensibilmente le performance nell'individuazione delle classi positive e negative. Inoltre, emoticon ed emoji sono fondamentali nell'individuazione di opinioni positive e negative in quanto vengono poco utilizzate per identificare un parere neutrale. L'incidenza dei booster rappresentati da negazioni, congiunzioni di contrasto e intensificatori risulta invece molto bassa.

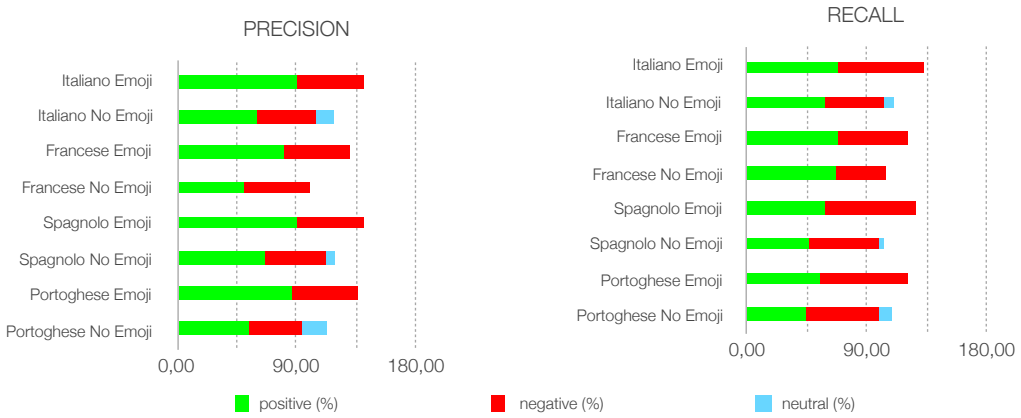


Figura 39: Precision e Recall emoji Google Play (0)

6.6. Valutazione correttezza

Un ulteriore valutazione all'interno di questo lavoro é stata condotta sul dominio di applicazione dei Social Media con lo scopo di ottenere un valore che indichi la precisione dello strumento realizzato per le lingue Italiano, Inglese e Spagnolo. Per poter effettuare questa procedura é stato necessario costruire un dataset di post multilingue sul quale poter svolgere la valutazione. Si é deciso quindi di utilizzare il servizio di Sentiment Analysis all'interno di un architettura proprietaria[51] che svolge operazioni di Social Media Analytics in real-time in modo da ottenere un dataset su cui lavorare ma allo stesso tempo testando il funzionamento dello strumento in un ambiente reale. In particolare l'architettura completa aveva il compito di monitorare la partita Roma - Real Madrid del 17 febbraio 2016 analizzando i post in lingua Italiano, Inglese e Spagnolo estratti da Twitter, Facebook e Instagram[52].

Il processo di valutazione della precisione consiste nel calcolare il numero di tweet correttamente classificati, rispetto al numero totale di classificazioni eseguite. Con "classificati correttamente" si intende che vengono classificati secondo la stessa polarità percepita da un annotatore umano. Chiaramente essendo il dataset molto ampio non é possibile controllare tutte le annotazioni, bisogna quindi utilizzare il concetto di campionamento statistico. Per campione statistico si intende un gruppo di unità che é sottoinsieme particolare dell'intera popolazione, individuato in modo da consentire, con un rischio definito di errore, la generalizzazione dell'intera popolazione. Applicato al nostro caso consiste quindi nel calcolare la dimensione minima sufficiente del campione di post, in modo da poterne verificare manualmente la correttezza di ognuno di essi ottenendo un valore di precisione del campione che sia generalizzabile sull'intero dataset con uno specifico errore.

[51] Fluxedo - http://www.fluxedo.com/it/social-listener

[52] Roma-Real by Fluxedo - http://www.fluxedo.com/socialometers/roma_realmadrid/

Il procedimento teorico prevede che come punto di partenza si annotino manualmente un numero arbitrario di tweet estratti in modo casuale dall'intero dataset e si stimi una proporzione $\hat{p}$ di annotazioni corrette. Tale proporzione seguirà una distribuzione binomiale (corretto / non corretto) che per grandi popolazioni é approssimabile a una distribuzione normale. Per questo motivo successivamente utilizzeremo il metodo Wald per stimare la grandezza effettiva del campione, dato un intervallo di confidenza *conf* e un errore *err* tale che il vero valore di precisione *p* sia p ∈ ($\hat{p}$ – err, $\hat{p}$ + err) con una percentuale di confidenza *conf*.

$$n = \frac{qnorm(conf)^2 \, \hat{p}(1-\hat{p})}{err^2}$$

Per poter utilizzare la formula enunciata è necessario scegliere i valori di errore e confidenza con *qnorm* che rappresenta il quantile della distribuzione normale. Come si può vedere dalla formula, più si scelgono valori stringenti, corrispondenti a errori ristretti e confidenza elevata, maggiore sarà la dimensione del campione statistico risultando quindi complicato da annotare manualmente. Per questo motivo vengono scelti i valori *err*=0.025 (i. e. 2.5%) e *conf*=95% trovando un compromesso ragionevole tra un buon grado di affidabilità del campione e un numero di post non eccessivo da annotare manualmente.

Il valore di precisione iniziale viene calcolato estraendo 300 tweet in modo casuale dal dataset completo e ottenendo un valore di correttezza del 50%. Applicando quindi la formula con questi parametri si ottiene un valore n = 1082. Si può quindi affermare che se su 1.082 post la correttezza é compresa tra 47.5% e 52.5% (50% +/- il 2.5% di margine di errore), allora la stima del 50% di precisione dell'algoritmo é validata con confidenza al 95%.

Partendo da questo concetto sono stati estratti casualmente 1082 post in Italiano, 1082 in Inglese e 1082 in Spagnolo e sono stati valutati da un gruppo di annotatori madrelingua che ne hanno valutato la correttezza ottenendo:

Lingua	Tweet estratti	Tweet corretti	Precisione
Italiano	1082	486	45,0%
Inglese	1082	548	50,7%
Spagnolo	1082	564	52,2%

Tabella 3: Risultati calcolo correttezza tweet

Dati questi valori si può affermare che la stima di precisione del 50% é quindi validata statisticamente per Inglese e Spagnolo, con confidenza 95% e errore 2.5%. Per quanto riguarda l'Italiano non avendo raggiunto un valore compreso tra 47.5% e 52.5% é necessario computare un nuovo valore di n tramite la precisione ottenuta del 45% per poi ripetere l'operazione verificando che la nuova correttezza sia compresa tra 42.5% e 47.5% (45% +/- il

2.5% di margine di errore). Il nuovo valore di *n* é però di 1072 che essendo inferiore a prima ci permette di affermare che la stima di precisione del 45% é validata statisticamente per l'Italiano, con confidenza 95% e errore 2.5%.

7. VISUALIZZAZIONE SENTIMENT

A seguito dello sviluppo di uno strumento con la capacità di annotare ogni testo ricevuto in input con un valore di polarità e una relativa etichetta compresa tra {positive, negative, neutral}, si é cercato di trovare un metodo che ne portasse alla luce le potenzialità all'interno di un dominio in continua espansione come quello di Twitter.

In particolare, la volontà é stata quella di visualizzare come il processo di Sentiment Analysis possa mostrare delle informazioni relative a un ampio numero di dati che altrimenti non si riuscirebbero a cogliere, ovvero quelle legate al pensiero generale degli utenti riguardo ad uno specifico argomento. Inoltre, sfruttando la proprietà di multilinguismo, risulta interessante focalizzarsi su un'area geografica in cui si parlano differenti lingue ed analizzare come il pensiero cambi da un paese all'altro. Viene mostrato anche il volume di informazioni che si muovono lungo una linea temporale mettendo in evidenza "quando", "quanto" e "come" le persone parlano di uno specifico argomento.

Per poter estrarre testi riguardanti uno stesso tema si é deciso di utilizzare come punto di partenza il meccanismo degli hashtag, usato ormai su tutti i principali social network. Gli utenti creano e utilizzano hashtag collocando il carattere hash # davanti a una parola o a una frase, nel testo principale di un messaggio o alla fine di esso.

Molto spesso le persone non condividono le stesse opinioni ed é proprio per questo che risulta interessante svolgere un ulteriore studio chiamato Contraddiction Analysis [21], attraverso il quale é possibile mostrare come le persone siano d'accordo o in contrasto su uno stesso tema.

L'obiettivo principale di questa fase consiste nella realizzazione di un'applicazione Web che permetta all'utente di ricercare un hashtag e di visualizzarne l'andamento relativo al sentiment, offrendo la possibilità di filtrare gli elementi rilevati in base alla lingua, alla posizione e al tempo. L'architettura realizzata si compone di un Rest Web Server che ha il compito di estrarre i dati relativi a gli hashtag e inviarli al client in un formato che sia facilmente interpretabile. Esisterà poi un interfaccia Web costruita sfruttando le principali tecnologie lato client quali HTML 5, CSS 3, Javascript e jQuery e che avrà il compito di visualizzare i dati ricevuti.

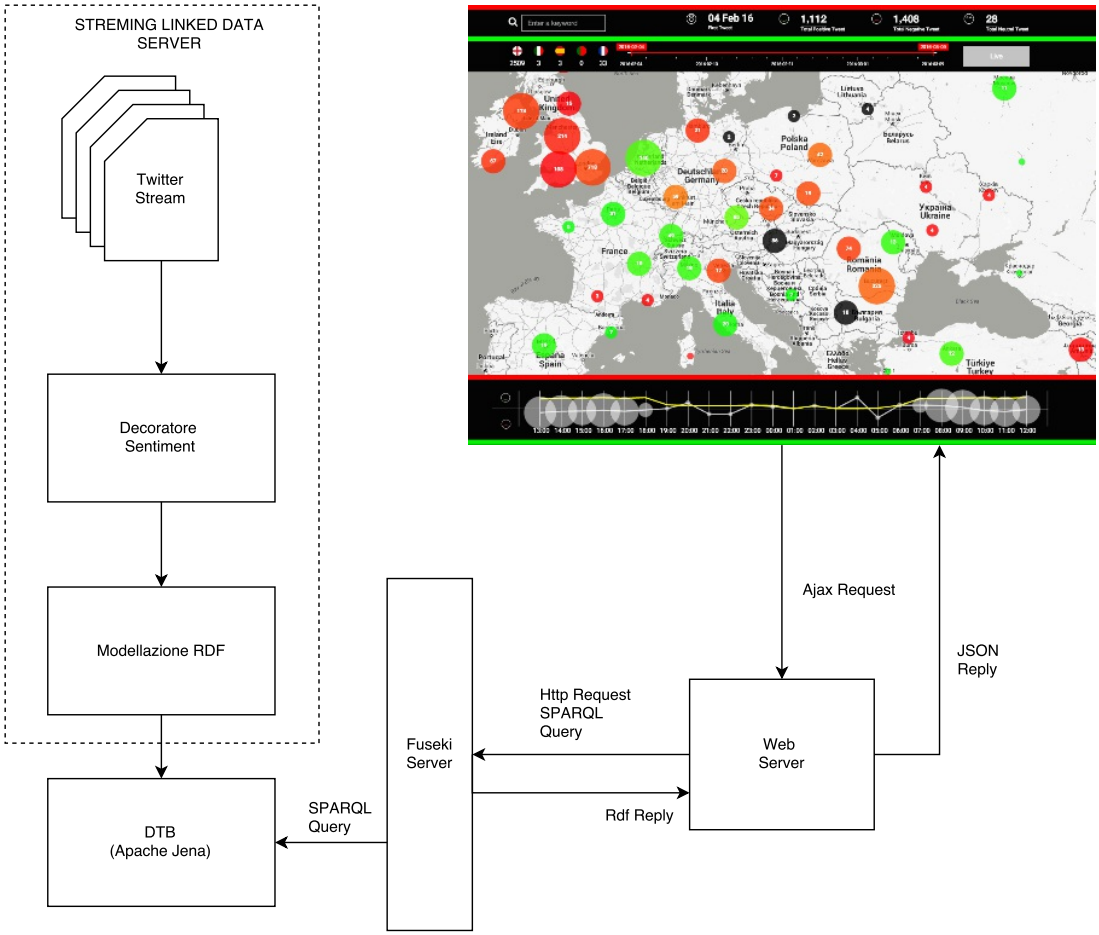


Figura 40: Architettura generale visualizzazione

7.1. Estrazione dati

I dati utilizzati sono raccolti ed immagazzinati in modelli RDF tramite il framework SLD [3]. Si tratta di tweet geo-localizzati nell'area dell'Europa e sono relativi ad un periodo che parte dal 04 febbraio 2016 ed é in continuo aggiornamento. I dati sono descritti da un'ontologia contenente tutte le informazioni relative ai singoli tweet con l'aggiunta di un valore di sentiment estratto attraverso lo strumento descritto nei capitoli precedenti.

