Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Informatica

Due approcci alla *sentiment polarity classification* di *tweet* per la lingua italiana

Relatore:
Chiar.mo Prof.
FABIO TAMBURINI

Presentata da:
PIERLUIGI DI GENNARO

Sessione III
Anno Accademico 2015-2016

"L'uomo deve perseverare nell'idea che
l'incomprensibile sia comprensibile;
altrimenti rinuncerebbe a cercare."
J. W. Goethe

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

"E non ci si rende abbastanza conto che, se in Italia non verranno sviluppate le ricerche sulle tecnologie sulla lingua - soprattutto il Trattamento Automatico del Linguaggio - la lingua italiana è destinata a diventare sempre più marginale, fin quasi a scomparire. Se questa è la cattiva notizia, la buona notizia è che il TAL, nel mondo della ricerca italiano, gode di molte attenzioni."

"The field of humanities and journalism is strongly debating, at times improperly, the need to defend Italian against often unavoidable Anglicisms and against the threat deriving from the use of new technologies, such as SMSs. The community doesn't realize that the Italian language is destined to become ever more marginal, and finally disappear if research in new language technologies - in particular, research in Natural Language Processing - is not pursued. This is the bad news. The good news is that Italian research in Natural Language Processing enjoys considerable attention, as this White Paper shows."
Giordano Bruno Guerri (Presidente della fondazione "Il Vittoriale degli Italiani")

Da ormai qualche decennio, è in corso una rivoluzione digitale che sta avendo un impatto radicale sulla comunicazione e sulla società dei nostri giorni. Con l'affermarsi del World Wide Web e la crescita esponenziale di siti e applicazioni Web 2.0 che rendono sempre più facile la comunicazione e collaborazione tra le persone, il modo di comportarsi e di relazionarsi nella propria vita sociale e, più in generale, nella società è cambiato drasticamente.

Le persone esprimono sempre più le proprie opinioni, punti di vista, sentimenti sui profili *social*, sui blog, sui forum e trovano più comodo (e si

sentono più protette) nel chiedere e dare consigli e pareri online sugli argomenti più disparati: politica, medicina e salute, cronaca, gossip, prodotti ed eventi artistici, etc... Questa enorme quantità di informazioni ha offerto l'opportunità di sviluppare teorie e tecnologie per l'elaborazione automatica del linguaggio naturale su una quantità di dati mai vista in precedenza nell'era dell'informazione.

L'analisi del sentimento (*sentiment analysis*) è uno dei campi degli studi computazionali che affronta i problemi dell'elaborazione automatica del linguaggio naturale legati all'estrazione e alla classificazione di opinioni e rientra tra gli ambiti di ricerca che hanno beneficiato maggiormente di questo processo di rivoluzione digitale: identificare, estrarre e classificare tutte le opinioni espresse dalle persone è un attività manualmente erculea ed un processo automatico sembra esserne proprio la soluzione migliore.

Purtroppo, nonostante i considerevoli passi avanti degli ultimi anni, il ritmo tenuto dal progresso tecnologico nell'ambito della trattazione automatica del linguaggio è ancora troppo lento: correttori ortografici e grammaticali sono ancora prevalentemente monolingue, i servizi di traduzione automatica online, sebbene utili per approssimare rapidamente il contenuto di un documento, sono ancora poco accurati, le applicazioni di analisi, comprensione e riassunto automatico di testi sono ancora lontane dal poter essere usate in modo affidabile nella realtà. Tutto ciò, principalmente, a causa della complessità intrinseca del linguaggio umano: modellizzare una lingua in modo automatico è un processo lungo e costoso e, inoltre, non è uguale per tutte le lingue.

L'inglese, seconda solo al cinese mandarino e allo spagnolo tra le lingue più parlate al mondo (Lewis *et al.*, 2016), è stata ed è ancora oggi la lingua più presente nei documenti online e ciò ne ha favorito notevolmente gli studi di ricerca nell'ambito dell'elaborazione del linguaggio naturale di testi scritti quali la *question answering*, l'*information extraction*, la *text classification* ed anche la *sentiment analysis*. Per la lingua italiana, invece, nonostante sia tra le 25 lingue più parlate al mondo con 63 milioni di parlanti nativi e 125

milioni di persone che la usano come seconda lingua, la ricerca nelle tecnologie automatiche legate all'elaborazione di testi è molto meno sviluppata di quella per la lingua inglese. Come risultato, vi sono meno corpora annotati o risorse specifiche e spesso si è costretti ad utilizzare, per il trattamento automatico, strumenti e software nati per la lingua inglese previa opportuna traduzione. Ma l'italiano e l'inglese sono estremamente diverse tra loro ed un modello costruito per l'una non può essere utilizzato per l'altra lingua se non accettando un elevato grado di inaccuratezza.

Il professor Giordano Bruno Guerri, presidente della fondazione "Il Vittoriale degli Italiani" che si occupa, tra le tante attività, di promuovere e diffondere le opere di Gabriele D'Annunzio, afferma come oramai il trattamento automatico di una lingua sia diventato importante al punto tale che, qualora una lingua ne fosse sprovvista o carente, ciò ne comporterebbe persino la scomparsa.

Questo lavoro di tesi vuole essere un'altra piccola "attenzione" fra le tante che, come sostiene lo stesso Guerri, il mondo della ricerca italiano riserva al trattamento automatico della nostra lingua.

Organizzazione del lavoro di tesi

Questo lavoro di tesi si pone l'obiettivo di fornire un'ampia panoramica sull'attuale stato dell'arte della ricerca sulla *sentiment analysis* mostrandone le metodologie, le tecniche e le applicazioni realizzate negli ultimi anni e di presentare un'implementazione concreta di due diversi sistemi per la *sentiment polarity classification* di *tweet* in lingua italiana, il primo (*FICLIT+CS@Unibo System*) utilizzando un approccio basato sull'orientamento semantico ed il secondo con un approccio di tipo *machine learning*.

FICLIT+CS@Unibo System (Di Gennaro, Rossi, Tamburini, 2014) è stato presentato alla campagna di valutazione di EVALITA 2014 per il task SENTImentPOlarityClassification (SENTIPOLC), in un lavoro di ricerca congiunto con il prof. Fabio Tamburini dell'Università di Bologna ed Arianna Rossi, al-

lora studentessa del Dipartimento di Filologia Classica e Italianistica (Rossi, 2014). Si basa su un approccio legato all'identificazione di *opinion word*, precedentemente etichettate per orientamento semantico, ed una classificazione dei *tweet* basata sull'aggregazione della polarità di singole parole.

Il secondo sistema nasce da un processo di apprendimento supervisionato basato su un classificatore di tipo *Support Vector Machine* (SVM). L'algoritmo sviluppato estrae, per ogni *tweet*, un vettore di *feature* sintattiche, semantiche e di polarità che, opportunamente somministrate ad un classificatore SVM, permettono di costruire un modello generale per la *sentiment polarity classification*.

Il lavoro di tesi è così organizzato:

Capitolo 1 Rapida panoramica sulla *sentiment analysis*: metodologie, tecniche, classificazioni ed uno sguardo alla ricerca per la lingua italiana

Capitolo 2 Approfondimento delle tecniche per la *sentiment analysis* basate sul lessico

Capitolo 3 Approfondimento delle tecniche per la *sentiment analysis* basate su algoritmi di *machine learning*

Capitolo 4 Panoramica sulla *Twitter sentiment polarity classification* ed introduzione ai sistemi realizzati per questo lavoro di tesi

Capitolo 5 Presentazione approfondita dell'implementazione del sistema basato sul lessico (*FICLIT+CS@Unibo System*) presentato alla campagna di valutazione EVALITA 2014

Capitolo 6 Presentazione approfondita dell'implementazione del sistema basato su algoritmi di *machine learning*

Capitolo 7 Panoramica sulle metriche di valutazione utilizzate ed illustrazione dei risultati ottenuti dai sistemi realizzati per questo lavoro di tesi

Conclusioni e sviluppi futuri

Capitolo 1

Sentiment analysis

La *sentiment analysis* (SA, analisi del sentimento) rientra nel contesto più generale dell'elaborazione del linguaggio naturale, della linguistica computazionale e dell'analisi testuale e si occupa di identificare ed estrarre, in modo automatico, opinioni, sentimenti ed emozioni contenute in un qualsiasi documento testuale, sia esso un articolo di giornale, una recensione, un commento o un post su *social network*, *microblog*, *blog*, *forum*.

L'obiettivo della *sentiment analysis* è, quindi, quello di identificare la presenza e classificare le opinioni espresse su un determinato soggetto, oggetto o argomento o, più in generale, nell'intero documento.

1.1 Opinione e sentimento

Si osservino le seguenti definizioni di opinione e sentimento:

Opinione ciò che si pensa di qualcuno o di qualcosa; idea, parere, giudizio[1]

Sentimento ogni stato affettivo della coscienza, di segno positivo o negativo; ogni moto soggettivo dell'animo che dia una particolare tonalità affettiva alle nostre sensazioni, rappresentazioni, idee[2]

[1] http://www.garzantilinguistica.it/ricerca/?q=opinione

[2] http://www.garzantilinguistica.it/ricerca/?q=sentimento

Tali definizioni, seppur linguisticamente molto differenti tra loro, sono identiche dal punto di vista della *sentiment analysis*, poiché l'attenzione è focalizzata esclusivamente sulla polarità dell'emozione, del giudizio, del punto di vista espresso, indipendentemente che esso sia una opinione o un sentimento. Per tale ragione, d'ora in poi, verranno utilizzati indistintamente l'uno o l'altro termine.

Una definizione formale di opinione è stata fornita da (Liu, Zang, 2012). Essi rappresentano una opinione come una quintupla

$$\text{opinione} = (\mathrm{e_j},\ \mathrm{a_{jk}},\ \mathrm{oo_{ijkl}},\ \mathrm{h_i},\ \mathrm{t_l})$$

ove:

$\mathbf{e_j}$ rappresenta un'entità cioè l'oggetto su cui si sta esprimendo l'opinione, sia esso una persona, un evento, un argomento, un prodotto. L'entità può avere diverse proprietà che la caratterizzano ed essere composta da altri componenti (sotto-entità) che, a loro volta, possono avere proprietà e componenti. Formalmente, quindi, ad ogni evento può esser associata una coppia (C, A) dove C è l'insieme di tutti i suoi componenti mentre A è l'insieme di tutte le sue proprietà

$\mathbf{a_{jk}}$ è la proprietà (o componente) k dell'entità e_j, rispetto alla quale è espressa l'opinione

$\mathbf{h_i}$ è l'autore dell'opinione

$\mathbf{t_l}$ è l'istante temporale in cui h_i ha espresso la propria opinione

$\mathbf{oo_{ijkl}}$ rappresenta l'orientamento (polarità) dell'opinione sulla proprietà a_{jk} dell'entità e_j al tempo t_l da parte di h_i. Tale orientamento può essere positivo o negativo e può avere diversi livelli di intensità. Ad esempio un'opinione positiva può variare tra felicità ed estasi o una negativa tra sconforto e disperazione

La parte più difficile nella definizione di una quintupla di (Liu, Zang, 2012) è determinare l'orientamento dell'opinione (oo_{ijkl}) poiché la classificazione

della polarità di un qualsiasi testo è un'attività spesso molto soggettiva: una frase può essere etichettata positiva da alcuni ma neutra da altri, in base ad una scala di valori del tutto personale.

A questo punto, è importante osservare come, dal punto di vista formale, le opinioni possano essere espresse in modo diretto o indiretto. Tale diversità si rivela essere estremamente rilevante ai fini di una corretta analisi del sentimento.

Una opinione è detta "diretta" se esprime esplicitamente un giudizio su un'entità (o su una delle sue proprietà). La definizione di (Liu, Zang, 2012) può essere applicata alle opinioni dirette, come nel seguente esempio:

La batteria di questo telefono dura troppo poco.

$\mathbf{e_j}$ telefono

$\mathbf{a_{jk}}$ batteria

$\mathbf{h_i}$ oggi

$\mathbf{t_l}$ autore

$\mathbf{oo_{ijkl}}$ *negativo*

Tale definizione non è però applicabile alle opinioni "indirette" poiché, solitamente, coinvolgono due o più entità. Una opinione è detta indiretta, infatti, se esprime implicitamente un giudizio su un'entità (o su una sua proprietà) tramite il confronto con un'altra entità (o una sua proprietà). Ad esempio:

La batteria degli iPhone dura di più di quella degli smartphone Samsung.

Nonostante i problemi di soggettività, generalmente, una opinione diretta è più o meno facilmente etichettabile in una sola classe di polarità: positiva, negativa o neutra. Le opinioni indirette, invece, si classificano diversamente

a seconda dell'entità d'interesse coinvolta nel confronto e, ovviamente, ciò rappresenta un problema nella determinazione della polarità di una intera frase o documento. Le entità coinvolte, infatti, possono essere poste in similitudine o in contrapposizione: un oggetto A può essere buono tanto quanto un oggetto B e, in altri casi, l'oggetto A è migliore dell'oggetto B o viceversa. Nel primo caso l'opinione è positiva per entrambe le entità ma negli altri casi l'opinione espressa è esattamente in contrapposizione. Dedurre la polarità generale di opinioni indirette risulta essere, quindi, un problema alquanto difficile.

D'ora in poi, sfruttando le definizioni di (Liu, Zang, 2012), si utilizzerà il termine entità per riferirsi all'oggetto su cui è espressa l'opinione.

1.2 Le difficoltà della *sentiment analysis*

Diverse ricerche dimostrano come la *sentiment analysis* risulti essere più difficile di un tradizionale problema di *data mining*, quale, ad esempio, la *topic-based classification*, nonostante il numero di classi genericamente inferiore, solitamente solo due o tre (Pang, Lee, 2008). Una delle difficoltà sta nella sottilissima distinzione che spesso esiste tra sentimento positivo e negativo (distinzione che, come già affermato in precedenza, risulta difficile anche per un essere umano) ma il principale motivo tale per cui la *sentiment analysis* risulta essere più difficile di un qualsiasi problema di *topic detection* è che quest'ultimo può essere risolto con il solo utilizzo di *keyword* (parole chiave) che, purtroppo, non funzionano altrettanto bene per l'analisi del sentimento (Turney, 2005).