Per la realizzazione dell'applicazione Web, i dati interessanti riguardano sostanzialmente gli hashtag, le indicazioni temporali e spaziali relative a ciascun tweet, l'id, la lingua di espressione oltre appunto al valore di polarità di ciascun elemento. I tweet, per poter essere elaborati, vengono inseriti in un database TDB[53], componente di Apache Jena, che permette

[53] https://jena.apache.org/documentation/tdb/index.html

l'immagazzinamento di dati e l'esecuzione di query in linguaggio SPARQL. Il dataset TDB é reso accessibile da remoto via HTTP attraverso un server SPARQL chiamato Fuseki[54]. La query necessaria ad estrarre i dati rilevanti ai fini dei nostri scopi é:

```
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix sioc:<http://rdfs.org/sioc/ns#>
prefix geo:<http://www.w3.org/2003/01/geo/wgs84_pos#>
prefix dt:<http://purl.org/dc/terms/>
prefix xsd:<http://www.w3.org/2001/XMLSchema#>
prefix sma:<http://www.citydatafusion.org/ontologies/2014/1/sma#>
select ?id ?sentiment  ?lat ?long ?data ?lang
where {
        ?mp sioc:id ?id ;
        sma:sentiment ?sentiment ;
        dt:created ?data ;
        dt:language ?lang ;
        geo:location ?location ;
        sioc:topic ?topic .
        ?location geo:lat ?lat ;
        geo:long ?long .
        ?topic rdfs:label ?label .
        """+hashtag+"""
       }
```

Figura 41: Query SPARQL

Come possiamo notare dalla query in figura 41 viene utilizzato un parametro hashtag che permette di ottenere tutti i tweet contenenti quel particolare elemento. La query restituisce una lista di elementi in formato RDF contenenti:

- *ID,* relativo al tweet in considerazione;
- *sentiment*, ovvero il valore di polarità numerico;
- *latitudine*, coordinata per permettere la geolocalizzazione;
- *longitudine*, coordinata per permettere la geolocalizzazione;
- *data*, data e ora in cui é stato pubblicato il tweet.

Come per il Server Web utilizzato per esporre il servizio di Sentiment Analysis anche in questo caso é stato utilizzato il framework Flask-Restful e allo stesso modo é stato implementato un metodo post(). Questo metodo ha il compito di gestire le richieste inviate tramite javascript contenenti un parametro che rappresenta l'hashtag inserito dall'utente ed effettuare la query SPARQL verso l'endpoint. Il risultato della query é espresso in formato RDF e viene formattato sfruttando il formato JSON gestibile facilmente sia lato server che lato client. Un esempio di tweet in formato JSON é:

[54] https://jena.apache.org/documentation/serving_data/

```
{
    'polarity': u'0.0875',
    'lang': u'en',
    'sentiment': 'positive',
    'longitude': u'-1.647135',
    'latitude': u'52.807792',
    'date': u'2016-02-04T10:21:11+01:00',
    'id': u'695175255317319680'
}
```


7.2. Interfaccia Web

L'interfaccia Web é realizzata sfruttando una tecnica di sviluppo software per la realizzazione di applicazioni web interattive chiamata AJAX (Asynchronous JavaScript and XML). Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio asincrono di dati in background fra web browser e server che consente l'aggiornamento dinamico di una pagina web senza che venga ricaricata dall'utente.

La tecnica Ajax utilizza HTML e CSS per lo stile dell'interfaccia, Javascript o jQuery per la modifica del DOM (Document Object Model) in modo da poter mostrare i dati ed interagirvi e l'oggetto XMLHttpRequest o la funzione jQuery ajax() per l'interscambio asincrono dei dati tra il browser dell'utente e il web server. In genere viene usato XML come formato di scambio dei dati anche se di fatto qualunque formato può essere utilizzato, incluso testo semplice o HTML preformattato. Per questo lavoro é stato scelto il formato JSON essendo particolarmente adatto allo scambio di informazioni con il server Web.

Il codice della richieste POST effettuate dal client web é mostrato in figura 42. I parametri fondamentali impostati per la richiesta sono: *url*, che contiene l'indirizzo del Web Server, *type* che identifica il tipo della richiesta http (in questo caso é POST), e *data* contenente l'hashtag inserito dall'utente. Il parametro *success* imposta una funzione di callback che verrà eseguita nel momento in cui i dati sono ricevuti dal Web Server.

La rappresentazione generata può essere suddivisa in tre parti fondamentali:

- visualizzazione dei dati generali riguardanti l'intero set di tweet contenenti l'hashtag inserito dall'utente tramite l' apposita form;
- illustrazione dei tweet all'interno di una mappa con la presenza di appositi filtri per poter selezionare le lingue e le finestre temporali;
- identificazione di una timeline con le frequenze dei tweet su tutte le ventiquattro ore giornaliere, in questo punto verrà mostrato l'andamento medio del sentimento nel tempo e l'andamento della contraddizione.

```
$.ajax({
            url: form_url,
            type: form_method,
            data:"hashtag=" + input,
            dataType: "json",
            context: document.body,
            cache: false,
            success: function(json){

                ...


            }
        });
```

Figura 42: Richiesta AJAX

7.2.1. Visualizzazione Generale

La prima parte dell'interfaccia si occupa di visualizzare i dati generali riguardanti l'hashtag inserito dall'utente. Questi dati sono:

- numero totale di tweet positivi;
- numero di tweet negativi;
- numero di tweet con polarità neutrale;
- data in cui é comparso per la prima volta l'hashtag dal momento in cui é stata avviata l'applicazione;

Figura 43: Visualizzazione generale

In questa parte é presente anche una casella di testo in cui é possibile inserire un nuovo hashtag e che, attraverso una procedura di autocompletamento, consiglia i top 200 hashtag più utilizzati.

7.2.2. Visualizzazione Mappa

Per la visualizzazione dei tweet geolocalizzati é stato deciso di utilizzare una mappa sfruttando le API Javascript versione 3 messe a disposizione da Google Maps[55]. La mappa

[55] https://developers.google.com/maps/documentation/javascript/reference

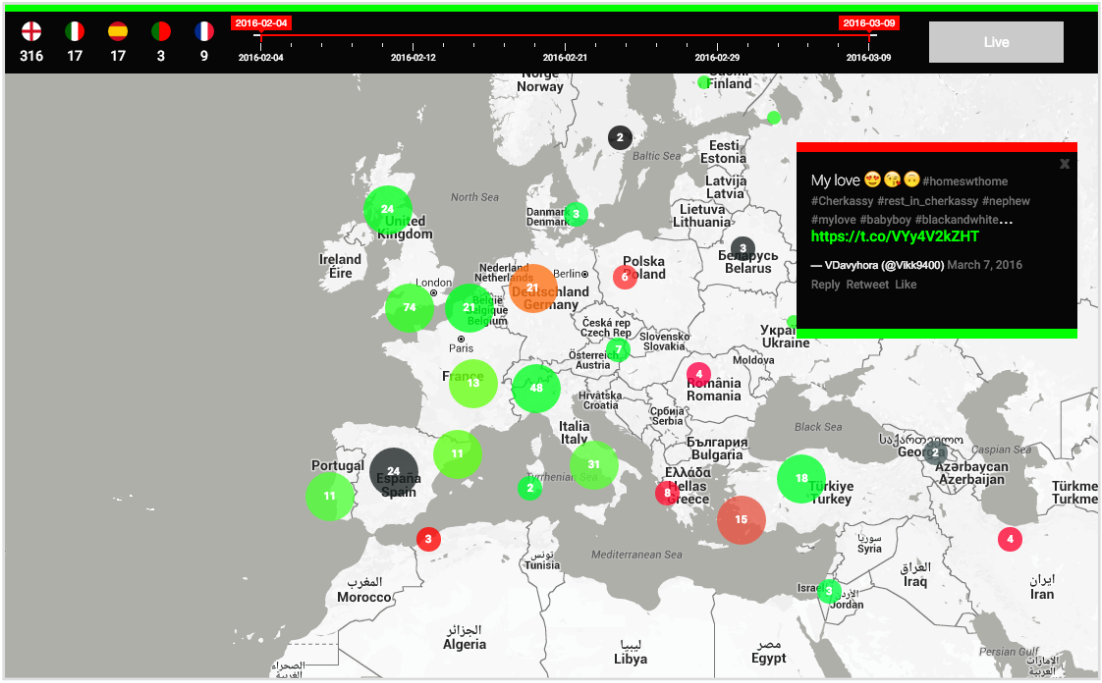


Figura 44: Visualizzazione mappa

permetterà di visualizzare i dati relativi all'hashtag che viene cercato fornendo una visione generale delle tre classi positive, negative e neutral e permettendo di filtrare i tweet in base alla loro lingua, alle finestre temporali e alla loro posizione. Infine, sarà possibile leggere il testo di ogni singolo tweet attraverso un'apposita InfoBox[56]. Ogni tweet é rappresentato attraverso un marker, posizionato in base ai valori di latitudine e longitudine di forma circolare la cui colorazione cambia in base alla sua polarità. Vengono utilizzati i colori verde, rosso e grigio per evidenziare rispettivamente elementi positivi, negativi e neutrali.

La raffigurazione di un gran numero di questi marker sulla mappa risulta essere poco intuitiva e difficilmente interpretabile dall'utente, per questo motivo é stato deciso di applicare un processo di clustering che porta ad unificare i marker in gruppi chiamati cluster in base alla loro vicinanza. Esistono molti metodi di clustering; tuttavia, nel caso in questione é stato scelto l'approccio grid-based implementato dalla libreria Google MarkererCluster[57].

Il grid-based clustering opera dividendo la mappa in quadrati di una determinata dimensione (che varia a seconda dello zoom) raggruppandoli all'interno di quest'area e visualizzandoli con un unico elemento. Ogni cluster viene rappresentato da un cerchio il cui raggio é proporzionale al numero di marker che contiene. Per rendere più intuitiva l'interpretazione delle quantità all'interno dei cluster é stato deciso di variare il raggio su una scala di cinque misure in base alla quantità di tweet:

---

[56] http://google-maps-utility-library-v3.googlecode.com/svn/trunk/infobox/docs/reference.html

[57] http://google-maps-utility-library-v3.googlecode.com/svn/trunk/markerclustererplus/docs/reference.html

- *Misura 1 (25 x 25 px),* per numero di marker tra 1 e 10;
- *Misura 2 (50 x 50 px),* per un numero di marker tra 10 e 100;
- *Misura 3 (75 x 75 px),* per un numero di marker tra 100 e 1000;
- *Misura 4 (100 x 100 px),* per un numero di marker tra 1000 e 10.000;
- *Misura 5 (125 x 125 px),* per un numero di marker tra 10.000 e 100.000.

Per la rappresentazione del colore invece viene considerata l'incidenza delle tre categorie di polarità (positive, negative, neutral) per ottenere un colore che ne sia la combinazione e porti quindi a percepire in modo intuitivo quali sono le categorie più presenti nel cluster. Le tre classi sono rappresentate dai colori verde (positive), rosso (negtive) e grigio (neutral), la costruzione del codice RGB é realizzata calcolando la numerosità delle tre classi e ampliando la componente relativa alla classe più numerosa. Si fa attenzione a smorzare la componente neutrale per mettere in risalto maggiormente i positivi e i negativi. All'interno di ogni cluster viene mostrato un numero che indica il totale di tweet presenti al suo interno. Cliccando all'interno di ogni cluster viene effettuata automaticamente una procedura di zoom che porta alla suddivisione dei cluster in cluster più piccoli, in base al cambiamento della dimensione della griglia di clustering fino ad arrivare al singolo marker.

Per ottenere il tweet da mostrare all'interno dell'InfoBox (estensione di Google Maps), una volta selezionato il marker, é necessario effettuare una richiesta alle rest API di Twitter inviando come parametro l'id del tweet. In questo caso viene effettuata una richiesta come mostrato in figura 45, ricevendo come risposta un JSON contenente un codice HTML da inserire nell'interfaccia. Il codice da inserire può essere visualizzato utilizzando lo stile di Twitter sfruttando il widget ufficiale, ma per coerenza di grafica é stato deciso di personalizzarlo mantenendo i tasti like, retweet e replay come richiesto nelle policy di Twitter.

Come si può vedere in figura 44, nella parte superiore della mappa sono posizionati i filtri necessari alla selezione di un sottogruppo rispetto al totale dei tweet. Verranno mostrati il numero di tweet per ogni lingua ed é possibile selezionare le lingue di cui si vogliono mantenere gli elementi scartando tutti gli altri. Si ha inoltre, uno slider temporale che permette di impostare una finestra tramite una data e un'ora di inizio, una data e un'ora di fine. Il bottone Live servirà a impostare la finestra temporale nelle ultime ventiquattro ore.

```
$.getJSON("https://api.twitter.com/1/statuses/oembed.json?id="+id+"&align=center&callback=?",
        function(data){

        }
    );
```

Figura 45: Richiesta API Twitter

7.2.3. Visualizzazione Timeline

L'ultima parte dell'interfaccia include una timeline che visualizza tre informazioni differenti su una finestra temporale:

- *Frequenze:* mostra le frequenze di tweet per ognuna delle 24 ore giornaliere;
- *Media:* identifica il valore medio di sentiment e come esso evolve all'interno delle 24 ore giornaliere;
- *Contraddizione:* visualizza l'andamento della contraddizione media all'interno delle delle 24 ore giornaliere.

Per poter effettuare questa visualizzazione é necessario che i tweet vengano raggruppati in base all'ora in cui sono stati pubblicati ottenendo quindi una lista di ventiquattro *array*: uno per ogni ora del giorno. Come si può vedere nella figura 46 le frequenze dei tweet vengono rappresentate come dei cerchi allineati su ciascuna ora del giorno. Il calcolo delle frequenze é molto semplice e corrisponde alla dimensione degli array costruiti in precedenza. Il raggio del cerchio é proporzionale al numero di tweet contenuti e va da una grandezza minima di 0 px a una grandezza massima di 35 px. Il conteggio viene effettuato considerando la frequenza massima e attribuendogli il raggio maggiore, le altre frequenze vengono calcolate come segue:

$$r_h = \left(\frac{size\left(array_h\right)}{\max} \cdot 35px \right)$$

Per il calcolo dell'andamento medio del sentiment e della contraddizione rappresentate in figura 46 rispettivamente dalla linea grigia e dalla linea gialla é stata utilizzata una libreria javascript SimpleStatistics.js[58] contenente funzioni per il calcolo di variabili statistiche. Il primo valore corrisponde al calcolo della media statistica della variabile continua polarity per ognuno dei ventiquattro array corrispondenti alle ore del giorno. Questo valore esprime la dominanza di opinione positiva negativa o neutrale all'interno di un argomento per ciascuna finestra temporale giornaliera.

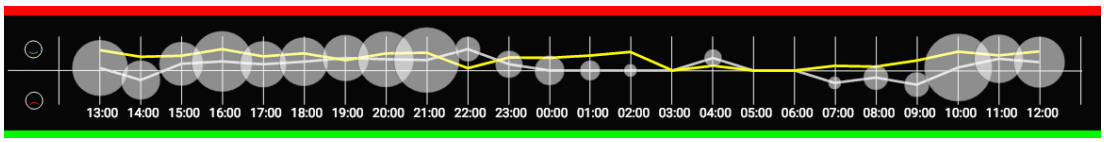


Figura 46: Timeline

[58] http://simplestatistics.org/

Una volta estratto il valore medio di ciascun array é necessario posizionare i punti all'interno della timeline e unirli attraverso la linea grigia. Si calcola il valore massimo delle medie e si procede associandolo alla massima distanza dall'asse x della timeline che, in questo caso é 25px. Il valore di distanza viene calcolato come segue:

$$d_h = \frac{mean\left(array_h\right)}{\max} \cdot 25px$$

Chiaramente, se il valore relativo alla media risulta positivo verrà sommato alla y=0 altrimenti verrà sottratto. Una volta calcolati i punti vengono uniti tramite una linea e i cerchi relativi alle frequenze vengono centrati in essi.

L'ultimo valore da ricercare riguarda invece i valori di contraddizione. Si potrebbe utilizzare la media, più in particolare una media vicino allo zero, per rappresentare un alto livello di contraddizione; tuttavia, questo procedimento non é sempre vero in quanto potrebbero esserci post neutrali che portano la media a zero ma non evidenziano un valore di contraddizione. Per questo motivo viene utilizzata la varianza: infatti, più alta é la varianza maggiore sarà il tasso di variabilità della polarità all'interno di ciascun array. In particolare viene calcolato un valore C di contraddizione per ognuna delle ventiquattro ore come segue:
Al denominatore é stato aggiunto α che permette di limitare il valore di contraddizione quando la media é uguale a zero.

$$C_h = \frac{var\left(array_h\right)}{\alpha + \left(mean\left(array_h\right)\right)^2}$$

Questa formula cattura l'intuizione che la contraddizione C sarà alta quando il valore medio é vicino allo zero e la varianza é grande. Una volta calcolati i ventiquattro valori di contraddizione vengono posizionati sulla timeline utilizzando lo stesso metodo descritto per la visualizzazione della media.

7.3. Esempi di utilizzo

Attraverso l'utilizzo dell'applicazione Web costruita, é possibile mostrare alcuni esempi interessanti che evidenziano l'andamento dell'opinione relativa ad alcuni hashtag che costituiscono un trend, ovvero un principale tema di discussione intorno al quale vengono scambiati messaggi da una grande maggioranza di utenti. In particolare vengono presentati gli hashtag *#spring*, *#design*, *#fashion*, *#morning*, *#sunrise*, *#sunset*, *#night*, *#job* e *#snow* analizzando le informazioni che si deducono dall'interfaccia.

In figura 47 viene mostrato il primo esempio relativo all'hashtag *#spring* in cui si può vedere una distribuzione uniforme e altamente positiva su tutta l'area geografica dell'Europa tranne che per il Portogallo in cui é leggermente negativa.

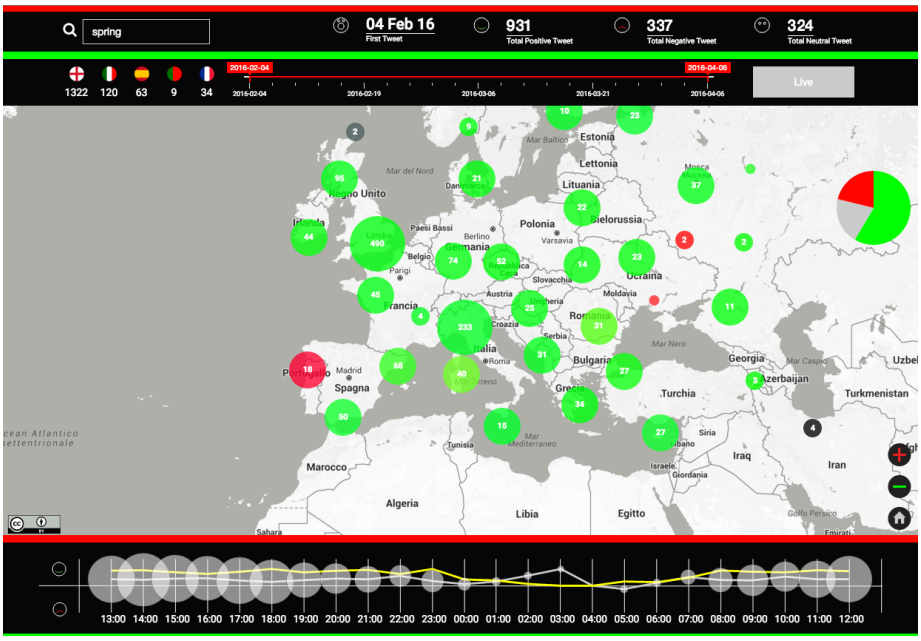


Figura 47: Visualizzazione Sentiment per l'hashtag #spring

All'interno della Timeline si vede anche qui una andamento uniforme su tutte le ore del giorno che va pian piano diminuendo fino quasi a scomparire nelle ore notturne. In particolare le concentrazioni più alte di tweet si hanno durante il pomeriggio a partire dalle 12. Simili andamenti si possono vedere in figura 48 e 49 anche per gli hashtag *#fashion* e #design anche se per quest'ultimo vengono presi in considerazione molti meno tweet rispetto agli altri due.