Non sempre le opinioni sono espresse solamente tramite l'uso di *opinion words* (parole intrinsecamente polarizzate); in molti casi il sentimento è espresso in modo indiretto o tramite artifici linguistici quali metafore o altre figure retoriche, o con l'uso di espressioni informali e gerchi non appartenenti al vocabolario linguistico, o con l'uso di ironia e sarcasmo, dove l'interpretazione del significato è strettamente soggettiva.

Ulteriori problematiche sono dovute alle ambiguità sintattiche, alla difficoltà di determinare la soggettività/oggettività di frasi e testi e alla difficoltà di evincere il dominio trattato: si osserva, infatti, come alcune opinioni siano strettamente dipendenti dal contesto e dall'ambito di riferimento. Aggettivi e avverbi che risultano essere positivi per alcune entità possono risultare negativi per altre; frasi come "Si è spenta!" può esprimere un'opinione positiva se si sta parlando di una fastidiosa sveglia del lunedì mattina ma molto negativa se si tratta di una televisione che non si accende il giorno della finale dei mondiali.

1.3 *Sentiment analysis* come problema di classificazione

Come finora osservato, seppur più difficile, l'analisi del sentimento può essere ricondotta ad un problema di classificazione *topic-based* a due, tre o più classi.

La letteratura, ad oggi, si è concentrata particolarmente sui seguenti problemi di classificazione legati alla *sentiment analysis*:

- Soggettività/oggettività (*Subjectivity classification*)
- Polarità di testi soggettivi (*Sentiment polarity classification*)
- Estrazione dell'autore dell'opinione (*Opinion holder extraction*)
- Estrazione delle entità/proprietà soggette ad opinione (*Object/feature extraction*)

1.3.1 *Subjectivity classification*

Solitamente, un qualsiasi documento testuale contiene informazioni di due tipi: "fatti" e "opinioni". I "fatti" sono frasi oggettive circa un determinato soggetto, oggetto, evento o argomento; le "opinioni", invece, sono frasi

soggettive che descrivono, appunto, opinioni, giudizi, punti di vista, emozioni. La *subjectivity classification* si occupa di classificare un documenti in due classi: "oggettivo" e "soggettivo" in base ai "fatti" e alle "opinioni" ivi contenute.

(Tang *et al.*, 2002) hanno formalizzato la *subjectivity classification* come segue: sia $S = \{s_1,...,s_n\}$ l'insieme delle frasi di un documento D, l'obiettivo della *subjectivity classification* è di distinguere le frasi usate per esprimere"opinioni" (che formeranno l'insieme S_s) da quelle usate per descrivere "fatti" (che formeranno l'insieme S_o), ove $S_s \cup S_o = S$.

1.3.2 *Sentiment polarity classification*

Il problema della *sentiment polarity classification* consiste nell'identificare l'orientamento dell'opinione delle frasi soggettive che compongono un documento al fine di ottenerne una classificazione globale.

Generalmente, la classificazione può essere binaria ("positivo", "negativo") o ternaria ("positivo", "negativo", "neutro") o anche multiclasse ("positivo", "negativo", "neutro", "misto") a seconda del livello di dettaglio a cui si è interessati.

1.3.3 *Opinion holder extraction*

Tra i task della *sentiment analysis* rientra anche l'*opinion holder extraction* che si occupa di riconoscere l'autore e le fonti dirette ed indirette delle frasi soggettive del documento.

Queste informazioni si rivelano importanti negli articoli di informazione, negli articoli scientifici o nei documenti formali, nei quali molto spesso sono presenti opinioni (anche discordanti tra loro) appartenenti ad autori diversi. Nei *social network*, *microblog*, *blog* e *forum*, invece, l'autore è solitamente colui che ha scritto il commento o il post e, quindi, la sua identificazione è tendenzialmente più semplice.

1.3.4 *Object/feature extraction*

In piattaforme come *social network* e *microblog* ove, tendenzialmente, non vi è un solo specifico e determinato argomento di discussione ma, anzi, si è inclini a commentarne svariati, risulta estremamente importante determinare l'entità soggetta ad opinione. Tale attività rientra nella *object extraction*. Qualora, invece, l'opinione espressa coinvolga una o più proprietà di una stessa entità, ciò accade, ad esempio, nelle recensioni di prodotti, è necessaria un'estrazione più dettagliata per raccogliere sia le proprietà sia le singole opinioni espresse per ognuna di esse. In questi casi si parla di *feature extraction*.

1.4 Livelli di applicazione

Le classificazioni descritte nel precedente paragrafo possono essere applicate su diversi livelli di granularità (intero documento, singola frase o singola parola), così come rappresentato in figura 1.1. Le classificazioni su livelli diversi non sono indipendenti l'una dall'altra ma, anzi, ognuna di esse dipende dall'analisi di granularità inferiore: le classificazioni a livello di documento dipendono fortemente da quelle di frase che, a loro volta, dipendono da quelle a livello di parola.

1.4.1 *Document level sentiment analysis*

La *document level sentiment analysis* applica il processo di classificazione ad un intero documento, assumendo che questi contenga esclusivamente opinioni di un solo autore su una stessa entità. Diverse ricerche (Zhao *et al.*, 2014; Dhande, Patnaik, 2014; Sharma *et al.*, 2014; Kumar *et al.*, 2012) sono state svolte a riguardo sia nell'ambito della classificazione della polarità sia in quella della soggettività.

(Zhao *et al.*, 2014) descrivono diversi metodi di *machine learning* per la classificazione di articoli online. (Dhande, Patnaik, 2014) combinano un

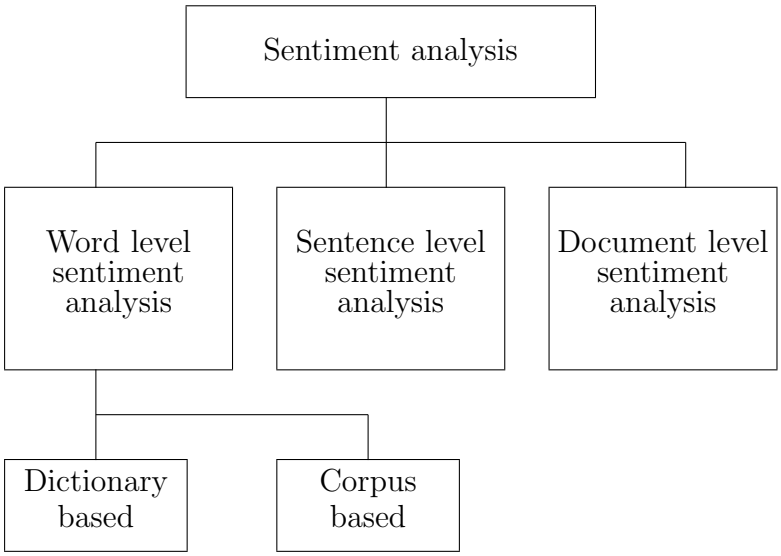


Figura 1.1: Livelli di applicazione della *sentiment analysis*

classificatore Naive Bayes ed uno basato su reti neurali (vedi par. 3.1.2) per la classificazione della polarità di recensioni di film, raggiungendo un'accuratezza del 80.65%. Sempre nell'ambito dei metodi basati sull'apprendimento automatico anche gli articoli di (Sharma *et al.*, 2014) e (Kumar *et al.*, 2012) mostrano tecniche per la classificazione della polarità di recensioni *online* utilizzando, però, metodi di apprendimento non supervisionati.

(Taboada *et al.*, 2011), invece, descrivono ed usano diversi metodi basati su dizionari e lessici annotati (vedi cap. 2) sia per la *sentiment polarity classification* sia per la *subjectivity classification* mentre (Turney, 2005) descrive un metodo di classificazione basato sulla combinazione lineare delle polarità delle opinioni espresse nei documenti sfruttando classificatori semantici non supervisionati (vedi cap 3 per ulteriori dettagli).

Le principali difficoltà della *document level sentiment analysis* risiedono nelle seguenti problematiche:

- distinguere fatti da opinioni

- identificare le polarità delle opinioni
- opinioni indirette
- opinioni espresse senza *opinion word*
- presenza di frasi con polarità discordanti tra loro

Da ciò si evince come uno dei passi fondamentali per la classificazione a livello di documento è la corretta classificazione delle frasi che compongono il documento stesso.

1.4.2 *Sentence level sentiment analysis*

La *sentence level sentiment analysis* si occupa del riconoscimento e della classificazione di singole frasi o di brevi messaggi di testo (*short-text*). Diverse ricerche sono state svolte a riguardo sia nell'ambito della classificazione di polarità sia in quella di soggettività.

Tra le ricerche in questo ambito si possono menzionare (Yu, Hatzivassiloglou, 2003) che hanno utilizzato un algoritmo di apprendimento supervisionato per l'identificazione ed un metodo simile a quello usato in (Turney, 2005) per la classificazione di frasi soggettive e (Liu *et al.*, 2005) che, invece, hanno formulato un semplice metodo di aggregazione delle polarità di singole parole presenti nelle frasi.

Ci si potrebbe aspettare che il riconoscimento di frasi soggettive sia realizzabile semplicemente con l'utilizzo di un dizionario di *opinion word* opportunamente etichettate ma, purtroppo, anche le frasi oggettive contengono *opinion word* che, se contenute in tale dizionario, porterebbero ad un'errata classificazione. Ciò però non riduce l'importanza per molti approcci alla *sentiment analysis*, in particolare quelli basati sul lessico (vedi cap. 2), della realizzazione di un'analisi accurata delle *opinion word*.

1.4.3 *Word level sentiment analysis*

In letteratura, molte ricerche sulla *sentiment analysis* a livelli di frase utilizzano la polarità a priori delle singole parole contenute nella frase stessa. Il valore di polarità viene solitamente ricavato accedendo ad un dizionario di *opinion word* (detto *lexicon*), costruito manualmente o automaticamente ed opportunamente etichettato. La creazione manuale di un *lexicon* prevede la selezione e la classificazione di aggettivi, nomi, verbi e avverbi a partire da un tradizionale dizionario. Per una creazione automatica (o semiautomatica), le tecniche da utilizzare sono sostanzialmente due: approccio basato su dizionari (*dictionary-based approach*) o approccio basato su corpora (*corpus-based approach*).

Approccio basato su dizionari

Per la realizzazione di un lessico polarizzato tramite il metodo *dictionary-based*, in primis, si realizza manualmente un piccolo dizionario di parole semanticamente orientate, successivamente, si estende tale insieme estraendo, ricorsivamente, sinonimi e contrari da un dizionario digitale, quale ad esempio WordNet[3] (Fellbaum, 1998). Ad ogni termine così ottenuto viene assegnato un valore reale, di solito compreso tra "-1" ed "1", da cui è possibile dedurne la sua classificazione. Valori positivi e negativi indicano una parola soggettiva con polarità positiva o negativa ("bello", "buono" - "brutto", "cattivo"), un valore prossimo allo "0" indica parole oggettive ("ieri", "oggi", "sole", "mare").

(Kim, Hovy, 2004) per la realizzazione del loro *lexicon*, hanno creato manualmente due insiemi disgiunti: uno di verbi e aggettivi positivi e l'altro di verbi e aggettivi negativi. Per ognuno di essi hanno estratto automaticamente sinonimi e contrari da WordNet ed inseriti opportunamente in uno e nell'altro insieme.

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

Basandosi sulle relazioni lessicali di WordNet, invece, (Kamps *et al.*, 2004) hanno creato un *lexicon* composto da grafi di sinonimi, ottenuto collegando coppie di parole dello stesso synset. Il punteggio di ogni parola è calcolato in funzione alla distanza relativa dalle parole *good* (buono) e *bad* (cattivo).

Lo svantaggio nell'uso di questo metodo risiede nell'impossibilità di classificare una parola in modo diverso in base al dominio in cui è inserita. Infatti, parole che in alcuni contesti risultano positive possono risultare negative in altri o viceversa.

Approccio basato su corpora

Il metodo *corpus-based*, per la classificazione di singole parole, usa tecniche sintattiche e statistiche di co-occorrenza di termini all'interno di corpora solitamente molto grandi.

(Hatzivassiloglou, McKeown, 2004), a partire da un insieme di aggettivi già etichettati, calcolano la polarità di termini presenti nei corpora assumendo che ogni coppia di parole ha stesso orientamento se collegata da congiunzioni copulative ("e", "anche", "pure", "inoltre", "ancora", "neppure", "nemmeno") e orientamento opposto se collegata da congiunzioni avversative o disgiuntive ("o", "ma", "oppure", "neppure"). Per poter ottenere risultato accurati, però, questo metodo richiede corpora estremamente grandi.

(Turney, 2005), invece, sfrutta la relazione di associazione tra parole assegnando una polarità positiva a parole con un buon grado di associazione (e.g."sole caldo"). Per la classificazione viene sfruttata, quindi, la relazione di associazione tra parole sconosciute ed un insieme di parole manualmente selezionate (poiché estremamente polarizzate).

Possiamo, quindi, riassumere i task della *sentiment analysis* come riportato nella tabella 1.1.