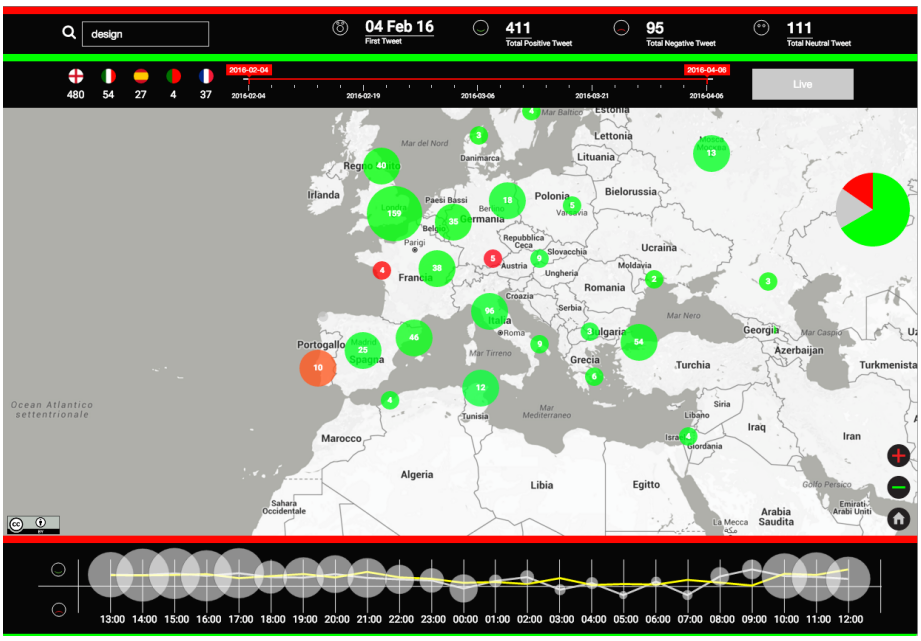


Figura 48: Visualizzazione Sentiment per l'hashtag #design

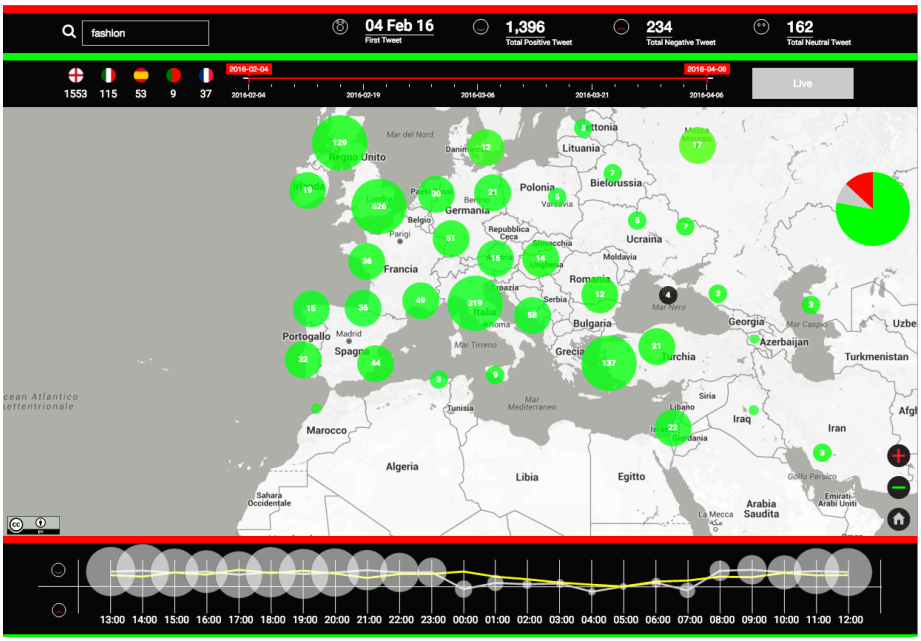


Figura 49: Visualizzazione Sentiment per l'hashtag #fashion

Nelle figure 50, 51, 52 e 53 vengono invece mostrati alcuni tra gli hashtag che rappresentano i momenti più importanti della giornata come *#morning*, *#sunrise*, *#sunset* e *#night*.

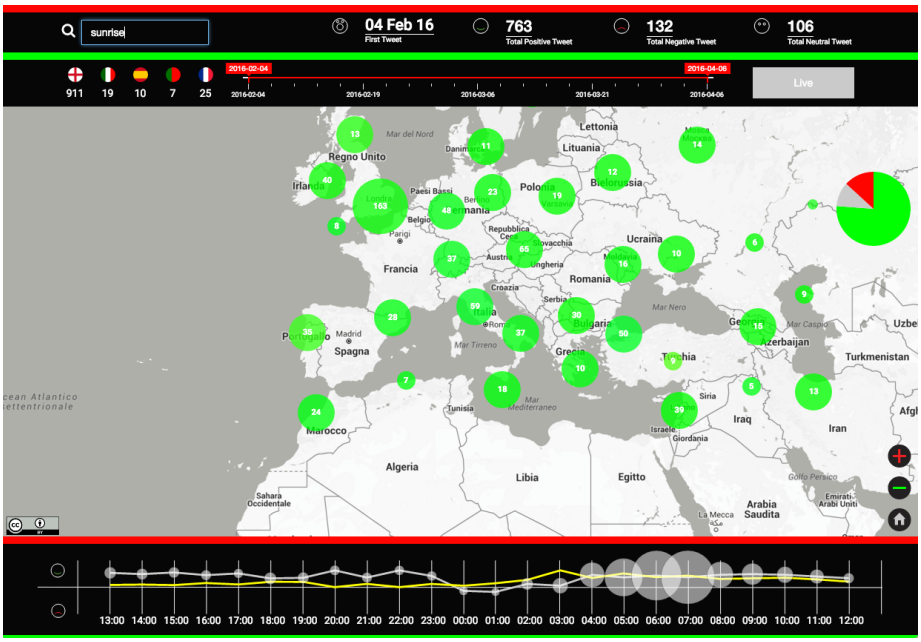


Figura 50: Visualizzazione Sentiment per l'hashtag #sunrise

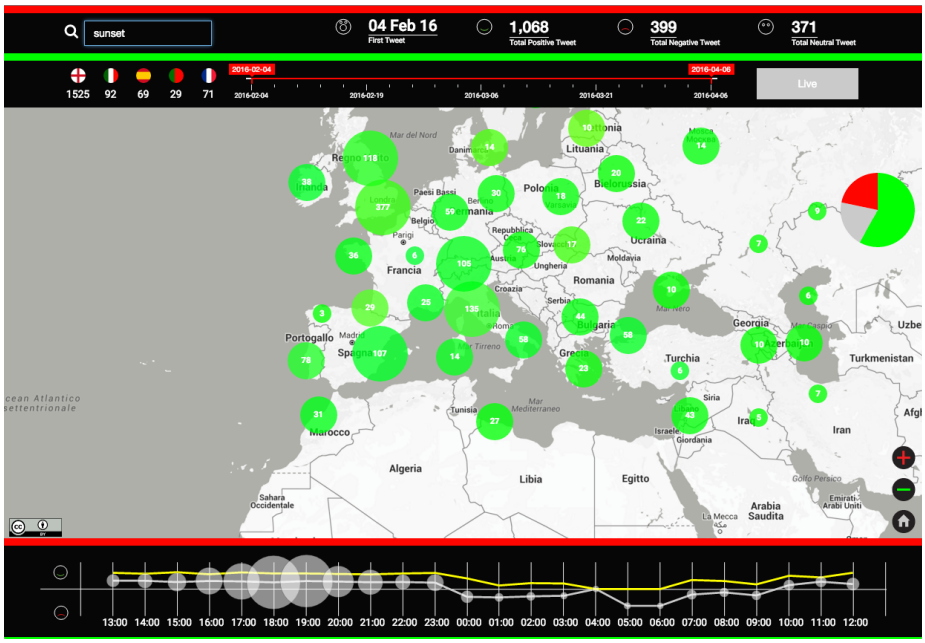


Figura 51: Visualizzazione Sentiment per l'hashtag #sunset

Per quanto riguarda *#sunrise* e *#sunset* abbiamo una distribuzione maggiormente positiva su tutta l'Europa concentrata però sono in alcuni momenti della giornata, in particolare *#sunrise* viene utilizzato nelle prime ore del giorno indicando appunto l'alba, mentre *#sunset* viene utilizzato tra le ore 16 e le 21 in cui tramonta il sole.

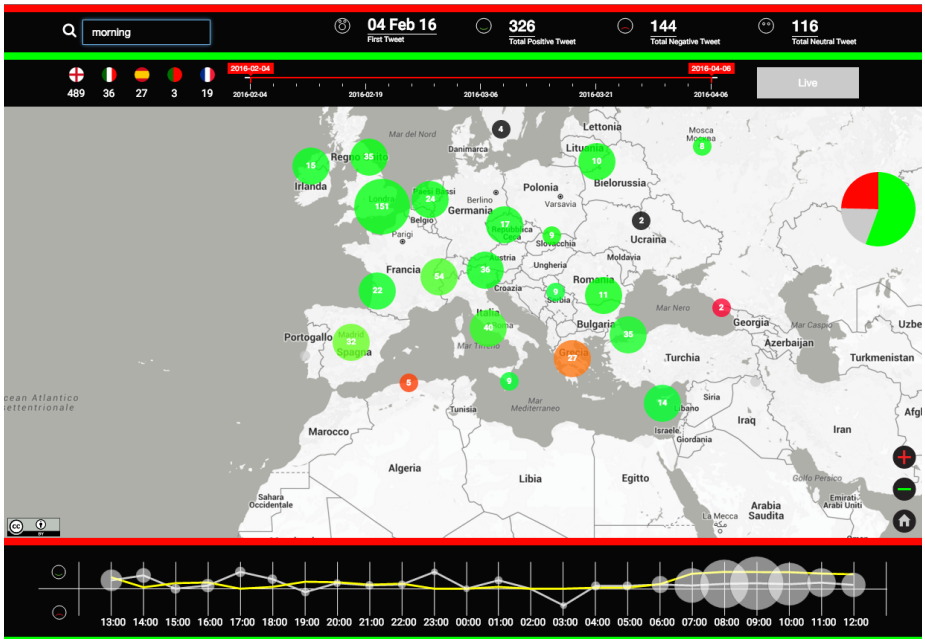


Figura 52: Visualizzazione Sentiment per l'hashtag #morning

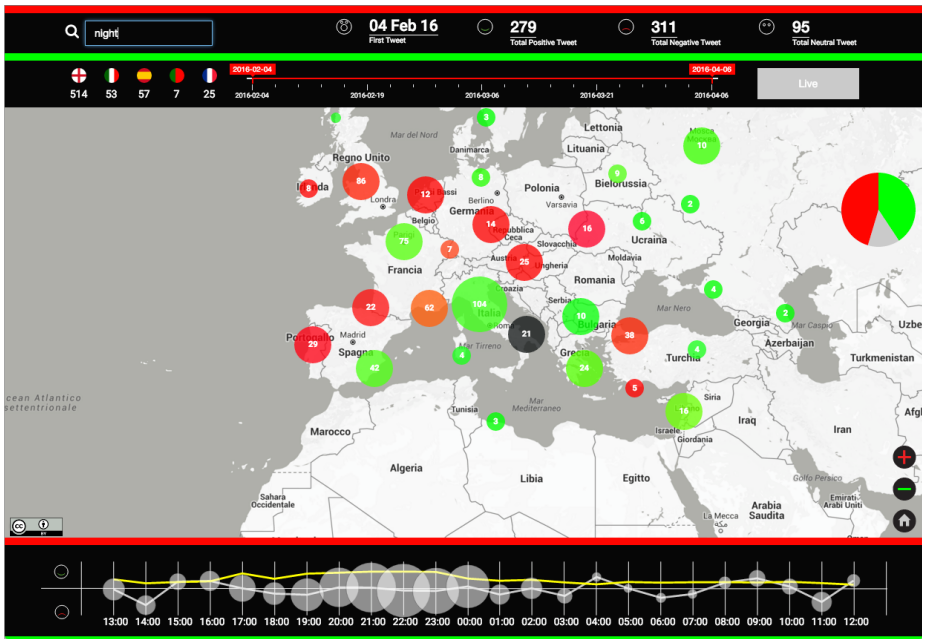


Figura 53: Visualizzazione Sentiment per l'hashtag #night

L'hashtag *#morning* é anch'esso maggiormente positivo anche se a differenza dei precedenti presenta qualche punta negativa. Compare soprattutto nelle ore della mattina comprese tra le 06 e le 13.