Granularità	**Classificazione**	**Classi**	**Assunzioni**
Documento	*Sentiment polarity*	2 o più	Ogni documento contiene opinioni di un solo autore su una stessa entità
Documento	*Subjectivity*	2	Ogni documento contiene opinioni di un solo autore su una stessa entità
Documento	*Opinion holder extraction*	-	-
Documento	*Object/feature extraction*	-	-
Frase	*Sentiment polarity*	2 o più	Ogni frase contiene una sola opinione
Frase	*Subjectivity*	2	-
Frase	*Opinion holder extraction*	-	-
Frase	*Object/feature extraction*	-	-
Parola	*Sentiment polarity*	2 o più	Ogni frase contiene una sola opinione
Parola	*Subjectivity*	2	-

Tabella 1.1: Classificazioni della *sentiment analysis* per granularità

1.5 Tecniche di classificazione

Come già osservato, la *sentiment analysis* può essere ricondotta ad un problema di classificazione di testi sfruttando gli approcci e le metodologie descritte nei paragrafi precedenti.

Finora si sono descritti i problemi di classificazione e la granularità del testo in cui possono essere applicabili ma, quali algoritmi possono essere

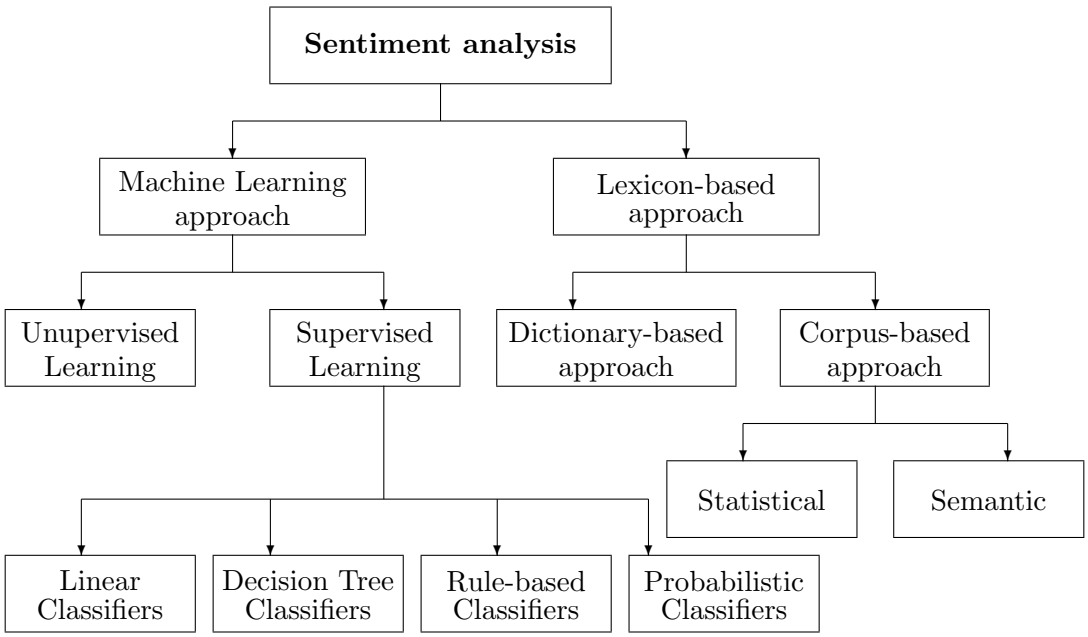


Figura 1.2: Tecniche per la classificazione della *sentiment analysis*

utilizzati?

All'attuale stato dell'arte, le tecniche di classificazione, a prescindere dal livello a cui sono applicate, sono riassumibili, in base agli algoritmi e agli strumenti utilizzati, in due principali categorie:

- basate sul lessico (*lexicon-based approach*)
- basate sull'apprendimento automatico (*machine learning approach*)

Questi approcci possono essere utilizzati individualmente o in combinazione tra loro e sono implementati con algoritmi che riprendono tecniche dell'intelligenza artificiale, con l'utilizzo di strumenti sintattici e semantici, con l'applicazione di concetti statistici e di calcolo delle probabilità.

La figura 1.2 riassume i differenti approcci e gli algoritmi più utilizzati per la classificazione nella *sentiment analysis*.

Gli approcci basati sul lessico, per la classificazione di frasi o documenti, sfruttano appositi dizionari creati seguendo le tecniche precedentemente descritte per l'analisi del sentimento a livello di parola. Uno dei *lexicon* più utilizzati per la lingua inglese è *SentiWordNet*[4] (Esuli, Sebastiani, 2006) che assegna ad ogni synset di *WordNet* tre score di polarità: "*positive*", "'*negative*" "*objective*".

Gli approcci basati sull'apprendimento automatico, invece, utilizzano tecniche di intelligenza artificiale che, a partire da un *training set* di esempi manualmente pre-etichettato, permettono di generalizzare la classificazione di qualsiasi altro contenuto testuale. La generalizzazione è realizzata tramite vettori di proprietà (dette *feature*) sintattiche e semantiche opportunamente create a partire da frasi o documenti.

Per una trattazione più accurata e approfondita dei due approcci si rimanda ai capitoli 2 e 3.

1.6 *Sentiment analysis* e la lingua italiana

Se le definizioni, gli approcci e le tecniche per la *sentiment analysis* finora descritte sono valide a prescindere dalla lingua utilizzata ciò non è altrettanto vero per gli strumenti, le regole grammaticali ed i modelli linguistici. Ogni lingua è diversa dall'altra e, ad esempio, pensare di condurre una *sentiment analysis* per la lingua italiana utilizzando gli stessi strumenti, gli stessi modelli e le stesse regole che funzionano per quella inglese, porterebbe solamente a risultati inaccurati.

Occorre pensare e sviluppare strumenti specifici della lingua soggetta all'analisi del sentimento: *POS-tagging*, alberi sintattici, tokenizzazione, lemmatizzazione, analisi sintattica e morfologica.

Di seguito un breve elenco di alcuni degli strumenti più utilizzati per la *sentiment analysis* per l'italiano:

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

- I dizionari lessicali: **MultiWordNet**[5] e **ItalWordNet**[6], le versioni per l'italiano di WordNet. Il primo è una mera traduzione dei termini, dei synset e delle relazioni lessicali e semantiche di WordNet, il secondo, invece, nasce interamente da progetti dedicati alla creazione di risorse linguistiche e software legati alla lingua italiana sia scritta sia parlata;

- **TextPro**[7] per l'analisi morfologica, lemmatizzazione, *POS-tagging*;

- **MaltParser**[8] per la realizzazione di strutture sintattiche a dipendenze;

- **TULE**[9] per l'analisi morfologica di frasi in linguaggio naturale e per la costruzione di strutture sintattiche a dipendenze.

L'attenzione per il trattamento automatico della lingua italiana è aumentata notevolmente nel corso degli ultimi anni, aumentano le aziende e le università che investono tempo e denaro sulle tecnologie linguistiche per la lingua italiana (forumTAL, 2009) e, quindi, anche sulla *sentiment analysis*. È possibile far riferimento a (Calzolari *et al.*, 2009) per una panoramica sul trattamento automatico della lingua italiana a partire dal 2009.

Un aiuto molto importante nell'ambito della ricerca italiana arriva anche dalle campagne di valutazione sui tool per la linguistica computazionale di EVALITA[10]. Tali campagne hanno lo scopo di mettere a confronto sistemi e tecnologie riguardanti il trattamento automatico della lingua italiana al fine di supportarne la diffusione e la ricerca di nuove. Le campagne di valutazione sono aperte a tutte le società e le università italiane o internazionali che lavorano sulla lingua italiana e, tra i task con più partecipanti, vi è proprio la *sentiment polarity classification*.

Da tali campagne sono nati diversi lavori di ricerca, tra i quali:

[5]http://multiwordnet.fbk.eu/
[6]http://www.ilc.cnr.it/iwndb_php/
[7]http://hlt-services2.fbk.eu/textpro-demo/textpro.php
[8]http://www.maltparser.org/
[9]http://www.tule.di.unito.it/
[10]http://www.evalita.it

- (Basile, Novielli, 2014) posizionatosi primo ad EVALITA 2014 con un sistema di classificazione basato sull'estrazione di *feature blog-based*, semantiche e *sentiment* e di un lessico annotato ottenuto per traduzione di SentiWordNet;

- (Cimino *et al.*, 2014) con un approccio supervisionato basato su algoritmi stocastico/statistici di *machine learning* ed estrazioni di *feature*.

Capitolo 2

L'approccio basato sul lessico

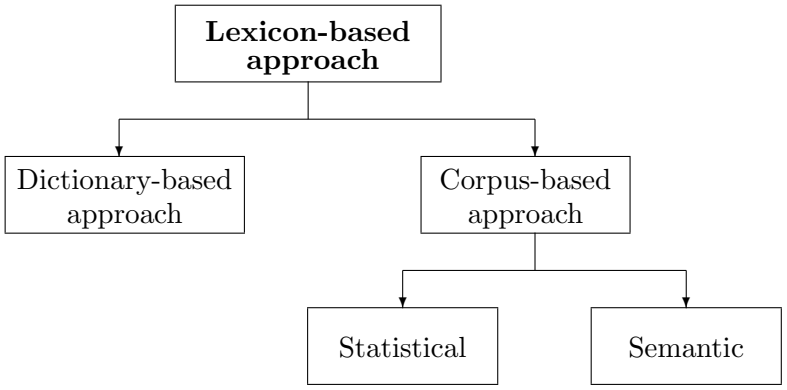


Figura 2.1: Tecniche per la classificazione della *sentiment analysis* basate sul lessico

L'approccio *lexicon-based* rientra tra le tecniche di classificazione non supervisionate. La classificazione è svolta calcolando l'orientamento semantico di frasi e documenti a partire dalla polarità di parole.

I termini e le espressioni verbali che trasmettono, da sole, una opinione marcata sono estremamente interessati per la *sentiment analysis* e possono essere raccolte ed etichettate per comporre un *opinion lexicon* (lessico delle opinioni) necessario alla classificazione. A tal fine si può procedere in modo

manuale, con il lavoro di linguisti e di madrelingua che identificano e categorizzano parole nelle classi di interesse, ma poiché tale approccio richiede molto tempo, si tende ad utilizzare un approccio automatizzato.

La classificazione *lexicon-based* può essere svolta seguendo due diverse metodologie:

- basata su dizionari (*Dictionary-based approach*)
- basata su corpora (*Corpus-based approach*)

Entrambi i metodi prevedono l'individuazione di parole semanticamente rilevanti all'interno del documento o della frase da classificare ma mentre il *corpus-based approach* effettua una ricerca in un corpus (solitamente molto ampio) annotato, il *dictionary-based approach* ricerca le parole all'interno di dizionari opportunamente costruiti etichettando ogni termine semanticamente rilevante con informazioni sintattico-lessicali, sinonimi e contrari ma, soprattutto, con un punteggio di orientamento del sentimento. Di seguito si discutono i due metodi nel dettaglio.

2.1 L'approccio basato su dizionari

Negli approcci di tipo *dictionary-based* viene, in primis, manualmente costruito un insieme ridotto di *opinion word* alle quali viene assegnato un valore di polarità. Successivamente questo insieme viene allargato grazie all'integrazione di sinonimi e contrari presenti all'interno di dizionari lessicali (come WordNet) o di appositi dizionari di sinonimi. I nuovi termini trovati vengono, di volta in volta, aggiunti alla lista iniziale. Tale procedimento viene iterato finché non si individua nessuna nuova parola. Solitamente, a questo punto, si esegue un controllo manuale per apportare eventuali correzioni.

Uno dei dizionari più utilizzati per la *sentiment analysis* è SentiWordNet (per la lingua inglese ma utilizzabile anche per l'italiano tramite l'incrocio con MultiWordNet che ne consente un buon processo automatico di traduzione).

Lo svantaggio principale nell'utilizzo di questo approccio è l'impossibilità di trovare, riconoscere ed etichettare parole con un orientamento specifico in un certo dominio o con un significato diverso in base alla posizione o al contesto in cui si trova.

2.2 L'approccio basato su corpora

I metodi di tipo *corpus-based* aiutano a risolvere il problema degli approcci basati su dizionario precedentemente descritto.

Si utilizzano schemi sintattici o, comunque, schemi che, in concomitanza con una lista di partenza di *opinion word*, consentono di trovare nuove parole all'interno di un corpus e di identificarne l'orientamento. Uno di questi metodi è stato presentato da (Hatzivassiloglou, McKeown, 2004), i quali, a partire da una lista di aggettivi e un insieme di vincoli linguistici, hanno costruito un metodo basato su corpora per identificare nuove parole e la loro polarità. I vincoli sono rappresentati dai connettivi come "e", "o", "ma", etc. Solitamente, infatti, una congiunzione indica uno stesso orientamento tra aggettivi collegati, mentre una disgiunzione indica un cambio di opinione. Questi tipi di collegamenti tra gli aggettivi formano un grafo su cui vengono successivamente applicati algoritmi di *clustering* (analisi di gruppi), per poter ottenere due insiemi di parole distinte, quelle con valenza positiva e quelle con valenza negativa.

Un altro metodo è stato presentanto da (Turney, 2005) che, sfruttando correlazione e co-occorenze fra parole, assegna una polarità positiva a parole con un buon "grado di associazione" (e.g. "sole caldo", "ottimo voto", "grandi ascolti", "batte forte il cuore") e polarità negativa altrimenti. Per la classificazione viene sfruttata, quindi, la co-occorrenza tra parole fino a quel momento sconosciute ed un insieme di parole manualmente selezionate poiché particolarmente polarizzate. Il grado di correlazione è determinato contando il numero di risultati di una ricerca online su *AltaVistaSearch Engine* e

calcolando l'informazione mutua puntuale (*point-wise mutual information* - PMI).

Gli approcci *corpus-based*, si sono rivelati essere meno efficaci di quelli basati sul dizionario per via della difficoltà nella costruzione di corpora abbastanza grandi da coprire tutte le parole (e le loro combinazioni) di una lingua. Nonostante ciò, facendo uso di appositi metodi statistici o semantici, è possibile comunque assegnare valori di polarità ai termini presenti in un corpus con la possibilità di assegnarne di diversi a stesse parole ed espressioni in base al contesto e al dominio in cui sono inserite. Ciò è, invece, irrealizzabile con un approccio *dictionary-based*.

Approccio statistico

Gli approcci statistici eseguono ricerche per trovare occorrenze di termini che esprimono opinioni, determinandone la polarità a posteriori tramite co-occorrenze con altri termini presenti in una raccolta manuale o con un insieme di documenti opportunamente classificati. La polarità di una parola può, infatti, essere identificata analizzandone la frequenza in una raccolta di corpus: se appare maggiormente in testi positivi è possibile assegnare orientamento positivo, se appare maggiormente in testi negativi è possibile assegnare polarità negativa, nel caso di stessa frequenza tra testi negativi e positivi è possibile assumere una valenza neutra.

L'idea di base, all'attuale stato dell'arte, è che parole con polarità simile compaiono all'interno dello stesso contesto e, quindi, tendono ad avere stessa polarità. Ciò consente di determinare l'orientamento di una parola calcolandone la frequenza relativa, all'interno di un testo, rispetto ad un'altra parola di cui già si conosce la polarità. Per questo scopo si può utilizzare PMI che, nella statistica e nella teoria dell'informazione, è usata per misurare il grado di associazione tra due eventi.

Approccio semantico

Gli approcci basati sulla semantica assegnano valori di polarità alle parole affidandosi ad alcuni principi semantici per analizzare la somiglianza tra parole. In particolare, in molte ricerche, si è partiti dal principio tale per cui parole semanticamente "vicine" hanno polarità simile. Seguendo tale idea, sono nati diversi metodi e strumenti ad oggi estremamente diffusi in ambito della *sentiment analysis* poiché permettono la costruzione di ottimi modelli lessicali per il calcolo della polarità di *opinion word*. Questo approccio, ad esempio, è alla base dei maggiori lessici presenti online, quali WordNet e SentiWordNet (si veda paragrafo successivo).

Metodi statistici e metodi semantici possono anche essere combinati. (Zang, Xu, 2009), ad esempio, si sono serviti di entrambi i metodi per il loro studio di ricerca sulle debolezze di un prodotto. In particolare sono riusciti a determinare gli aspetti meno soddisfacenti degli articoli commerciali analizzati sfruttando le opinioni espresse dagli utilizzatori degli stessi in alcune recensioni *online*. Per far ciò, tramite metodi statistici hanno estratto le proprietà ed i componenti dei prodotti identificando quelli più e quelli meno frequenti. Tramite metodi semantici, hanno raggruppato le proprietà ed i componenti che fanno riferimento allo stesso aspetto del prodotto e, infine, con metodi di *sentence-level sentiment analysis*, hanno determinato la polarità di ogni aspetto per evincere quelli più deboli.

2.3 WordNet e SentiWordNet

Due database semantico-lessicali, nati da un approccio semantico *corpus-based* e diventati oramai dei pilastri per la *sentiment analysis* con metodi basati sul lessico, sono WordNet e SentiWordNet.

2.3.1 WordNet

WordNet (Fellbaum, 1998) nasce nel 1985 da un progetto di linguisti e psicologi dell'Università di Princeton e, seppur realizzato per la lingua inglese, è stato successivamente tradotto in molteplici lingue, fra le quali anche l'italiano.

WordNet si distingue da un normale dizionario per due aspetti principali:

- divisione del significato delle parole basata due concetti: la forma scritta (*word form*), i cosiddetti lessemi o lemmi, ed il suo significato (*word meaning*)
- diversa catalogazione del lessico basata su categorie sintattiche, relazioni lessicali e semantiche invece che da un un mero ordinamento alfabetico

Il dizionario di WordNet è diviso in quattro categorie sintattiche: sostantivi, verbi, aggettivi ed avverbi ognuna delle quali è, a sua volta, raggruppata in insiemi di sinonimi detti *synset* (*synonym set*). Ogni insieme di sinonimi rappresenta un particolare concetto ed è posto in relazione con altri synset tramite relazioni semantiche. Le relazioni fra singoli lemmi o *synset* avviene, invece, tramite relazioni lessicali.

Tra le relazioni più importanti presenti in WordNet vi sono:

Sinonimia la relazione tra due lessemi con stesso significato

Antonimia relazione tra due lessemi di significato opposto (esempio "buono/cattivo" o "aprire/chiudere")

Polisemia la proprietà di un lessema di avere due o più significati (esempio "squadra, "lira", "miglio")

Iponimia relazione semantica tra due termini, uno dei quali (detto iponimo) è semanticamente incluso nell'altro (esempio "sedia", "tavolo", "armadio" sono iponimi di "mobile")

Meronimia relazione semantica tra due termini, uno dei quali (detto meronimo) è un costituente o un membro dell'altro (esempio "naso" è meronimo di "viso")

2.3.2 SentiWordNet

SentiWordNet (Esuli, Sebastiani, 2006) è una risorsa lessicale sviluppata da Andrea Esuli e Fabrizio Sebastiani con lo scopo di realizzare, partendo da WordNet 3.0, un lessico adatto alla *sentiment analysis lexicon-based*. Attualmente arrivato alla versione 3.0, SentiWordNet riporta, per ogni synset, tre punteggi di polarità, "*positive*", "*negative*, "*objective*", partendo dai seguenti presupposti:

- tutti i termini appartenenti allo stesso *synset* hanno medesima polarità
- una stessa parola, se appartiene a *synset* diversi, ha diversa polarità in base al suo significato

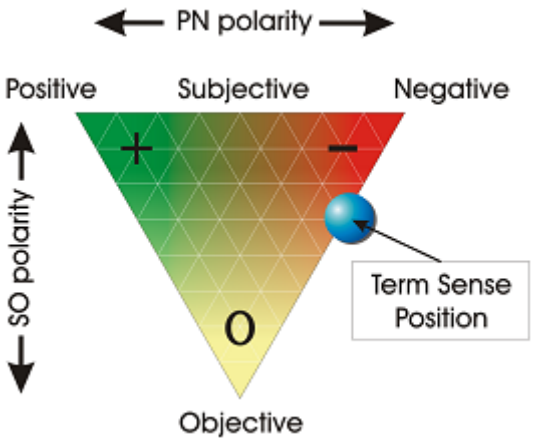


Figura 2.2: Rappresentazione grafica dei punteggi dei *synset* di SentiWordNet

La costruzione di SentiWordNet parte, quindi, dalla classificazione dei *synset*, i cui punteggi sono ottenuti per *ensamble learning* (Russel, Norving, 2005) su vari classificatori ternari "allenati" su differenti *training set*.

I punteggi dei *synset* possono essere rappresentati graficamente in un triangolo come in figura 2.2. Gli angoli rappresentano le tre etichette (in alto a sinistra la classe positiva, in alto a destra la classe negativa, in basso la classe oggettiva), il pallino blu indica la posizione del significato del termine, gli assi *PN-polarity* e *SO-polarity*, infine, consentono l'interpretazione corretta della polarità e dell'oggettività del termine.

Uno sguardo al sito[1] di SentiWordNet per la lingua inglese permette di osservare come il termine *good* (buono) abbia 21 utilizzi diversi in qualità di aggettivo, 4 come nome e 2 come avverbio. Ciò deriva dal fatto che, nelle lingue naturali, uno stesso termine può essere utilizzato in modi, contesti e significati diversi e rappresenta uno dei principali motivi per cui, ancora oggi, non è stato implementato un programma informatico in grado di analizzare e comprendere esattamente il linguaggio umano.

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

Capitolo 3

L'approccio basato sull'apprendimento automatico

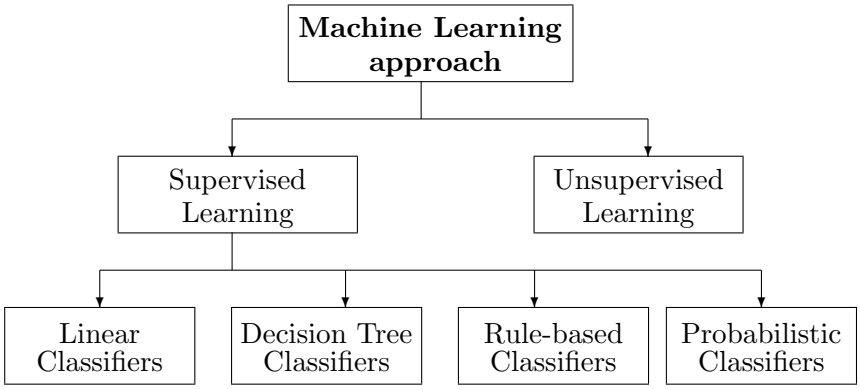


Figura 3.1: Tecniche per la classificazione della *sentiment analysis* basate sull'apprendimento automatico

Gli approcci basati sul *machine learning* si affidano ad algoritmi di intelligenza artificiale (Russel, Norving, 2005) per la risoluzione dei problemi della *sentiment analysis*. Generalmente, tali algoritmi prendono in input un insieme di esempi (*training set*) e restituiscono in output un modello generale per la classificazione.

All'attuale stato dell'arte, si distinguono due categorie di approcci all'analisi del sentimento basati su algoritmi di *machine learning*:

con apprendimento supervisionato (*supervised learning*) il sistema acquisisce conoscenze ed esperienza per la classificazione a partire da un *training set* di testi già classificati ed etichettati

con apprendimento non supervisionato (*unsupervised learning*) il sistema acquisisce conoscenze ed esperienza per la classificazione dall'estrazione di caratteristiche comuni da un *training set* di testi non etichettati

3.1 Apprendimento supervisionato

I metodi di machine learning di tipo supervisionato necessitano, quindi, di documenti etichettati per la "fase di apprendimento" del classificatore. Poiché un *training set* di interi testi è eccessivamente grande per poter essere utilizzato in pratica, gli algoritmi di *machine learning* prevedono che, per ogni documento da classificare, sia definito un vettore di proprietà (dette *feature*) che lo rappresentino.

3.1.1 L'estrazione delle *feature*

Dato un qualsiasi testo, l'estrazione delle *feature* è il processo di estrapolazione delle sue caratteristiche e proprietà salienti. Queste proprietà devono, appunto, rappresentare le caratteristiche fondamentali del testo ma la loro estrazione non è così immediata: se da un lato dovrebbero discriminare e descrivere il più possibile il testo originale, dall'altro dovrebbero anche ridurre l'ampia dimensione dei dati di origine ed evitare ridondanze.

Le *feature* più utilizzate in letteratura sono:

Parole identificazione di unigrammi, bigrammi, n-grammi di parole presenti nel documento

Parti del discorso presenza/assenza di aggettivi, avverbi, nomi, verbi solitamente riconosciuti tramite *POS-tagging*. Le parti del discorso sono usate per disambiguare, seppure in parte, il senso dei termini e per identificare aggettivi e avverbi che, solitamente, sono ottimi indicatori dell'orientamento semantico

Sintassi riconoscimento di combinazioni sintattiche, di solito ottenute tramite *parser* e strutture sintattiche a dipendenze. Alcuni studi hanno dimostrato come algoritmi con *feature* sintattiche e algoritmi con *feature* basate sugli n-grammi forniscono risultati simili (Pang, Lee, 2008)

Opinion word riconoscimento di parole che, da sole, esprimono un'opinione netta

Negazione presenza/assenza di negazioni che, solitamente, invertono le opinioni espresse

Una volta estratte le *feature* è necessario calcolare il loro "peso" all'interno del documento. Un approccio è quello basato sulla presenza: "0" se la *feature* non appare ed "1" se la *feature* appare nel documento. Altri approcci, usati solitamente per *feature* di parole, sono quelli basati sulla *term frequency* (frequenza del termine all'interno del testo) e dell'*inverse document frequency* (in quanti testi compare il termine). In generale, nell'*information retrieval* e nella classificazione dei testi, è preferibile pesare queste *feature* utilizzando la *term frequency* in quanto consente di ottenere risultati migliori. (Pang *et al.*, 2002) hanno però dimostrato come nella *sentiment analysis* sia preferibile assegnare un valore alle *feature* basandosi sulla presenza/assenza piuttosto che sulla *term frequency*.

3.1.2 I classificatori

Negli ultimi decenni, gli studi nell'ambito degli algoritmi di classificazione basati sull'apprendimento automatico sono aumentati notevolmente.

È possibile distinguere:

- Classificatori probabilistici
- Classificatori lineari
- Classificatori basati su alberi di decisione
- Classificatori basati su regole
- Classificatori basati su reti neurali

Classificatori probabilistici

I classificatori probabilistici fanno uso di modelli statistici ed utilizzano le *feature* estratte dai testi per identificare una corretta classificazione. Nella fase di *training*, quindi, vengono memorizzati i parametri delle distribuzioni di probabilità delle classi e delle *feature*; la classificazione viene poi realizzata tramite valutazione delle probabilità delle *feature* estratte nelle diverse classi.

I più famosi classificatori probabilistici sono il **Naive Bayes** ed il **Maximum Entropy**.

Il classificatore Naive Bayes è il classificatore probabilistico più utilizzato in letteratura. Questo modello calcola la probabilità a posteriori con la quale una certa *feature* appartiene ad una particolare classe (o etichetta) in base alla distribuzione delle parole nel documento, all'utilizzo di *feature* di parole e al teorema di Bayes. Per far ciò assume che le *feature* siano tutte indipendenti tra loro; assunzione ovviamente non vera in tutti i casi reali.

Sotto questo presupposto, la classificazione Naive Bayes si può ricondurre alla soluzione della seguente equazione:

$$P(\frac{label}{feature}) = \frac{P(label)*P(\frac{feature1}{label})*P(\frac{feature2}{label})*...*P(\frac{featureN}{label})}{P(features)}$$

$P(label)$ è la probabilità a priori di un'etichetta (la probabilità con cui un insieme di *feature* casuale ricade in tale etichetta), $P(\frac{featureX}{label})$ è la probabilità a priori che una certa *feature* sia classificata con quell'etichetta, mentre $P(features)$ è la probabilità a priori, di un certo insieme di feature, di apparire insieme.