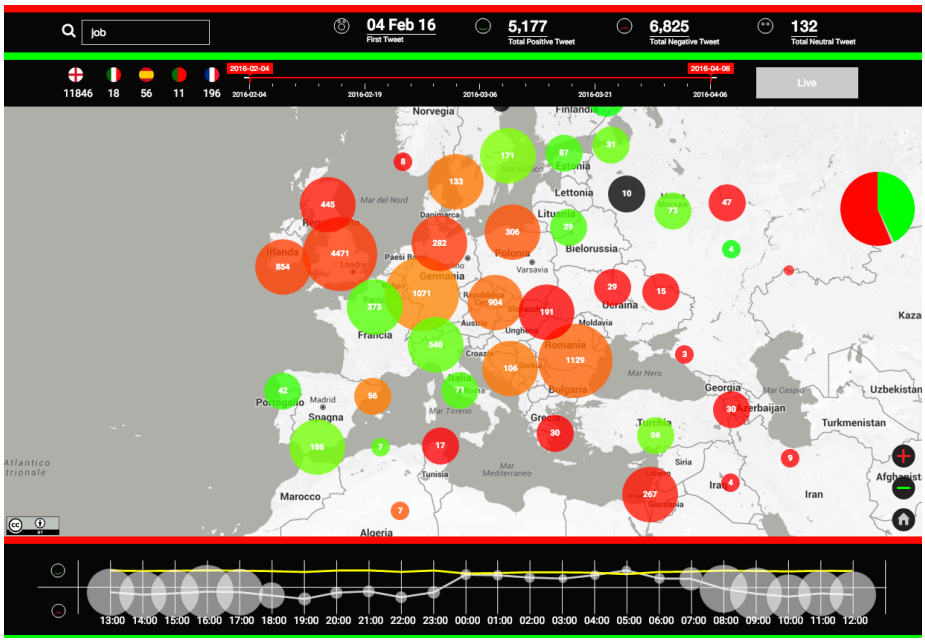


Figura 54: Visualizzazione Sentiment per l'hashtag #job

In figura 53 possiamo vedere l'hashtag *#night* in cui abbiamo una distribuzione che oscilla tra il positivo e il negativo come dimostrano sia la mappa che la timeline. Le ore in cui si utilizza l'hashtag sono quelle notturne in cui appunto abbiamo un alto livello di contraddizione in quanto ci sono tante opinioni positive ma in egual maniera tante opinioni negative.

Un hashtag che risulta particolarmente interessante é *#job* che, come mostrato in figura 54 in quanto é particolarmente utilizzato in tutta Europa infatti all'interno della nostra applicazione tocca più di diecimila tweet. La distribuzione sulle ore del giorno é alta durante l'orario lavorativo quindi partendo dalle ore 07 fino ad arrivare alle ore 18. L'andamento medio mostrato nella timeline dimostra un opinione generale leggermente negativa anche se dalla mappa possiamo notare come in alcune parti dell'Europa sia utilizzato in maniera positiva come ad esempio in Italia. Per questo motivo risulta alto anche il valore di contraddizione.

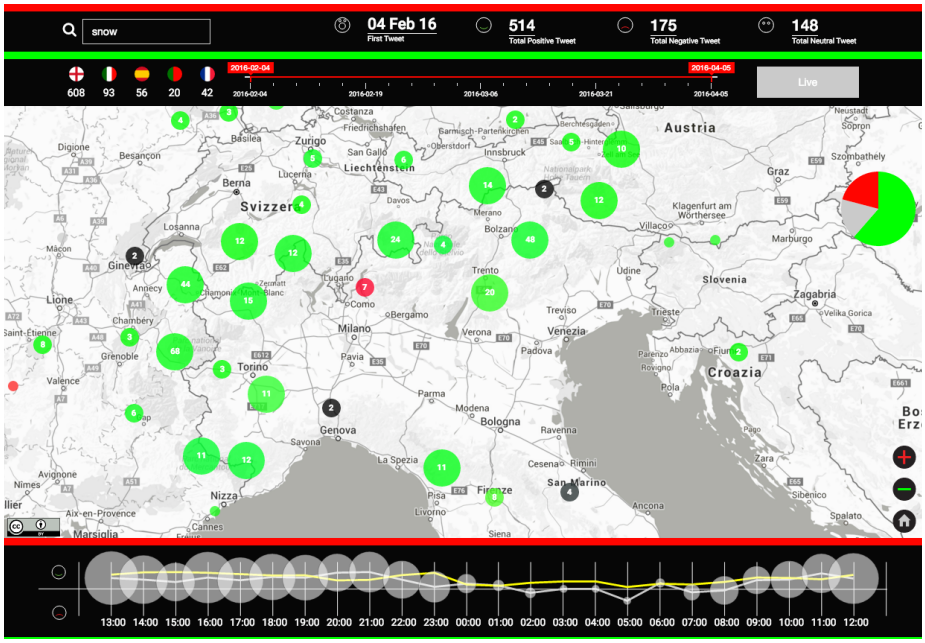


Figura 55: Visualizzazione Sentiment per l'hashtag #snow

Come spiegato nella sezione 7.2.2 cliccando su ogni cerchio visualizzato nella mappa si può effettuare lo zoom automatico scomponendo il cluster in elementi più piccoli e vedendo la distribuzione dell'opinione su aree più circoscritte. In figura 55 viene mostrato l'hashtag *#snow* focalizzando l'attenzione sul nord Italia e mostrando la presenza di un alta distribuzione positiva in corrispondenza delle alpi mentre, quasi nulla nelle zone pianeggianti.

Questi esempi mostrano come, attraverso l'utilizzo di una procedura di Sentiment Analysis, si possano scoprire informazioni interessanti su una grandi quantità di dati. Chiaramente focalizzandosi su aree geografiche più ristrette si potrebbero compiere

operazioni utili a livello commerciale come ad esempio monitorare un brand aziendale o un programma televisivo.

8. CONCLUSIONE E SVILUPPI FUTURI

Il lavoro descritto in questa tesi ha visto l'approfondimento di un innovativo campo di applicazione identificato come Sentiment Analysis il cui obbiettivo é quello di analizzare testi scritti attraverso tecniche di Natural Language Processing. Il fine ultimo di tale approccio é quello di ottenere un valore detto polarità, che riesca ad attribuire connotazione positiva negativa o neutrale al testo oggetto dell'analisi. Tradizionalmente, i lavori sviluppati in questo ambito si concentrano su un'unica lingua di analisi, più frequentemente l'inglese, e su un dominio di applicazione specifico permettendo di ottenere precision e recall elevate.

La mia tesi si inserisce all'interno di questo particolare contesto e presenta una possibile soluzione che permetta di estendere le risorse, fin'ora maggiormente presenti per la sola lingua inglese, sul numero più ampio di lingue possibile senza concentrarsi su un solo specifico dominio, utilizzandole all'interno di un processo di Sentiment Analysis e dimostrando la sua applicabilità tramite il procedimento di valutazione.

La fase iniziale può essere definita come una fase esplorativa in cui é stata ricercata e valutata la letteratura, avendo l'obiettivo di evidenziare aspetti negativi e positivi di ogni metodo proposto, ai fini di comprendere quale fosse il migliore su cui poter lavorare per raggiungere lo scopo finale della ricerca. Questo primo step é stato fondamentale per comprendere quali fossero le varie metodologie legate alla rappresentazione del sentimento di un testo scritto ed é riuscito a dimostrare come la soluzione più semplice sia una classificazione ternaria basata sulla positività, negatività, oggettività. Attraverso le informazioni acquisite nella necessaria fase esplorativa, si é passati alla formulazione del problema che prevede la necessità di effettuare Sentiment Analysis su testi scritti provenienti da domini differenti, nel più alto numero di lingue possibili, e all'interno di analisi di dati in real-time. Per questi motivi si é scelto di approfondire il metodo Lexicon-based. Questo approccio, come detto nella sezione 2.2.3, sfrutta un lessico per ciascuna delle lingue desiderate, che viene fornito in ingresso a un algoritmo rules-based in modo da ottenere un valore di polarità finale relativo ad un testo scritto.

Successivamente si é passati ad una terza fase, rappresentata nei capitoli 4 e 5, necessaria alla progettazione e successiva implementazione di un'architettura software fondamentale

per il raggiungimento degli obiettivi prefissati. In particolare, la prima operazione svolta é stata la creazione di un algoritmo di propagazione che permettesse di trasportare le risorse presenti per la lingua inglese (WordNet, SentiWordNet e Global WordNet) su un dominio multilingue. In particolare sono stati mappati i WordNet provenienti da GlobalWordNet e allineati con la versione inglese Princeton WordNet (PWN) su un dominio legato all'opinione tramite SentiWordNet. Il risultato di questa operazione ha permesso di ottenere uno strumento che, dato un WordNet in una specifica lingua e allineato con PWN, restituisca un lessico composto da una lista di elementi affiancati da un valore di polarità. Oltre a propagare i termini provenienti dai WordNet, tale strumento é in grado di gestire altre categorie di elementi come emoticon ed emoji e di creare un ulteriore lessico contenente parole dette intensificatori, negazioni, e congiunzioni di contrasto. Grazie alla continua ricerca nell'ambito dei database lessicali con susseguente sviluppo di nuove risorse in lingue differenti, é possibile applicare l'algoritmo di propagazione creando nuovi lessici pronti all'uso. Allo stesso tempo essendo le risorse già esistenti in continuo aggiornamento con lo scopo di raffinare la rete semantica al loro interno, ri-applicando il suddetto algoritmo si possono ottenere lessici correttamente aggiornati e di conseguenza migliorati.

Tramite questo algoritmo sono stati ottenuti 17 lessici provenienti da Inglese, Finlandese, Thailandese, Danese, Spagnolo, Arabo, Greco, Croato, Italiano, Olandese, Norvegese, Portoghese, Rumeno, Lituano, Slovacco, Sloveno, Francese che come detto potranno essere estesi ulteriormente, nel caso in cui si sviluppassero nuovi WordNet, in modo facile e veloce. La presenza di questi lessici permette di avere delle risorse legate al sentimento che difficilmente si otterrebbero in altro modo e in maniera così specifica e pertinente tanto da poter permettere ulteriori sviluppi nella ricerca di metodi di Sentiment Analysis più sofisticati ed innovativi.