Il classificatore Maximum Entropy, conosciuto per essere un classificatore esponenziale condizionale, trasforma insiemi di *feature* in vettori opportunamente codificati. Tali vettori sono poi utilizzati direttamente per il calcolo dei valori delle *feature* e combinati tra loro per determinare la classe di appartenenza più adatta per un certo insieme di *feature* (cioè di un testo).

La probabilità di ogni etichetta è calcolata utilizzando la seguente equazione:

$$P(\frac{fs}{feature}) = \frac{dotprod(weights,encode(fs,label))}{sum(dotprod(weights,encode(fs,I),per\ ogni\ label)}$$

Dove $dotprod(weights, encode(fs, label))$ indica il prodotto scalare tra i pesi e il vettore ottenuto tramite codifica e $sum(dotprod(weights, encode(fs, I),$ *per ogni label*) rappresenta la somma di tutti i prodotti scalari, per ogni etichetta.

Classificatori lineari

Come i classificatori probabilistici precedentemente descritti, anche i classificatori lineari hanno l'obiettivo di usare i vettori di *feature* per identificare la corretta classe di appartenenza del testo che rappresentano. Un classificatore lineare raggiunge, però, questo scopo tramite una combinazioni lineare di *feature*.

Il più famoso classificatore lineare è il **Support Vector Machine** (SVM).

I classificatori di tipo SVM rientrano nei metodi di *machine learning* di tipo supervisionato che, a partire da un *training set* opportunamente etichettato, costruiscono un modello generale per la classificazione. Il *training set* rappresenta un insieme di punti di un piano. Ogni esempio è collocato nel piano sfruttando i valori del vettore di *feature* che lo rappresenta.

L'algoritmo di SVM trova le migliori linee di separazione che definiscono i più grandi iperpiani capaci di rappresentare, con il minimo errore, le diverse etichette della classificazione. Tali linee sono ottenute massimizzando la distanza dei punti più vicini delle diverse classi rappresentate nel *training set*.

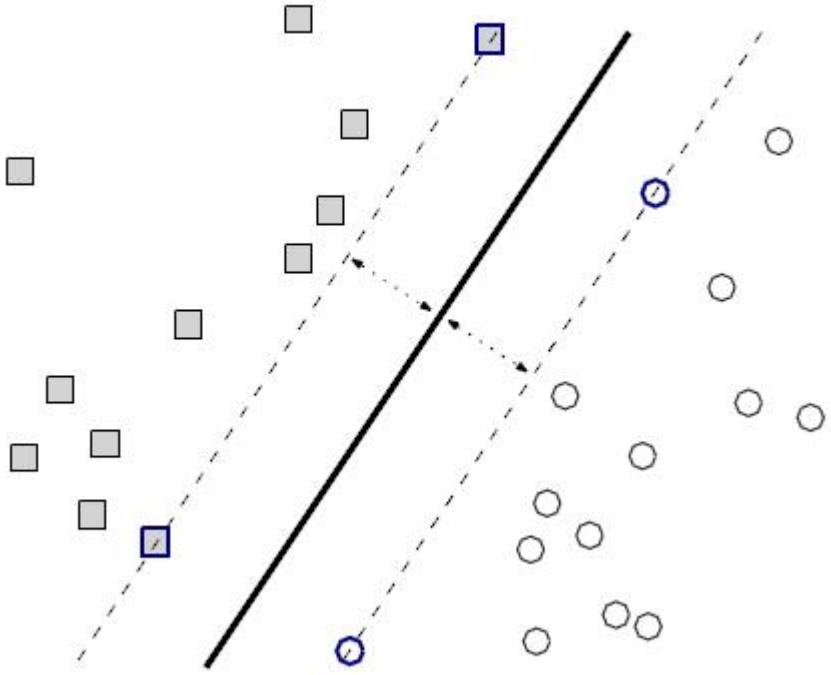


Figura 3.2: SVM lineare con vettori di supporto e massima distanza tra due classi. I vettori di supporto sono rappresentati da rettangoli e cerchi, in blu i punti più vicini al margine di separazione.

La figura 3.2 rappresenta graficamente un esempio di iperpiano costruito da un classificatore lineare SVM per un'etichettatura a due classi. Sotto forma di rettangoli e cerchi sono riportati, tramite i loro vettori di *feature*, gli esempi del *training set* divisi dalla linea in grassetto che rappresenta la miglior separazione possibile poiché massimizza la distanza tra i punti più vicini (evidenziati in blu).

A questo punto, è possibile classificare un qualsiasi testo identificando l'iperpiano di appartenenza a partire dal vettore di *feature* che lo rappresenta. Ovviamente, più il *training set* è ampio e più le *feature* estratte sono rappresentative e maggiore sarà l'accuratezza raggiungibile. È possibile utilizzare i classificatori di tipo SVM anche per costruire piani di decisione non lineari nello spazio delle *feature*, mappando in modo non lineare le istanze di dati

con uno spazio vettoriale e ricercando curve e linee capaci di rappresentare, con il minimo errore, le diverse classi della classificazione.

I classificatori SVM sono diventati estremamente popolari nell'ambito della *sentiment analysis*. Tale popolarità è dovuta agli ottimi risultati mostrati in diversi articoli di ricerca. (Pang *et al.*, 2002) hanno osservato i risultati prodotti da un classificatore Naive Bayes, uno Maximum Entropy ed uno SVM osservando come quest'ultimo porti ad una classificazione migliore. Ciò è dovuto, in particolare, alla natura intrinseca delle *feature* estraibili dai testi: sono tutte rilevanti, sono spesso correlate tra loro e, generalmente, rientrano in categorie linearmente separabili.

Classificatori basati su alberi di decisione

I classificatori basati su alberi di decisione utilizzano le proprietà degli esempi presenti nel *training set* per costruire un albero (detto albero di decisione) che rappresenti una decomposizione gerarchica dell'intero spazio dei dati. La creazione dell'albero avviene scegliendo ricorsivamente le proprietà più discriminanti (cioè che dividono nel maggior numero di parti lo spazio dei dati) al fine di costruire i più brevi percorsi radice-foglia che rappresentino tutti gli esempi presenti nel *training set*. Le foglie contengono le etichette delle classi per la classificazione.

Nell'ambito della *sentiment analysis*, un classificatore basato su alberi di decisione utilizza le *feature* del *training set* per la costruzione dell'albero; in questo modo è possibile ottenere la classificazione di un qualsiasi testo seguendo il percorso radice-foglia che rispetti i valori delle *feature* che lo rappresentano.

Classificatori basati su regole

Nei classificatori basati su regole, i dati sono modellizzati da insiemi di regole. Una regola è una relazione tra una congiunzione di *feature* (solitamente espressa in forma normale congiuntiva) ed una etichetta di classe ed è utilizzata per memorizzare e manipolare la conoscenza.

Un sistema che usa regole per la classificazione è solitamente composto da due parti fondamentali: un insieme di regole per rappresentare la conoscenza acquisita ed un sistema di inferenze per dedurre informazioni e prendere decisioni.

Nell'ambito della *sentiment analysis*, un classificatore basato su regole usa le *feature* del *training set* per costruire la propria base di conoscenze ed un sistema di inferenza, solitamente basato su deduzioni logiche, per combinare le regole ed ottenere le classificazioni.

Classificatori basati su reti neurali

I classificatori basati su reti neurali, infine, utilizzano un modello matematico (una rete neurale artificiale) per simulare una rete di neuroni biologici con l'intento di risolvere problemi copiando il comportamento e le capacità del cervello umano.

Il cervello umano è composto da miliardi di neuroni collegati tra loro ed ogni neurone è composto da tre parti: il corpo cellulare, il dentrite ed l'assone. Un neurone riceve segnali tramite il dentrite ed invia impulsi elettrici attraverso l'assone che, tramite diverse ramificazioni, raggiunge gli altri neuroni. Al termine di ogni ramificazione vi è la sinapsi che, convertendo l'attività dell'assone in impulsi elettrici, attiva o disattiva il corrispondente neurone di destinazione. L'apprendimento umano, quindi, avviene attraverso l'aumento di efficacia delle sinapsi cioè dell'influenza di un neurone sull'altro.

Le reti neurali artificiali seguono questo tipo di comportamento: calcolano l'assetto dei propri "neuroni" in base agli esempi forniti nella fase di apprendimento ed utilizzano questa configurazione per prendere decisioni e classificare elementi mai visti.

3.2 Apprendimento non supervisionato

Nella *sentiment analysis*, così come in molti problemi legati alla classificazione di testi, risulta molto difficile creare un *training set* di documenti

etichettati. È molto più facile costruirne uno di testi non etichettati.

Diversamente dall'apprendimento supervisionato, un algoritmo di *machine learning* con apprendimento non supervisionato per l'analisi del sentimento utilizza, durante la fase di *training*, solamente testi non annotati (e non *feature*) e lavora confrontando parole, cercando similarità e differenze.

Un esempio di apprendimento non supervisionato molto utilizzato nel trattamento automatico del linguaggio naturale è il modello *Latent Dirichlet Allocation* (LDA). È un modello di analisi che permette di dedurre gli argomenti trattati in un determinato testo analizzando la somiglianza tra la distribuzione dei termini del documento e quella assegnata a priori ad un insieme di argomenti predefiniti. Un documento è considerato come un insieme di argomenti, ognuno dei quali è caratterizzato da una particolare distribuzione di termini e parole.

L'algoritmo LDA attribuisce un argomento ad ogni parola del documento analizzandone la co-occorrenza con l'insieme di parole chiave assegnate a priori ad ogni argomento. Di conseguenza, associa ad ogni documento gli argomenti più rappresentati dalle parole che lo compongono. In questo modo, senza dover analizzare né semanticamente né sintatticamente le frasi dei documenti, è possibile identificare gli argomenti di un testo solamente osservando le co-occorrenze delle parole rispetto ad una base di conoscenza di riferimento.

(Pang *et al.*, 2013) hanno presentato un modello LDA per l'identificazione degli argomenti discussi in recensioni in lingua cinese e l'assegnamento della polarità del sentimento ad ognuno degli argomenti estratti. Hanno dimostrato come i buoni risultati ottenuti dall'uso di un algoritmo LDA per il riconoscimento degli argomenti comporti un miglioramento dell'accuratezza della *sentiment analysis*.

Gli approcci *machine learning* supervisionati e non supervisionati, in diversi studi, sono stati combinati per cercare di ottenere risultati migliori.

Capitolo 4

Due approcci alla *sentiment polarity classification* di *tweet* per la lingua italiana

Questo lavoro di tesi illustra la costruzione di due sistemi automatici per la rilevazione e la valutazione delle opinioni espresse sul *social network* Twitter. Tali sistemi dovranno essere in grado di classificare le informazioni contenute nei *tweet* in lingua italiana, opportunamente estratti da Twitter, valutando la polarità delle opinioni espresse.

4.1 Twitter

Twitter, nato nel marzo 2006 per mano della Obvious Corporation di San Francisco, è una piattaforma gratuita di *social network* e *microblogging* usata da milioni di persone nel mondo per la condivisione e la visualizzazione di informazioni. Nel 2016, è il quinto *social network* del mondo per numero di utenti attivi (320 milioni), alle spalle di Facebook (1.6 miliardi), QZone (653 milioni), Tumblr (555 milioni) e Instagram (400 milioni). In Italia, invece, è la terza piattaforma *social* più usata alle spalle di Facebook e Google+ (Kemp, 2016).

Twitter si caratterizza, rispetto ad altri *social network*, per la possibilità di pubblicare soli messaggi brevi (detti *tweet*) con al massimo 140 caratteri. Inizialmente tali messaggi erano unilaterali, senza poter costruire delle vere e proprie conversazioni, successivamente è stata introdotta la possibilità di rispondere e menzionare altri utenti nei propri *tweet* inserendo il carattere "@" prima del nome utente stesso. È possibile, inoltre, creare dibattiti e conversazioni su specifici argomenti aggiungendo il carattere "#" prima del nome del *topic*.

Ogni utente può "seguire" altri utenti ed "essere seguito" da uno o più utenti al fine di riceverne i messaggi e di consentirne la visualizzazione dei propri. I *follower* sono tutti gli utenti da cui si è "seguiti" mentre i *following* sono tutti gli utenti che si "seguono". A differenza di altri *social network*, la relazione tra utenti può essere unilaterale ed un utente può essere *follower* di un altro utente e non *following* e viceversa.

Uno dei principali motivi tale per cui Twitter si presta maggiormente, rispetto ad altri *social network*, alla *sentiment analysis* è la sua natura "pubblica":

"Quello che condividi su Twitter può essere visto istantaneamente in tutto il mondo. Sei quello che twitti!"[1]

A tal scopo Twitter mette a disposizione una serie di API[2] per visualizzare i *tweet* degli utenti, per ottenere l'insieme dei *followers* e dei *following*, per ricercare *tweet* per contenuto, etc...

4.2 *Twitter-specific sentiment analysis*

La *Twitter sentiment analysis* si distingue dalla tradizionale analisi del sentimento per la struttura stessa dei *tweet*: brevi, presenza di slang, *emoti-*

[1] https://twitter.com/privacy?lang=it
[2] https://dev.twitter.com/docs

con, errori ortografici. In molti articoli di ricerca, la classificazione dei *tweet* è realizzata tramite l'uso di tecniche di apprendimento automatico per due principali motivi:

1. Disponibilità di un enorme quantità di dati per la fase di *training*

2. Presenza esplicita di indicatori di polarità (es. *emoticon*) nel testo stesso dei *tweet* che, in molti casi, consentono di evitare l'oneroso compito dell'etichettatura manuale

(Go *et al.*, 2009) hanno realizzato una classificazione di polarità a due classi (positiva e negativa) per la lingua inglese. Le *emoticon* sono state utilizzate per collezionare ed etichettare i *tweet* estratti tramite le API di Twitter. Ogni *tweet* così ottenuto è stato opportunamente manipolato prima di essere utilizzato per la costruzione del classificatore: sostituzione di *url* e nomi utenti con i termini URL e USERNAME rispettivamente, rimozione di caratteri duplicati, etc. Ogni *tweet* è stato, infine, trasformato in un insieme di *feature* di unigrammi ed utilizzato per la classificazione tramite tre classificatori supervisionati: Naive Bayes, Maximum Entropy e SVM (vedi cap. 3). I migliori risultati sono stati ottenuti con l'algoritmo Naive Bayes (84% di accuratezza).

(Barbosa, Feng, 2010) hanno realizzato, invece, un classificatore in due fasi per la *subjectivity classification* e, successivamente, per la *sentiment polarity classification* basato sulle meta-informazioni di Twitter e sulla sintassi dei *tweet* stessi invece che sugli n-grammi poiché ritenuti poco efficaci per la classificazione di messaggi così brevi. Hanno realizzato vettori di *feature* basati sui *POS-tag*, sulla polarità e soggettività a priori delle parole nonché sulla presenza di *hashtag* ("#"), *mention* ("@"), *retweet*, *emoticon*, punteggiatura, parole in maiuscolo. I risultati ottenuti mostrano un'accuratezza del 81.9% e del 81,3% per la *subjectivity* e la *polarity classification* rispettivamente.

Per la lingua italiana, invece, negli ultimi anni sono stato pubblicati diversi lavori di ricerca nell'ambito della *Twitter sentiment analysis*. Tra le attività di ricerca più importanti si possono menzionare le campagne di valutazione per la *subjectivity classification* e la *sentiment polarity classification* di EVALITA. Nel 2014[3] e nel 2016[4], infatti, sono state lanciate due campagne di valutazione di *tool* per la *sentiment polarity classification* di *tweet* in lingua italiana che hanno visto la partecipazione di decine di sistemi.

4.3 I due approcci: in breve

Nei successivi capitoli saranno descritti nel dettaglio due sistemi per *sentiment polarity classification* di *tweet* per la lingua italiana sviluppati per questo lavoro di tesi. I due sistemi, entrambi realizzati in linguaggio C++, si distinguono per gli approcci utilizzati: il primo (*FICLIT+CS@Unibo System*) utilizza un approccio basato sul lessico mentre il secondo è basato su algoritmi stocastico/statistici di apprendimento automatico.

FICLIT+CS@Unibo System (Di Gennaro, Rossi, Tamburini, 2014) è stato presentato alla campagna di valutazione di EVALITA 2014 per il task SENTIment POlarity Classification (SENTIPOLC). Si basa su un approccio legato all'identificazione di *opinion word*, precedentemente etichettate per orientamento semantico in un lessico, ed una classificazione dei *tweet* basata sull'aggregazione della polarità a priori di singole parole.

Il secondo sistema, invece, nasce da un processo di apprendimento supervisionato, tramite un classificatore di tipo *Support Vector Machine*. L'algoritmo sviluppato estrae da ogni singolo *tweet* un vettore di *feature* lessicali, morfosintattiche e di polarità che, opportunamente somministrate ad un clas-

[3] http://www.evalita.it/2014/tasks/sentipolc
[4] http://www.evalita.it/2016/tasks/sentipolc

sificatore SVM, permettono di costruire un modello generale per la *sentiment polarity classification* di *tweet* in lingua italiana.

Capitolo 5

Primo sistema: l'approccio basato sul lessico (FICLIT+CS@UniBO System)

In questo capitolo si descrive la progettazione di un sistema automatico per la classificazione della *sentiment polarity classification* di *tweet* per la lingua italiana che ha partecipato al task SENTIment POLarity Classification (SENTIPOLC) della campagna di valutazione EVALITA 2014.

Il lavoro di progettazione e di sviluppo del sistema ha previsto la realizzazione delle seguenti attività:

- Realizzazione di un lessico di *opinion word* (vedi par. 5.1)
- Implementazione del sistema di classificazione (vedi par. 5.2)
  - *Pre-processing* dei *tweet*
  - Analisi sintattica
  - Costruzione di un albero di dipendenze sintattiche
  - Etichettatura lessicale e calcolo della polarità di ogni parola dei *tweet*

- Calcolo della polarità di ogni frase presente nei *tweet* per combinazione delle polarità di singole parole
- Calcolo della polarità dei *tweet* per combinazione di polarità di singole frasi

5.1 Lessico di *opinion word*

Il lessico utilizzato per l'implementazione del sistema è stato creato attraverso la selezione e l'annotazione automatica di parole a partire da diverse fonti in lingua italiana. A questi termini sono stati successivamente aggiunti e manualmente annotati i *polarity shifter*, cioè tutte quelle parole che, per loro natura, invertono la polarità delle altre.

5.1.1 Aggettivi e avverbi

Gli aggettivi e gli avverbi sono stati estratti dal dizionario italiano elettronico (De Mauro, 2000). Per ognuno di essi, sono state considerate le glosse dei vari sensi presenti nel dizionario stesso e sono state automaticamente classificate usando il sistema online di *Sentiment Analysis API* fornito da *Ai Applied*[1] che assegna polarità positiva in un intervallo [0.5, 1] o negativa in un intervallo [-1, -0.5].

5.1.2 Nomi e verbi

Oltre agli aggettivi e agli avverbi, fondamentali per la classificazione di testi soggettivi (Taboada *et al.*, 2011), sono stati considerati anche nomi e verbi estratti da Sentix (Basile, Nissim, 2013). Per l'assegnazione dei valori di polarità si è utilizzata la medesima procedura descritta in precedenza per aggettivi e avverbi.

[1] http://ai-applied.nl/sentiment-analysis-api

5.1.3 Verifica manuale

L'*opinion lexicon*, così automaticamente ottenuto ed etichettato, è stato poi controllato manualmente da una collega linguista per correggere le eventuali annotazioni sbagliate. A tal fine è stato assegnato, per ogni significato scorretto, un valore di polarità di "1.01" o "-1.01" così da poter distinguere le annotazioni manuali da quelle automatiche. Tutti i termini considerati, a priori, oggettivi sono stati eliminati e, comunque, non considerati nella creazione del lessico. Per queste parole è stato convenzionalmente considerato un valore di polarità pari a "0".

5.1.4 *Sentiment polarity shifter*

Ci sono diversi fenomeni linguistici che possono causare l'inversione di polarità di una parola o l'aumento della sua intensità semantica (Taboada *et al.*, 2011). Per questo motivo, per la realizzazione del lessico sono stati considerati "negatori" ed "intensificatori".

Negatori: parole come "non", "nessuno", "niente", "mai" invertono la polarità di altre parole. Ai negatori è stato assegnato un valore moltiplicativo di "-1". In una frase come "Non si vede bene", ad esempio, il termine "non" nega "bene" invertendone la sua polarità da "+0,76" a "-0,76".

Intensificatori: questi termini aumentano o diminuiscono il valore di polarità delle parole che "accompagnano" (Taboada *et al.*, 2011). Per ognuno di essi è stata manualmente assegnata una percentuale positiva o negativa; questa percentuale viene usata per amplificare o ridurre il valore di polarità di altre parole. Ad esempio "felice" ha uno *score* di +0.84 e "molto felice" diventa +0.84 x (1 + 0.25)= 1.05 mentre "grave" ha uno *score* di -0.7 e "poco grave" diventa -0.7 x (1 - 0.25) = -0.52.

La tabella 5.1 riporta alcuni intensificatori presenti nel *lexicon*.

Intensificatore	Valore
completamente	+0.75
drasticamente	+0.50
molto	+0.25
leggermente	-0.50
poco	-0.25
abbastanza	-0.15

Tabella 5.1: Percentuale di intensificazione di alcuni intensificatori positivi e negativi presenti nel lessico

5.1.5 Parole dipendenti dal contesto

Gran parte delle parole di una lingua non ha un valore positivo o negativo assegnabile a priori ma, al contrario, può assumere una polarità diversa a seconda del contesto in cui si trova (Liu, 2012). Si consideri, ad esempio, la parola "forte": in "maniere forti" ha un significato negativo mentre in "forte legame" ne ha uno positivo. Inoltre, alcune di queste parole sono oggettive in certi domini ma soggettive in altri. La parola "usignolo", ad esempio, è obiettiva in frasi come "Ieri ho visto un usignolo vicino al fiume" ma soggettiva se usata in modo metaforico come in "Giovanni canta benissimo, sembra un usignolo!".
Nella realizzazione del lessico annotato per il sistema, le diverse accezioni di queste parole non sono state considerate in quanto necessiterebbero di un più sofisticato e complesso lavoro di identificazione di metafore e di *word sense disambiguation* (WSD).

Per una trattazione più approfondita sulla realizzazione, sulle regole e sulle tecniche utilizzate per la costruzione del *lexicon* di questo sistema per la classificazione di polarità di *tweet* in lingua italiana si rimanda a (Rossi, 2014).

5.2 Implementazione del sistema

Il primo passo nella realizzazione del sistema è stata l'implementazione dell'algoritmo di classificazione proposto in (Basile, Nissim, 2013) che, a partire da corpora di *tweet* in lingua italiana (TWITA), assegna polarità positiva, negativa o neutra ai *tweet* tramite un *lexicon* di parole (Sentix) opportunamente etichettate per polarità tramite SentiWordNet, Multi-WordNet e WordNet. A partire dallo sviluppo, quindi, del medesimo algoritmo proposto in (Basile, Nissim, 2013) ma sul lessico descritto nel precedente paragrafo si è realizzata una prima implementazione del sistema che può essere riassunta nei seguenti passi:

1. Il sistema calcola il punteggio di polarità di ogni parola presente nel lessico come media dei punteggi dei diversi significati presenti per ogni lemma

2. Dato un *tweet*, il sistema assegna un punteggio di polarità (ottenuto al punto 1) ad ogni parola che lo compone ed il cui lemma è presente nel lessico

3. Il sistema calcola il punteggio di polarità di ogni *tweet* come somma dei punteggi di polarità delle singole *opinion word*: un punteggio maggiore di "0" indica una polarità positiva, un punteggio minore di "0" ne indica una negativa, "0" indica un *tweet* neutro.

A partire dai risultati ottenuti da questa esperienza è stato successivamente sviluppato un sistema più complesso guidato dall'analisi sintattica e da una combinazione di tecniche di propagazione della polarità che considerano intensificatori e negatori opportunamente etichettati nel lessico stesso.

5.2.1 *Pre-processing* dei *tweet*

Prima di procedere con l'analisi sintattica, con la costruzione degli alberi sintattici e con la propagazione delle polarità lungo gli stessi, sono state

Etichetta	***Emoticon***
EMOPOS	(: :) :] [: :-) (-: [-: :-] (; ;) ;] [; ;-) (-; [-; ;-] :-D :D :-p :p (=: ;=D :=) :S @-) XD
EMONEG	:( ): :-( )-: ;( ); :-[ ]-: ;-( )-; :'[:'( )':]: :[:| :/ |: /: :=(:=| :=[xo :| D: O:

Tabella 5.2: *Emoticon* considerate nella fase di *pre-processing* dei *tweet*.

applicate diverse regole di sostituzione ed eliminazione di tutte le parti testuali considerate irrilevanti per la classificazione e che possono influenzare il *POS-tagging*, la lemmatizzazione ed il *parser*. In particolare:

- Tutte le url di siti internet sono state sostituite con una etichetta generica "URL"
- I caratteri "#" e "@" sono stati rimossi dagli hashtag (#abc) e *mention* (@abc)
- Le emoticon positive e negative (vedi tabella 5.2) sono state sostituite con una etichetta generica "EMOPOS" ed "EMONEG" rispettivamente

Poiché, in molti *tweet*, *hashtag* e *mention* sono parti integranti del discorso, si è deciso di non eliminarli o sostituirli completamente ma di rimuoverne solamente le etichette del *tag*. In questo modo si è riusciti a preservare il significato complessivo di questi *tweet*. Inoltre, al fine di considerare nel punteggio di polarità di un *tweet* anche le *emoticon*, sono state aggiunte al *lexicon* anche le etichette "EMOPOS" ed "EMONEG" con punteggio "+1" e "-1" rispettivamente.

La tabella 5.3 riporta alcuni *tweet* del *training set* di EVALITA 2014 (vedi par. 7.2) prima e dopo la fase di *pre-processing*. Nei primi due *tweet*

***Tweet* originale**	***Tweet* filtrato**
La #costamagna dice che c'è un'intervista esclusiva a #Grillo. Ma se ieri l'hanno intervistato quasi tutti... #Robinson	La costamagna dice che c'è un'intervista esclusiva a Grillo. Ma se ieri l'hanno intervistato quasi tutti... Robinson
Analisti, sondaggisti, giornalisti con #Grillo date il meglio e il peggio di voi... Tutti a casa! Siete inutili e dannosi! @beppegrillo	Analisti, sondaggisti, giornalisti con Grillo date il meglio e il peggio di voi... Tutti a casa! Siete inutili e dannosi! beppegrillo
@Zziagenio78 @miaceran @LiaCeli le delusioni capitano a tutti. non arrenderti! :)	Zziagenio78 miaceran LiaCeli le delusioni capitano a tutti. non arrenderti! EMOPOS
ad oggi Autoalcentesimo ha fornito 1100 euro di bonus carburante per soli 33.88€.... anche noi nel nostro piccolo... http:// fb.me/1ohGjx4rn	ad oggi Autoalcentesimo ha fornito 1100 euro di bonus carburante per soli 33.88€.... anche noi nel nostro piccolo... URL

Tabella 5.3: *Tweet* del *training set* di EVALITA 2014 prima e dopo la fase di *pre-processing*.

presentati si può osservare come la rimozione di "#" e "@" abbia preservato il significato dell'intero *tweet*.