Per poter utilizzare i lessici nell'analisi di un testo scritto é stato necessario implementare un ulteriore algoritmo chiamato Rules-based. Questo strumento, permette di suddividere un testo in frammenti chiamati token e di cercarne le corrispondenze all'interno del lessico relativo alla specifica lingua di appartenenza. Al suo interno vengono applicate una serie di regole sintattiche e grammaticali che apportano modifiche alle polarità generali nel caso si dovessero incontrare intensificatori, negazioni, e congiunzioni di contrasto. Il risultato di questa operazione permette di ottenere un algoritmo indipendente dalla lingua di applicazione che può quindi essere utilizzato con un qualsiasi lessico purché sia in un formato compatibile. Spesso, nei precedenti lavori, questo non succedeva poiché, essendo sperimentazioni basate su una sola lingua, si utilizzavano funzioni di ottimizzazione troppo specifiche non sfruttabili con lessici multilingue.

Per il funzionamento dello strumento all'interno di applicazioni reali si é progettata e realizzata un'interfaccia Web sfruttando l'architettura REST. In tal modo é quindi possibile effettuare richieste HTTP contenenti testo e lingua da utilizzare nell'algoritmo Lexicon-based e ricevere risposte con il valore di polarità finale calcolato. Questo permette di ottenere un

vero e proprio servizio di Sentiment Analysis totalmente indipendente dall'applicazione che dovrà utilizzarlo. É un risultato utile perché permette di poter apportare miglioramenti continui all'algoritmo Lexicon-based senza dover modificare le piattaforme che sfruttano il servizio.

Per la fase di valutazione sono state estratte recensioni da due domini di applicazione differenti: TripAdvisor e Google Play. Le recensioni permettono di associare ad un testo scritto un valore di opinione numerico che va da 1 a 5 inserito dall'utente al momento della compilazione della recensione. Attraverso la progettazione e l'implementazione di un algoritmo di Web Scraping é stato possibile ottenere dataset per qualunque lingua appartenente ad una nazione in cui esistono le piattaforme di TripAdvisor e Google Play. Il risultato di questo processo permette di costruire delle risorse multilingue associate ad un'opinione annotata da un essere umano. Questo risulta rilevante per due motivi: innanzitutto perché é legato al fatto che diventa possibile valutare strumenti già presenti sul mercato, inoltre perché offre l'opportunità di poter creare nuovi modelli di Machine Learning basati su un approccio Supervised.

Il risultato finale di questo lavoro é quindi uno strumento offerto come servizio Web che, all'interno di un'architettura di analisi di dati real-time, sia in grado di annotare i testi con un valore di polarità. Attraverso la realizzazione di una applicazione Web incentrata su Twitter e in particolare sul meccanismo di hashtag viene mostrato il risultato finale e più importante. Infatti, lo strumento di Sentiment Analysis é stato incorporato all'interno di un architettura Stream Linked Data che, sfruttando tecnologie di Semantic Web, prende in ingresso lo stream offerto dalle API di Twitter e restituisce i singoli tweet decorati con il valore di polarità. Il risultato mostra come questo processo real-time funzioni correttamente e permetta di visualizzare informazioni e statistiche su una grande quantità di dati mostrando come si muovono le opinioni legate agli hashtag su Twitter.

La fase di valutazione mostra come sulle lingue Italiano, Inglese, Francese, Portoghese, Greco si riescano ad ottenere risultati simili e in alcuni casi leggermente superiori rispetto agli strumenti concorrenti come SpazionDati e SentiStrenght. Inoltre, i valori di Precision e Recall rimangono coerenti per tutte le lingue tranne Danese, Arabo, Greco, Norvegese e Polacco in cui ci sono delle discordanze tra Google Play e TripAdvisor, dovute probabilmente alla qualità dei lessici.

Un'osservazione interessante é data dall'incidenza delle categorie di entità analizzate nell'algoritmo rules-based in cui é stato evidenziato come emoji, emoticon e punteggiatura portano ad un aumento sensibile delle prestazioni soprattuto per le classi positive e negative, mentre i booster hanno un incidenza minima soprattutto in Google Play dove i testi sono molto brevi.

Essendo lo strumento progettato per funzionare su domini di applicazione differenti, durante la fase di valutazione è stato importante soffermarsi sui Social Media e in particolare su Twitter. Sono stati annotati, con il servizio di Sentiment Analysis, tweet relativi a un

evento sportivo, in Italiano, Inglese e Spagnolo con lo scopo di ottenere un indice di correttezza dello strumento. Il risultato di questa operazione mostra per tutte le lingue un valore intorno al 50%.

In conclusione, si può quindi affermare che tutti gli obiettivi iniziali siano stati correttamente soddisfatti, anche alla luce dei risultati registrati in fase di valutazione dello strumento. Vale la pena sottolineare come il processo di Sentiment Analysis rappresenta oggi un'area di ricerca in grande sviluppo a seguito della necessità, sempre crescente, di poter valutare le opinioni di grandi quantità di persone. Questi risultati possono essere considerati un punto di partenza da poter approfondire e ottimizzare lungo numerose direzioni.

8.1. Limiti

Nello sviluppo di un progetto così articolato ci si trova spesso a dover far fronte a limitazioni e imperfezioni.

Per prima cosa é opportuno sottolineare il concetto di sentimento o opinione all'interno di un testo scritto e dalla sua rappresentazione attraverso un valore numerico chiamato polarità. Chiaramente, l'utilizzo di un numero o di una sua traduzione in un'etichetta ternaria (positive, negative, neutral) risulta sicuramente un fattore limitante nella valutazione di un campo così vasto e legato alla psicologia umana. In particolare, esistendo così tante sfaccettature di un sentimento nel pensiero di una persona, risulta ancora difficile poterlo individuare correttamente all'interno di un processo automatico, soprattutto se svolto in real-time.

Uno dei problemi più evidenti nello strumento oggetto di questa tesi, ma anche nel generico processo di Sentiment Analysis, é legato alle figure retoriche e in particolare all'ironia. Quest'ultima infatti, permette di esprimere un concetto che é l'esatto opposto del suo significato letterale. Un esempio potrebbe essere la frase "Hai avuto proprio un'idea geniale!" davanti a una decisione che ha portato effetti disastrosi. In questo caso l'utilizzo di tecniche di Sentiment Analysis porterebbero, ad oggi, ad ottenere un valore di polarità positivo anche se in realtà l'intento dell'autore era quello di esprimere un opinione negativa.

Un limite della nostra architettura é dato dalla mancanza di riconoscimento delle entità all'interno della frase. Questo porta ad esempio ad attribuire al testo "il ristorante aveva una splendida atmosfera, ma il cibo era poco e mal cucinato" una connotazione neutrale data dalle somma di una parte positiva (la prima) e di una negativa (la seconda) mentre sarebbe utile riuscire ad identificare le entità "ristorante" e "pietanze" e potergli assegnare un valore di polarità separato.

Un secondo problema é legato alle risorse di partenza, utilizzate per lo sviluppo dei lessici multilingue e, nello specifico, SentiWordNet e Global WordNet. Per quanto riguarda SentiWordNet si é notata la presenza al suo interno di un'alta distribuzione di valori di polarità oggettivi che molto spesso non corrispondono alla vera connotazione delle parole.

Questo elemento, evidenziato anche all'interno di altri lavori legati alla lingua inglese, fa sì che risulti una valutazione dei testi scritti non precisa che conduce all'attribuzione di un valore finale diverso da quello desiderato. Un esempio può essere dato dalla parola italiana "ladro" che tramite SentiWordNet riceve tre punteggi in cui l'oggettività é a 1 mentre positività e negatività sono a 0. Il valore associato nella costruzione del lessico sarà quindi uguale a zero (positività - negatività). Quindi andando ad analizzare la frase "Paolo é un ladro" otterremo un valore finale di polarità uguale a zero in quanto l'unica parola che potrebbe incidere sulla polarità è "ladro" che però ha valore 0, anche se la frase possiede una connotazione altamente negativa. Oltre a questo problema si può affermare che la qualità delle risorse per le lingue come Finlandese, Thailandese, Danese, Arabo, Greco, Sloveno non é sempre alta fornendo reti semantiche incomplete e spesso povere di synset. Questo é dovuto al fatto che lo stato delle ricerche nelle specifiche lingue non risulta spesso in stato avanzato.

8.2. Sviluppi Futuri

Sicuramente un primo possibile sviluppo futuro in uno strumento come questo, che sfrutta i lessici per la computazione di un valore di Sentiment, potrebbe essere legato all'arricchimento dei dizionari, cercando di introdurre forme abbreviate o espressioni comuni di ciascuna lingua ai fini di poter avere prestazioni più alte soprattutto all'interno di domini come Twitter o Google Play.

Altro sviluppo interessante é relativo al campo dell'Opinion Aggregation in cui il Sentiment di un testo non é dato solamente dai valori ottenuti dal lessico, ma proviene da un calcolo più profondo, legato ad un gruppo molto ampio di testi relativi a uno stesso ambito. Questo permetterebbe di trovare parole fondamentali in base, ad esempio, alla loro occorrenza nell'intero set di testi in modo da poter effettuare un'analisi pesata all'interno del singolo testo. In questo modo le parole più vicine ai termini cardine per il dato argomento avrebbero un peso maggiore in termini di polarità finale.

Evoluzione interessante é relativa alla scomposizione del testo in entità fondamentali. Sarebbe importante raffinare la procedura di Tokenization in modo da poter gestire anche le lingue più complesse in cui non sempre il carattere "spazio" definisce la fine e l'inizio di un nuovo token come ad esempio nel Cinese o nel Thailandese. Inoltre, si potrebbe effettuare una procedura di analisi iniziale in cui i testi vengono scomposti in frasi analizzabili separatamente. Infatti, all'interno di un testo ci possono essere molteplici frasi la cui connotazione di opinione può essere differente e calcolata separatamente. Questo tipo di valutazione permetterebbe di effettuare un'operazione di Contraddiction Analysis a livello di singolo testo. Un esempio può essere "Il ristorante aveva una splendida atmosfera. Il cibo era poco e mal cucinato" , in questo testo ci sono due frasi con due connotazioni opposte, analizzandole insieme si può ottenere molto probabilmente un valore vicino al neutrale

mentre analizzate separatamente avremmo un valore altamente positivo per la prima frase e uno altamente negativo per la seconda riuscendo a riscontrare un alto valore di contraddizione.

Nello sviluppo degli strumenti di Sentiment Analysis oggetto di questa tesi, per il dominio di applicazione legato ai social media, é stato approfondito solamente Twitter. Il mondo dei social network é però in continua espansione e sarebbe quindi utile ampliare il campo d'azione su piattaforme diverse com ad esempio Facebook o Instagram, valutandone le differenze e le similitudini. Questo aspetto é valido anche per poter ampliare l'applicazione Web di visualizzazione che sfrutti il meccanismo di hashtag presente anche su le due piattaforme appena citate.