5.2.2 Analisi sintattica

Il sistema fa affidamento al parser TULE (Lesmo, 2007) per l'analisi sintattica della struttura dei *tweet*. TULE include un tokenizzatore, un analizzatore morfologico, un *POS-tagger* ed un *parser* a dipendenze: a partire da singole frasi in linguaggio naturale restituisce, quindi, un grafo di dipendenze che ne descrive la struttura sintattica. Per ogni termine identificato, il *parser*

include anche il suo *POS-tag*, il lemma e altre informazioni morfologiche. Per via della struttura e della composizione stessa dei *tweet*, si sono riscontrate alcune difficoltà nell'utilizzo di TULE. Per ovviare a ciò, si è resa necessaria l'aggiunta di un ulteriore *step* di *pre-processing* del testo che consiste nella sostituzione di eventuali caratteri speciali quali "$", "€", "£" con la loro equivalente parola in italiano: "dollaro", "euro", "sterlina".

Di seguito (figura 5.1) un esempio di output generato da TULE per il *tweet* *"Grillo. Mi fa paura la gente che urla. Ne abbiamo già visti almeno un paio, ed è finita com'è finita. Niente urla per me, grazie."*:

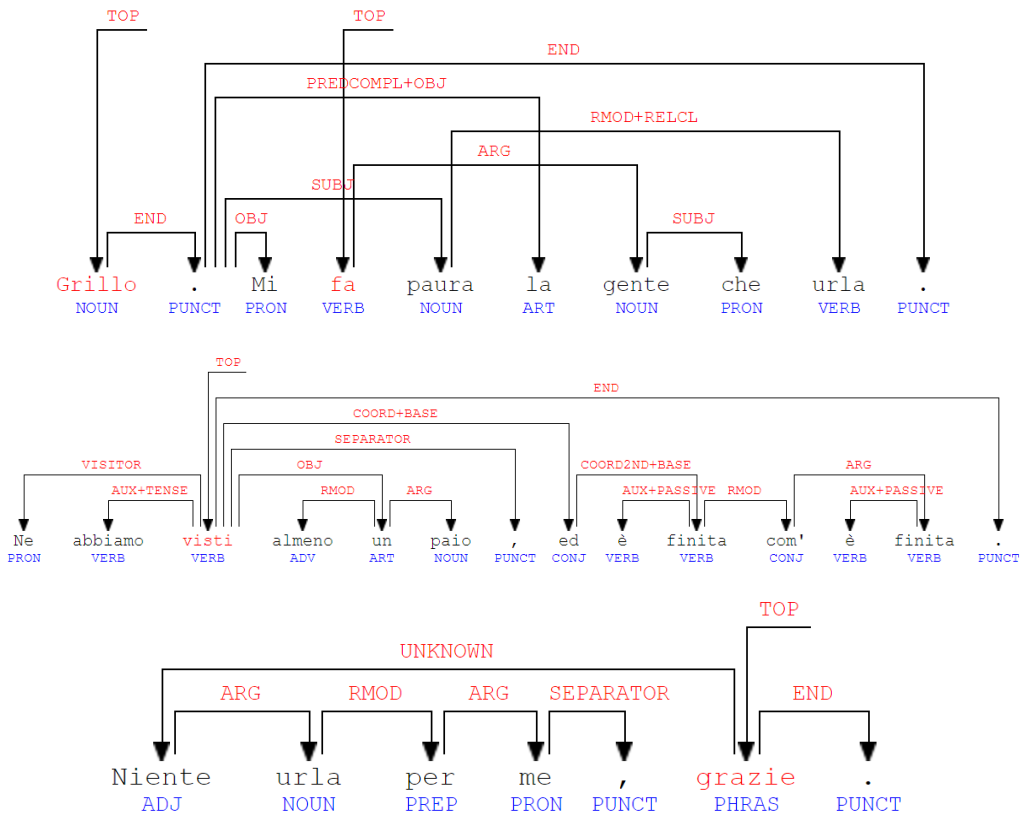


Figura 5.1: Esempio di grafo sintattico generato da TULE.

5.2.3 Algoritmo di classificazione

Il sistema usa liste di adiacenze (implementate tramite la libreria Boost[2]) per la rappresentazione degli alberi delle dipendenze sintattiche generati da TULE. Ogni nodo rappresenta una parola del *tweet* e contiene le seguenti informazioni: *POS-tag*, lemma (ottenuti da TULE), categoria lessicale (negatore o intensificatore) e punteggio di polarità calcolato a partire dal lessico annotato.

Per assegnare il punteggio di polarità ad una parola, il sistema, verifica la presenza o meno del suo lemma nel *lexicon*:

- Se il lemma non è presente viene convenzionalmente assegnato un punteggio pari a "0"
- Se il lemma è presente ma è annotato come un *polarity shifter*: il valore di polarità assegnato è la sua percentuale di intenficazione o il suo valore di negazione
- Se il lemma è presente e non è un *polarity shifter*: il valore di polarità assegnato è dato dalla media dei punteggi dei diversi significati del lemma presenti nel *lexicon*

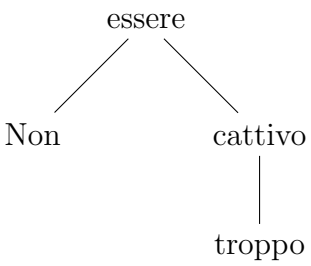
Una volta costruito l'intero albero sintattico ed assegnato un punteggio di polarità ad ogni nodo, il sistema classifica il *tweet* applicando una serie di regole di propagazione attraverso le dipendenze sintattiche rappresentate nell'albero stesso:

- Se nessun nodo dell'albero è un *polarity shifter* il punteggio è ottenuto sommando la polarità di ognuno di essi
- Se uno o più nodi sono *polarity shifter* è necessario calcolare la polarità considerando l'aumento, la diminuzione o l'inversione di polarità dei nodi collegati ad essi. Quindi, a partire dal *polarity shifter* più vicino

[2]http://www.boost.org

alle foglie dell'albero delle dipendenze, il sistema somma il punteggio di polarità dei nodi ad esso collegati e ne moltiplica il valore con quello di intensificazione del relativo intensificatore o negatore

Ad esempio, il punteggio di polarità (PS) della frase *"Non essere troppo cattivo"* è così ottenuto:



$$[(\mathrm{PS}(\textit{cattivo}) \cdot (\mathrm{PS}(\textit{troppo}) + 1)\ ) + \mathrm{PS}(\textit{essere})] \cdot \mathrm{PS}(\textit{Non})$$

Poiché un *tweet* può essere composto da più di una frase (e, quindi, da più di una struttura sintattica a dipendenze), il valore di polarità complessivo è ottenuto sommando il valore di polarità calcolato su ognuno degli alberi che lo compone.

Alla fine, il sistema classifica ogni *tweet* come:

Positivo se il punteggio di polarità è maggiore di "0"

Negativo se il punteggio di polarità è minore di "0"

Neutro se il punteggio di polarità è uguale a "0"

Per i risultati ottenuti dal sistema e la classificazione dello stesso in EVALITA 2014 si rimanda al capitolo 7.

Capitolo 6

Secondo sistema: l'approccio basato sull'apprendimento automatico

In questo capitolo si descrive la progettazione di un sistema automatico per la classificazione della *sentiment polarity classification* di *tweet* per la lingua italiana con un approccio basato su algoritmi stocastico/statistici di *machine learning*.

Il sistema nasce sfruttando gli strumenti costruiti ed utilizzati in *FICLIT+CS@UniBO System* e realizza una classificazione basata su un algoritmo di *Support Vector Machines* tramite LIBSVM (Chang, Lin, 2011) a partire dal lavoro presentato da (Cimino *et al.*, 2014) per il *task* SENTIPOLC della campagna di valutazione EVALITA 2014.

Il lavoro di progettazione e di sviluppo del sistema ha previsto le seguenti attività:

- Definizione delle *feature* da estrarre dai *tweet* (vedi par. 6.1)
- Implementazione di un algoritmo di classificazione automatica di *tweet* (vedi par. 6.2)

- *Pre-processing* dei *tweet*
- Analisi sintattica
- Costruzione di un albero di dipendenze sintattiche
- Etichettatura lessicale e calcolo della polarità di ogni parola del *tweet*
- Estrazione delle *feature* lessicali, morfosintattiche e dipendenti dal lessico
- *training*
- *test*

6.1 Le *feature* del sistema

Per consentire l'apprendimento automatico e la costruzione di un modello generale per la classificazione, ogni *tweet* deve essere trasformato in un vettore di *feature* significative da fornire all'algoritmo di SVM (vedi par. 3.1.2).

A tal fine è stato definito un insieme di *feature* così classificabili:

- *feature* lessicali
- *feature* morfosintattiche
- *feature* dipendenti dal *lexicon*

Tale categorizzazione segue i diversi livelli di analisi linguistica automatica realizzata attraverso il parser TULE ed il lessico del sistema FICLIT+CS@UniBO. Nel paragrafi seguenti si riportano le *feature* utilizzate dal sistema divise per categoria e, tra parentesi quadre, i loro nomi così come riportati nella tabella riassuntiva 6.1.

6.1.1 Feature lessicali

Lunghezza media numero di blocchi di al più 5 parole presenti nel *tweet* [AVERAGE_TWEET_LENGTH]

N-grammi di parole presenza o assenza di una sequenza di parole consecutive nel *tweet* [NGRAMS_TOKEN]

N-grammi di lemmi presenza o assenza di una sequenza di lemmi consecutivi nel *tweet* [NGRAMS_LEMMA]

Punteggiatura presenza o assenza di un "?" o di un "!" al termine del *tweet* [FINISHES_WITH_PUNCTUATION]

Emoticon presenza o assenza di una o più *emoticon* positive [SNT_EMOTICONS_POS] o negative [SNT_EMOTICONS_NEG] nel *tweet*

6.1.2 Feature morfosintattiche

N-grammi di parti del discorso presenza o assenza di una sequenza di parti del discorso consecutive nel *tweet*, corrispondenti alle principali categorie sintattiche (nome, verbo, aggettivo, avverbio) [NGRAMS_POS]

Distribuzione delle parti del discorso percentuale di distribuzione di nomi [POS_DISTR_PERC_NOUN], aggettivi [POS_DISTR_PERC_ADJ], avverbi [POS_DISTR_PERC_ADV] e numeri [POS_DISTR_PERC_NUM] all'interno del *tweet*

6.1.3 Feature dipendenti dal *lexicon*

N-grammi di opinion word per ogni n-gramma di lemma consecutivi presenti nel tweet, si verifica la polarità di ognuno di essi estraendola, ove presente, dal *lexicon*. Ad esempio, il trigramma "tutto molto interessante" è etichettato come "ABSENT_POS_POS". [NGRAMS_SNT]

Intensificatori presenza o assenza di parole che aumentano o diminuiscono il punteggio di polarità di altre parole presenti nel *tweet*. L'informazione è ottenuta, ove presente, dal *lexicon*. [HAS_INTENSIFIER]

Negatori presenza o assenza di parole che invertono il punteggio di polarità di una o più parole presenti nel *tweet*. L'informazione è ottenuta, ove presente, dal *lexicon* [HAS_NEGATIVE]

Modificatori di polarità per ogni lemma del *tweet* si verifica la presenza di intensificatori o negatori, tramite l'albero delle dipendenze sintattiche, che ne cambiano il valore di polarità. In tal caso, il valore della *feature* si è ottenuta concatenando l'intensificatore/negatore e la polarità del lemma. Ad esempio, il bigramma "non interessante" diventa "non_POS" mentre "molto brutto" diventa "molto_NEG". [SNT_WITH_MODIFIER]

PMI per ogni unigramma, bigramma, trigramma e quadrigramma di parole presenti nel *tweet* si calcola un punteggio ottenuto sommando le polarità dei singoli termini presenti nell'n-gramma. Per ogni n-gramma si considera solamente il valore minimo ed il valore massimo approssimato all'intero più vicino [PMI_SCORE]

Distribuzione della polarità questa *feature* calcola la percentuale di parole positive [SNT_DISTRIBUTION_POS] e negative [SNT_DISTRIBUTION_NEG] presenti nel *tweet*. Ogni valore è arrotondato al più vicino multiplo di 5

Polarità più frequente questa *feature* indica la polarità più frequente tra i lemmi presenti nel *tweet*: positiva [SNT_MAJORITY_POS] o negativa [SNT_MAJORITY_NEG]

Polarità più frequente in sezioni diviso il *tweet* in tre parti uguali, questa *feature* indica la polarità più frequente in ognuna delle tre parti usando la polarità dei lemmi presenti nel *lexicon* [SNT_POSITION_PRESENCE]

Punteggio dal lexicon è il punteggio ottenuto sommando le singole polarità a priori indicate nel *lexicon* di tutte le parole presenti nel *tweet* [CUSTOM_SCORE]

6.2 Implementazione del sistema

Il sistema realizzato, a partire dai *tweet* di un *training set* manualmente etichettato, costruisce un modello capace di generalizzare la classificazione di polarità di qualsiasi *tweet* in lingua italiana.

Il primo passo per la realizzazione di questo classificatore è stato quello di semplificare il più possibile l'estrazione delle *feature* da ogni *tweet*. Per far ciò si sono seguite le medesime procedure realizzate in *FICLIC+CS@Unibo System* per il *pre-processing* dei testi (vedi par. 5.2.1) e l'analisi sintattica (vedi par. 5.2.2) attraverso il parser TULE. Inoltre, così come il precedente sistema basato sul lessico, anche questo classificatore usa liste di adiacenze (basate sulla libreria Boost) per rappresentare gli alberi delle dipendenze sintattiche: ogni nodo rappresenta una parola del *tweet* e contiene il *POS-tag*, il lemma, la categoria lessicale (negatore o intensificatore) ed il punteggio di polarità calcolato a partire dal *lexicon*.