A. TABELLE RISULTATI

Tabella 4: Precision Totali Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	64,06	44,25	7,71
Italiano SPAZIO DATI	67,10	44,37	17,60
Italiano SSTRENGHT	51,80	24,28	40,11
Inglese M14	60,37	44,44	5,66
Inglese SPAZIO DATI	57,74	29,49	22,22
Inglese SSTRENGHT	63,20	24,87	32,07
Francese	48,14	49,52	1,14
Francese SSTRENGHT	40,79	32,21	36,43
Tedesco	43,69	24,78	48,24
Spagnolo	66,12	45,28	7,64
Portoghese	61,95	41,61	13,91
Portoghese SSTRENGHT	54,40	37,81	37,29
Olandese	40,00	41,86	18,17
Danese	35,27	22,43	48,57
Arabo	19,91	14,37	63,85
Greco	21,15	8,75	78,49
Greco SSTRENGHT	36,71	7,43	67,83
Norvegese	21,95	11,82	64,66
Polacco	25,39	25,59	51,39
Thailandese	6,00	9,19	96,55
Sloveno	25,40	26,92	52,41
Belga	13,30	9,74	73,70
Lituano	39,37	16,44	49,75
Finlandese	59,52	32,95	24,73
Rumeno	58,42	37,65	18,70

Tabella 5: Recall Totali Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	46,95	50,49	6,53
Italiano SPAZIO DATI	47,03	55,81	17,61
Italiano SSTRENGHT	48,46	52,67	25,34
Inglese M14	46,71	49,50	4,65
Inglese SPAZIO DATI	24,26	55,40	32,98
Inglese SSTRENGHT	53,17	61,25	22,97
Francese	54,59	46,80	0,71
Francese SSTRENGHT	53,35	46,74	20,10
Tedesco	50,01	54,85	25,72
Spagnolo	46,52	50,76	6,61
Portoghese	48,89	49,39	11,19
Portoghese SSTRENGHT	59,62	63,49	18,94
Olandese	46,72	43,30	11,31
Danese	42,06	47,15	24,96
Arabo	35,30	48,33	27,30
Greco	49,89	54,91	32,02
Greco SSTRENGHT	49,46	57,85	33,47
Norvegese	42,75	46,00	28,31
Polacco	52,76	42,63	24,12
Thailandese	61,17	61,76	33,69
Sloveno	38,41	54,26	23,46
Belga	49,65	42,99	29,68
Lituano	48,36	50,40	27,17
Finlandese	49,23	50,36	18,83
Rumeno	47,87	54,57	13,06

Tabella 6: Precision Totali Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	15,31	8,08	84,66
Italiano SPAZIO DATI	67,10	44,37	17,60
Italiano SSTRENGHT	51,80	24,28	40,11
Inglese M14	15,56	9,67	77,35
Inglese SPAZIO DATI	57,74	29,49	22,22
Inglese SSTRENGHT	63,20	24,87	32,07
Francese	8,23	19,36	75,95
Francese SSTRENGHT	40,79	32,21	36,43
Tedesco	8,94	4,11	91,79
Spagnolo	11,59	6,67	88,24
Portoghese	22,10	5,92	85,49
Portoghese SSTRENGHT	54,40	37,81	37,29
Olandese	2,80	13,51	78,51
Danese	3,22	1,73	94,57
Arabo	0,70	0,17	98,94
Greco	3,84	0,52	97,72
Greco SSTRENGHT	36,71	7,43	67,83
Norvegese	2,28	1,44	98,55
Polacco	3,40	2,95	95,37
Thailandese	0,20	0,00	99,89
Sloveno	1,61	4,31	95,96
Belga	3,66	1,51	95,25
Lituano	5,55	1,69	94,02
Finlandese	10,06	2,34	94,47
Rumeno	6,69	3,47	94,22

Tabella 7: Recall Totali Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	53,28	56,32	32,12
Italiano SPAZIO DATI	47,03	55,81	17,61
Italiano SSTRENGHT	48,46	52,67	25,34
Inglese M14	51,65	50,00	30,00
Inglese SPAZIO DATI	24,26	55,40	32,98
Inglese SSTRENGHT	53,17	61,25	22,97
Francese	57,06	52,82	26,74
Francese SSTRENGHT	53,35	46,74	20,10
Tedesco	54,69	63,63	32,76
Spagnolo	53,72	62,38	32,13
Portoghese	66,83	55,48	34,57
Portoghese SSTRENGHT	59,62	63,49	18,94
Olandese	32,46	54,16	32,56
Danese	40,00	73,07	32,39
Arabo	50,00	66,66	33,21
Greco	62,85	50,00	33,65
Greco SSTRENGHT	49,46	57,85	33,47
Norvegese	48,71	75,00	33,27
Polacco	58,20	52,71	32,65
Thailandese	50,00	0,00	99,89
Sloveno	33,33	84,61	31,90
Belga	50,00	50,00	32,98
Lituano	54,76	58,33	33,05
Finlandese	60,09	76,92	33,95
Rumeno	53,21	63,82	33,00

Tabella 8: Precision Totali TripAdvisor con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	82,93	49,84	0,00
Italiano SPAZIO DATI	76,47	50,00	2,89
Italiano SSTRENGHT	64,82	33,62	29,94
Inglese M14	88,17	50,00	0,00
Inglese SPAZIO DATI	89,07	47,64	4,56
Inglese SSTRENGHT	84,40	39,00	26,16
Francese	68,02	50,00	0,00
Francese SSTRENGHT	45,93	41,04	33,21
Tedesco	73,21	37,78	22,48
Spagnolo	84,27	50,00	0,00
Portoghese	85,62	49,83	0,00
Portoghese SSTRENGHT	71,89	36,29	28,75
Olandese	21,67	50,00	0,00
Danese	62,50	46,55	3,57
Arabo	55,17	44,96	14,94
Greco	47,56	34,61	26,82
Greco SSTRENGHT	50,00	30,58	51,21
Norvegese	51,69	33,33	25,42
Polacco	72,83	46,55	7,40

Tabella 9: Recall Totali TripAdvisor con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	49,33	60,66	0,00
Italiano SPAZIO DATI	25,00	59,13	3,50
Italiano SSTRENGHT	53,58	63,66	19,92
Inglese M14	49,00	64,19	0,00
Inglese SPAZIO DATI	33,47	67,14	6,27
Inglese SSTRENGHT	62,30	74,71	18,29
Francese	58,59	54,74	0,00
Francese SSTRENGHT	61,46	57,39	15,58
Tedesco	52,11	64,45	17,08
Spagnolo	48,95	63,24	0,00
Portoghese	49,52	62,97	0,00
Portoghese SSTRENGHT	65,67	62,71	18,56
Olandese	51,66	42,76	0,00
Danese	43,75	58,33	2,77
Arabo	41,37	45,57	12,03
Greco	44,31	52,94	15,94
Greco SSTRENGHT	50,61	76,47	25,30
Norvegese	42,65	44,15	20,00
Polacco	50,64	55,86	6,25

Tabella 10: Precision Totali TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	49,10	16,46	64,37
Italiano SPAZIO DATI	76,47	50,00	2,89
Italiano SSTRENGHT	64,82	33,62	29,94
Inglese M14	55,91	17,32	56,98
Inglese SPAZIO DATI	89,07	47,64	4,56
Inglese SSTRENGHT	84,40	39,00	26,16
Francese	34,80	28,94	61,48
Francese SSTRENGHT	45,93	41,04	33,21
Tedesco	26,14	6,20	88,11
Spagnolo	46,13	12,69	63,40
Portoghese	59,15	21,01	48,36
Portoghese SSTRENGHT	71,89	36,29	28,75
Olandese	6,29	43,00	38,46
Danese	12,50	10,16	85,71
Arabo	1,72	1,70	98,85
Greco	13,41	3,84	92,68
Greco SSTRENGHT	50,00	30,58	51,21
Norvegese	7,62	2,58	84,74
Polacco	25,92	12,72	82,71

Tabella 11: Recall Totali TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano M14	61,88	72,54	33,28
Italiano SPAZIO DATI	25,00	59,13	3,50
Italiano SSTRENGHT	53,58	63,66	19,92
Inglese M14	58,53	75,40	32,78
Inglese SPAZIO DATI	33,47	67,14	6,27
Inglese SSTRENGHT	62,30	74,71	18,29
Francese	68,88	63,93	24,75
Francese SSTRENGHT	61,46	57,39	15,58
Tedesco	72,77	81,17	35,65
Spagnolo	63,02	75,75	32,71
Portoghese	56,56	73,80	29,36
Portoghese SSTRENGHT	65,67	62,71	18,56
Olandese	64,28	46,65	14,47
Danese	50,00	100,00	30,37
Arabo	60,00	75,00	33,20
Greco	57,89	37,50	34,86
Greco SSTRENGHT	50,61	76,47	25,30
Norvegese	40,90	50,00	31,25
Polacco	58,57	77,77	32,52

Tabella 12: Precision Booster Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	74,70	48,65	2,78
Italiano No Booster	71,69	49,85	0,34
Inglese Booster	63,80	50,00	5,50
Inglese No Booster	63,88	50,00	0,00
Francese Booster	51,41	49,92	0,13
Francese No Booster	48,02	49,95	0,00
Spagnolo Booster	75,38	48,77	1,51
Spagnolo No Booster	76,79	49,61	0,35
Portoghese Booster	73,28	44,56	6,35
Portoghese No Booster	64,51	48,30	5,78

Tabella 13: Recall Booster Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	51,34	45,19	3,70
Italiano No Booster	56,00	46,25	0,38
Inglese Booster	38,83	62,50	4,33
Inglese No Booster	41,81	55,00	0,00
Francese Booster	51,34	45,68	0,09
Francese No Booster	59,16	46,38	0,00
Spagnolo Booster	52,13	47,77	1,77
Spagnolo No Booster	56,66	49,28	0,36
Portoghese Booster	46,50	43,86	10,00
Portoghese No Booster	49,77	45,96	6,59

Tabella 14: Precision Booster Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	30,30	13,34	70,73
Italiano No Booster	18,61	17,32	75,08
Inglese Booster	33,33	13,88	38,88
Inglese No Booster	30,55	17,50	77,77
Francese Booster	51,41	49,92	0,13
Francese No Booster	48,02	49,95	0,00
Spagnolo Booster	23,36	11,18	77,34
Spagnolo No Booster	16,19	12,39	82,71
Portoghese Booster	32,31	11,26	76,30
Portoghese No Booster	13,81	13,26	87,09

Tabella 15: Recall Booster Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	56,01	52,85	31,20
Italiano No Booster	57,03	53,20	28,82
Inglese Booster	37,50	55,55	21,21
Inglese No Booster	62,16	65,71	32,54
Francese Booster	51,34	45,68	0,09
Francese No Booster	59,16	46,38	0,00
Spagnolo Booster	58,49	57,92	30,17
Spagnolo No Booster	65,63	60,09	29,39
Portoghese Booster	58,47	61,47	38,67
Portoghese No Booster	62,10	61,34	36,10

Tabella 16: Precision Booster TripAdvisor con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	84,38	50,00	0,00
Italiano No Booster	81,49	50,00	0,00
Inglese Booster	89,82	50,00	0,00
Inglese No Booster	85,60	50,00	0,00
Francese Booster	67,80	50,00	0,00
Francese No Booster	61,62	50,00	0,00
Spagnolo Booster	84,73	50,00	0,00
Spagnolo No Booster	89,31	50,00	0,00
Portoghese Booster	86,63	50,00	0,00
Portoghese No Booster	78,34	50,00	0,00

Tabella 17: Recall Booster TripAdvisor con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	52,38	52,01	0,00
Italiano No Booster	55,00	50,00	0,00
Inglese Booster	51,71	62,66	0,00
Inglese No Booster	54,67	57,61	0,00
Francese Booster	60,57	48,20	0,00
Francese No Booster	65,68	46,80	0,00
Spagnolo Booster	51,86	51,42	0,00
Spagnolo No Booster	54,92	59,85	0,00
Portoghese Booster	52,80	63,90	0,00
Portoghese No Booster	57,62	62,88	0,00

Tabella 18: Precision Booster TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	56,97	18,72	62,20
Italiano No Booster	49,04	22,56	66,50
Inglese Booster	63,52	17,49	54,49
Inglese No Booster	52,35	20,00	58,73
Francese Booster	37,52	30,53	56,75
Francese No Booster	19,56	29,83	67,11
Spagnolo Booster	52,67	13,69	53,07
Spagnolo No Booster	52,67	15,13	67,69
Portoghese Booster	63,13	21,71	39,02
Portoghese No Booster	51,15	27,23	45,12

Tabella 19: Recall Booster TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Booster	64,05	70,66	37,14
Italiano No Booster	68,91	70,47	33,73
Inglese Booster	60,66	74,64	34,79
Inglese No Booster	66,14	67,28	30,57
Francese Booster	69,56	59,16	24,27
Francese No Booster	66,66	61,44	23,68
Spagnolo Booster	62,16	65,71	32,54
Spagnolo No Booster	70,76	75,67	37,44
Portoghese Booster	58,79	74,75	22,22
Portoghese No Booster	68,09	73,56	18,87

Tabella 20: Precision Punteggiatura GooglPlay con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Punctuation	35,47	20,24	63,13
Italiano No Punctuation	13,43	5,04	87,57
Inglese Punctuation	29,85	25,00	48,57
Inglese No Punctuation	20,89	14,04	77,14
Francese Punctuation	53,13	37,81	36,92
Francese No Punctuation	32,50	20,66	61,53
Spagnolo Punctuation	34,76	20,59	63,54
Spagnolo No	12,71	6,80	89,03
Portoghese Punctuation	42,83	18,16	61,11
Portoghese No	17,85	4,45	87,87

Tabella 21: Recall Punteggiatura GooglPlay con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Punctuation	50,58	57,47	27,98
Italiano No Punctuation	58,63	55,84	31,87
Inglese Punctuation	48,78	51,72	22,97
Inglese No Punctuation	56,00	50,00	31,76
Francese Punctuation	63,90	66,24	13,63
Francese No Punctuation	73,23	61,72	21,62
Spagnolo Punctuation	48,31	65,35	29,27
Spagnolo No	55,73	62,79	35,33
Portoghese Punctuation	64,38	52,72	25,63
Portoghese No	70,50	56,75	27,35

Tabella 22: Precision Punteggiatura TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Punctuation	67,42	32,97	40,74
Italiano No Punctuation	49,71	15,10	68,51
Inglese Punctuation	70,64	31,03	41,86
Inglese No Punctuation	54,12	20,83	69,76
Francese Punctuation	53,13	37,81	36,93
Francese No Punctuation	32,50	20,66	61,53
Spagnolo Punctuation	78,57	28,00	30,00
Spagnolo No	52,38	11,53	30,00
Portoghese Punctuation	70,76	35,71	50,00
Portoghese No	61,53	17,24	50,00

Tabella 23: Recall Punteggiatura TripAdvisor con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Punctuation	60,51	74,69	17,05
Italiano No Punctuation	73,72	78,37	24,83
Inglese Punctuation	59,23	80,00	21,68
Inglese No Punctuation	66,29	89,28	28,84
Francese Punctuation	63,90	66,24	13,63
Francese No Punctuation	73,23	61,72	21,62
Spagnolo Punctuation	66,00	93,33	11,53
Spagnolo No	75,86	85,71	10,34
Portoghese Punctuation	66,66	80,00	14,63
Portoghese No	49,77	45,96	6,59

Tabella 24: Precision emoji Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Emoji	89,61	50,00	0,00
Italiano No Emoji	60,00	45,14	13,15
Francese Emoji	80,57	50,00	0,00
Francese No Emoji	50,31	49,12	0,00
Spagnolo Emoji	90,79	49,81	0,00
Spagnolo No Emoji	66,94	45,85	6,14
Portoghese Emoji	86,66	50,00	0,00
Portoghese No Emoji	53,33	40,74	17,77
Italiano Emoji	89,61	50,00	0,00
Italiano No Emoji	60,00	45,14	13,15

Tabella 25: Recall emoji Google Play con soglia 0

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Emoji	68,32	64,55	0,00
Italiano No Emoji	58,20	43,64	8,92
Francese Emoji	69,31	52,45	0,00
Francese No Emoji	67,23	36,52	0,00
Spagnolo Emoji	57,71	69,38	0,00
Spagnolo No Emoji	46,51	51,36	5,69
Portoghese Emoji	54,54	67,02	0,00
Portoghese No Emoji	45,28	53,01	11,11
Italiano Emoji	68,32	64,55	0,00
Italiano No Emoji	58,20	43,64	8,92

Tabella 26: Precision emoji Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Emoji	61,92	27,17	53,94
Italiano No Emoji	20,38	15,56	75,00
Francese Emoji	48,08	30,40	55,00
Francese No Emoji	10,82	23,77	69,16
Spagnolo Emoji	60,66	19,31	64,91
Spagnolo No Emoji	17,57	8,40	82,45
Portoghese Emoji	63,33	14,81	62,22
Portoghese No Emoji	25,55	3,96	80,00
Italiano Emoji	61,92	27,17	53,94
Italiano No Emoji	20,38	15,56	75,00

Tabella 27: Recall emoji Google Play con soglia 0.25

Lingua	Positive (%)	Negative (%)	Neutral (%)
Italiano Emoji	75,23	70,42	22,77
Italiano No Emoji	58,88	56,52	20,80
Francese Emoji	75,12	62,29	23,15
Francese No Emoji	70,83	42,96	22,01
Spagnolo Emoji	68,72	76,27	36,09
Spagnolo No Emoji	60,00	61,29	31,02
Portoghese Emoji	73,07	72,72	36,36
Portoghese No Emoji	67,64	44,44	34,28
Italiano Emoji	75,23	70,42	22,77
Italiano No Emoji	58,88	56,52	20,80

Riferimenti bibliografici

[1] Altrabsheh, N., Gaber, M., & Cocea, M. (2013). SA-E: sentiment analysis for education. Paper presented at the 5th KES International Conference on Intelligent Decision Technologies.

[2] Andrea Esuli, & Sebastiani, F. (2006). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of LREC. Vol. 6. 2006

[3] Balduini, M., Della Valle, E., Dell'Aglio, D., Tsytsarau, M., Palpanas, T., & Confalonieri, C. (2013). Social Listening of City Scale Events Using the Streaming Linked Data Framework. Proceedings of The Semantic Web–ISWC 2013. Springer Berlin Heidelberg, 2013

[4] Banea, C., Mihalcea, R., & Wiebe, J. (2011). Multilingual sentiment and subjectivity analysis. Multilingual natural language processing, 6, 1-19.

[5] Basile, V., & Nissim, M. (2013). Sentiment analysis on Italian tweets. Paper presented at the Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.

[6] Bond, F., & Foster, R. (2013). Linking and Extending an Open Multilingual Wordnet. Paper presented at the ACL (1).

[7] Claudia Soria, Monica Monachini, & Vossen, P. (2009). Wordnet-LMF: Fleshing out a Standardized Format for Wordnet Interoperability. Proceedings of the 2009 international workshop on Intercultural collaboration. ACM, 2009.

[8] De Smedt, T., & Daelemans, W. (2012). Pattern for python. The Journal of Machine Learning Research, 13(1), 2063-2067.

[9] Denecke, K. (2008). Using sentiWordNet for multilingual sentiment analysis. Paper presented at the Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on.

[10] Derczynski, L., Ritter, A., Clark, S., & Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. Proceedings of RANLP.

[11] Diddy, E. D. (1998). Enhanced text retrieval using natural language processing. Bulletin of the American Society for Information Science.

[12] Ekman, P., Friesen, W., & Ellsworth, P. (1982). What emotion categories or dimensions can observers judge from facial behaviour? In, P. Ekman. Emotion in the Human Face.

[13] Fielding, R. T., & Taylor, R. N. (2000). Principled design of the modern Web architecture. ACM Transactions on Internet Technology (TOIT) 2.2 (2002): 115-150

[14] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1, 12.

[15] Greaves, F., Ramirez-Cano, D., Millett, C., Darzi, A., & Donaldson, L. (2013). Use of sentiment analysis for capturing patient experience from free-text comments posted online. Journal of medical Internet research, 15(11), e239.

[16] Hiroshi, K., Tetsuya, N., & Hideo, W. (2004). Deeper sentiment analysis using machine translation technology. Proceedings of the 20th international conference on Computational Linguistics.

[17] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

[18] Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of Eighth International AAAI Conference on Weblogs and Social Media. 2014.

[19] Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., & Kanzaki, K. (2008). Development of the Japanese WordNet. Paper presented at the LREC.

[20] Kasper, W., & Vela, M. (2011). Sentiment analysis for hotel reviews. Paper presented at the Computational linguistics-applications conference.

[21] Kerstin Denecke, ikalai Tsytsara, & Palpanas, T. (2009). Topic-related Sentiment Analysis for Discovering Contradicting Opinions in Weblogs.

[22] Magnini, B., Strapparava, C., Ciravegna, F., & Pianta, E. (1994). A Project for the Constraction of an Italian Lexical Knowledge Base in the Framework of Wordnet: Istituto per la Ricerca Scientifica e Tecnologica.

[23] Martin, J. H., & Jurafsky, D. (2000). Speech and language processing. International Edition.

[24] Maynard, D., & Funk, A. (2011). Automatic detection of political opinions in tweets. Paper presented at the The semantic web: ESWC 2011 workshops.

[25] Miller, G. A. (1995). WordNet: A Lexical Database for English. Commun. ACM, 38(11), 39-41. doi:10.1145/219717.219748

[26] Miselli, D., & Rasi, R. ONTOLOGIE LESSICALI MULTILINGUA: MULTIWORDNET ED EUROWORDNET.

[27] Novak, P. K., Smailovic, J., Sluban, B., & Mozetic, I. (2015). Sentiment of Emojis. CoRR, abs/1509.07761.

[28] Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Paper presented at the LREc.

[29] Pang, B., & Lee, L. (2007). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135. doi:10.1561/1500000011

[30] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. CoRR, cs.CL/0205070.

[31] Richardson, L., & Ruby, S. (2007). Restful web services. " O'Reilly Media, Inc.".

[32] Saravanakumar, M., & SuganthaLakshmi, T. (2012). Social media marketing. Life Science Journal, 9(4), 4444-4451.

[33] Thelwall, M. (2013). Heart and soul: Sentiment strength detection in the social web with sentistrength. Proceedings of the CyberEmotions, 1-14.

[34] Tsytsarau, M., & Palpanas, T. (2012). Survey on mining subjective data on the web. Data Mining and Knowledge Discovery 24.3 (2012): 478-514.

[35] Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics.

[36] Zhang, J., Kawai, Y., Kumamoto, T., & Tanaka, K. (2009). A novel visualization method for distinction of web news sentiment. (pp. 181-194). Springer Berlin Heidelberg.

[37] Altrabsheh, N., Gaber, M., & Cocea, M. (2013). SA-E: sentiment analysis for education. Proceedings of the 5th KES International Conference on Intelligent Decision Technologies.