6.2.1 Estrazione delle *feature*

Il sistema, una volta costruito l'albero che rappresenta il *tweet* ed assegnato un punteggio di polarità ad ogni nodo, procede con la costruzione degli n-grammi e l'estrazione delle *feature* descritte nel paragrafo precedente.

A partire dall'albero sintattico di ogni *tweet*, quindi, il sistema compone un vettore (*tweet_words*) di parole privato di punteggiatura, preposizioni ed articoli poiché irrilevanti ai fini della *sentiment polarity classification*.

```
boost::graph_traits<Tree>::vertex_iterator vi, vi_end;
// Per ogni nodo dell'albero sintattico
for (boost::tie(vi, vi_end) = boost::vertices(t); vi != vi_end; vi++) {
     if (t[*vi].token_id != 0) {
     //Ignoro punteggiatura, preposizioni, articoli
          if (t[*vi].pos != ``PUNCT'' &&
              t[*vi].pos != ''PREP'' &&
              t[*vi].pos != ''AR'') {
```

***Feature* lessicali**	
Nome	**Tipo**
NGRAMS_TOKEN	Presenza/Assenza
NGRAMS_LEMMA	Presenza/Assenza
FINISHES_WITH_PUNCTUATION	Presenza/Assenza
SNT_EMOTICONS_POS	Presenza/Assenza
SNT_EMOTICONS_NEG	Presenza/Assenza
AVERAGE_TWEET_LENGTH	Valore

***Feature* morfosintattiche**	
Nome	**Tipo**
NGRAMS_POS	Presenza/Assenza
POS_DISTR_PERC_NOUN	Valore
POS_DISTR_PERC_ADJ	Valore
POS_DISTR_PERC_ADV	Valore
POS_DISTR_PERC_NUM	Valore

***Feature* dipendenti dal lessico**	
Nome	**Tipo**
NGRAMS_SNT	Presenza/Assenza
HAS_INTENSIFIER	Presenza/Assenza
HAS_NEGATIVE	Presenza/Assenza
SNT_WITH_MODIFIER	Presenza/Assenza
SNT_MAJORITY_POS	Presenza/Assenza
SNT_MAJORITY_NEG	Presenza/Assenza
SNT_POSITION_PRESENCE	Presenza/Assenza
PMI_SCORE	Valore
SNT_DISTRIBUTION_POS	Valore
SNT_DISTRIBUTION_NEG	Valore
CUSTOM_SCORE	Valore

Tabella 6.1: *Feature* del sistema basato sull'apprendimento automatico

```
                // inserisco parola nel vettore
                tweet_words.push_back(t[*vi]);
            }
        }
}
```

E, a partire dal vettore così composto, costruisce uni-grammi, bi-grammi, tri-grammi e quadri-grammi.

```
for (int i = 0; i <= tweet_words.size() - 1; i++) {
    std::copy(tweet_words.begin() + i,
                tweet_words.begin() + (i + 1),
                std::back_inserter(vector_words));
    // vettore di unigrammi
    tweet_1grams[(*itr1).first].push_back(vector_words);
    vector_words.clear();
}
```

Per ogni n-gramma, il sistema estrae le *feature* lessicali (n-grammi di parole, n-grammi di lemmi), morfosintattiche (n-grammi di parti del discorso) e dipendenti dal *lexicon* (n-grammi di *opinion word*, PMI) utilizzando l'albero sintattico stesso ed i valori contenuti in ogni suo nodo.

Terminata l'estrazione delle *feature* degli n-grammi, procede con l'estrazione delle restanti *feature* sull'intero *tweet*. Ad ogni *feature* assegna, successivamente, un peso. Si possono contraddistinguere:

- *feature* di tipo presenza/assenza alle quali viene assegnato un peso di "1" o "0" rispettivamente
- *feature* che assumono valori discreti alle quali viene assegnato un valore numerico opportunamente calcolato

Di seguito il calcolo e la valorizzazione di alcune *feature*.

Punteggiatura:

```
if (tweet_words[tweet_words.size() - 1].token == "!" ||
    tweet_words[tweet_words.size() - 1].token == "?")
        ++features_counts[(*itr1).first]["FINISHES_WITH_PUNCTUATION"];
```

Polarità più frequente

```
if (positivelemma_count > negativelemma_count)
     features_counts[(*itr1).first]["SNT_MAJORITY_POS"] = 1;
else if (positivelemma_count < negativelemma_count)
     features_counts[(*itr1).first]["SNT_MAJORITY_NEG"] = 1;
```

Lunghezza media del *tweet*

```
features_counts[(*itr1).first]["AVERAGE_TWEET_LENGTH"] = tweet_words.size() / 5;
```

Feature degli n-grammi

```
void computeSingleNGramFeatures(map<string,
deque<vector<Word>>>&tweet_Ngrams,
map<string, map<string, vector<string>>>&features_Ngrams) {
     map<string, deque<vector<Word>>>::iterator itr1;
     for (itr1 = tweet_Ngrams.begin(); itr1 != tweet_Ngrams.end(); itr1++){
          deque<vector<Word>> vector_ngrams = (*itr1).second;
          double polarity_sum = 0.0;
          for (int j = 0; j <= vector_ngrams.size() - 1; j++) {
               string ngram_snt_str = "";
               string tweet_Ngrams_token = "";
               string tweet_Ngrams_lemma = "";
               string tweet_Ngrams_pos = "";
               for (int k = 0; k <= vector_ngrams[j].size() - 1; k++) {
                    if (vector_ngrams[j][k].polarity > 0 &&
                        !PoPaTree::isNegative(vector_ngrams[j][k]))
                             ngram_snt_str += "POS_";
                    else if (vector_ngrams[j][k].polarity < 0 &&
                                !PoPaTree::isNegative(vector_ngrams[j][k]))
                                     ngram_snt_str += "NEG_";
                   else ngram_snt_str += "ABSENT_";
                   polarity_sum += vector_ngrams[j][k].polarity;
                   tweet_Ngrams_token += vector_ngrams[j][k].token + "_";
                   tweet_Ngrams_lemma += vector_ngrams[j][k].lemma + "_";
                   tweet_Ngrams_pos += vector_ngrams[j][k].pos + "_";
               }
            features_Ngrams[(*itr1).first]["NGRAMS_SNT_" + ngram_snt_str].push_back("1");
```

```
            features_Ngrams[(*itr1).first]["NGRAMS_TOKEN_" + tweet_Ngrams_token].push_back("1");
            features_Ngrams[(*itr1).first]["NGRAMS_LEMMA_" + tweet_Ngrams_lemma].push_back("1");
            features_Ngrams[(*itr1).first]["NGRAMS_POS_" + tweet_Ngrams_pos].push_back("1");
          }
     }
}
```

Terminato questo processo, ad ogni *tweet* è possibile ricondurre il suo vettore di *feature*.

6.2.2 Apprendimento automatico e LIBSVM

Terminato il processo di estrazione delle feature, il sistema provvede alla generazione dei dati da somministrare all'algoritmo di SVM per la generazione del modello di generalizzazione.

Poiché, come già affermato in precedenza, ogni vettore di *feature* rappresenta un *tweet* del *training set* etichettato, il sistema assegna ad ognuno di essi l'etichetta di polarità del *tweet* di riferimento. Al termine di tale processo, genera un file che diventerà l'*input* dell'algoritmo di SVM di LISBSVM.

Di seguito un esempio: ogni riga è un *tweet* del *training set*, il primo valore indica l'etichetta di polarità ("1" per positivo, "0" per neutro, "-1" per negativo) mentre i successivi valori sono l'etichetta ed il peso delle *feature* separati da ":".

-1 199:1 330:2 342:1 357:1 458:1 471:1 519:1 539:1 615:1 686:1 694:1 757:1 775:1 863:1 873:1 970:1 990:1 1040:1 1050:1 1124:1 1147:1 1156:1 1179:1 1218:1 1231:1 1271:1 1313:1 1469:1 1499:1 1500:1 1613:2 1663:1 1726:1 1752:9 1756:2 1772:1 1807:1 1884:1 1969:1 2008:1 2056:15 2143:1 2164:1 2196:1 2212:1 2218:1 2223:1 2300:1 2341:1 2362:1 2441:1
1 64:1 159:1 170:1 249:1 256:1 357:2 494:1 535:1 585:1 695:1 700:1 752:1 760:1 782:1 809:1 824:1 863:1 944:1 976:1 1008:1 1048:1 1049:1 1050:2 1181:1 1194:1 1209:1 1334:1 1336:1 1364:1 1416:1 1448:1 1491:1 1525:1 1562:1 1575:1 1615:1 1647:1 1700:1 1716:1 1752:9 1791:1 2056:9 2112:1 2201:1 2232:1 2256:1 2278:1 2300:1 2336:1

0 17:1 159:2 172:1 206:1 237:1 262:1 276:1 313:1 357:1 427:1 443:1 494:4 555:1 701:1 775:1 782:1 800:1 853:1 938:1 972:1 1050:1 1235:1 1362:1 1413:1 1420:1 1430:1 1477:1 1563:1 1575:1 1587:1 1646:1 1714:1 1752:5 1806:1 1848:1 1863:1 1902:1 1910:1 2056:9 2245:1 2300:1 2305:1 2406:1 2507:1 2560:1 2563:3 2622:2 2655:3 2716:1 2773:1 2860:1

Come si può osservare, ogni etichetta di *feature*, per rispettare la sintassi di LIBSVM, è stata trasformata in un valore numerico tramite un'apposita funzione di *hash*.

LISBSVM, a partire dal *training set* così trasformato e sfruttando una classificazione di tipo lineare, genera un modello capace di generalizzare la classificazione di polarità di un qualsiasi *tweet* in lingua italiana.

Per i risultati ottenuti dal sistema finora descritto si rimanda al capitolo 7.

Capitolo 7

Risultati dei sistemi

7.1 Valutazione di un classificatore

A causa della natura solitamente soggettiva dei problemi di classificazione, generalmente, la valutazione delle prestazioni di un qualsiasi classificatore di testi è effettuata in modo sperimentale, misurando la sua efficacia ossia la capacità di prendere decisioni corrette durante il processo di classificazione.

Al termine del processo di classificazione, per ogni classe (c), è possibile raggruppare i testi nel seguente modo:

TP (*true positive*, veri positivi) numero di documenti/testi inseriti correttamente nella classe c;

FP (*false positive*, falsi positivi) numero di documenti/testi inseriti erroneamente nella classe c;

FN (*false negative*, falsi negativi) numero di documenti/testi erroneamente non inseriti sotto la classe c;

TN (*true negative*, veri negativi) numero di documenti/testi correttamente non inseriti nella classe c.

Il diagramma 7.1 illustra graficamente questi raggruppamenti utili per definire le metriche fondamentali per la valutazione di un qualsiasi siste-

ma automatico per la *sentiment analysis*: accuratezza (*accuracy*), precisione (*precision*), *recall* ed *F-score*.

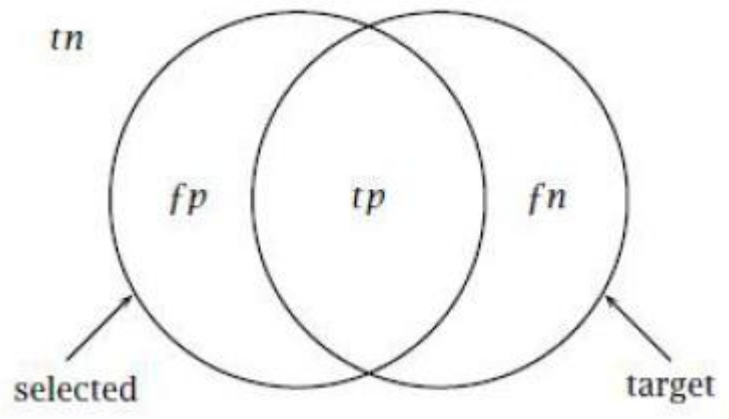


Figura 7.1: Diagramma per la definzione delle metriche di valutazione

Tali metriche sono calcolate nel seguente modo:

$$accuracy = \frac{tp+tn}{tp+fp+fn+tn}$$

$$precision = \frac{tp}{tp+fp}$$

$$recall = \frac{tp}{tp+fn}$$

$$F = \frac{2*precision*recall}{precision+recall}$$

Ognuna di esse permette di misurare diversi aspetti della classificazione:

- L'**accuratezza** misura la percentuale di documenti/testi correttamente classificati
- La **precisione** misura la correttezza del classificatore in termini di percentuale di documenti/testi correttamente etichettati in una certa classe rispetto al totale di documenti etichettati in tale classe

- **Recall** misura la completezza del classificatore in termini di percentuale di documenti/testi correttamente etichettati in una classe rispetto al totale di documenti che avrebbe dovuto classificare in tale classe
- **F-score** è la media geometrica tra precisione e *recall* e consente di avere una buona misura di correttezza e completezza totale del classificatore

Nei seguenti paragrafi saranno descritti i risultati ottenuti dai due sistemi implementati in questo lavoro di tesi usando le metriche appena descritte.

7.2 Il *training set* di EVALITA 2014

Per la fase di test e verifica del sistema FICLIT+CS@Unibo nonché per la fase di *training* e di valutazione del sistema basato sull'apprendimento automatico è stato utilizzato il *training set* del *task* SENTIPOLC di EVALITA 2014 composto da circa 4000 *tweet* manualmente etichettati su quattro classi: polarità positiva, polarità neutra, polarità negativa, polarità mista.

Ogni riga del *training set* contiene: un identificativo, il testo stesso del *tweet* e la relativa classificazione manuale per i diversi *task* previsti nella campagna di valutazione. Per il *task* sulla *sentiment polarity classification* sono da considerare le etichette "pos" e "neg" manualmente valorizzate con "1" o con "0" a seconda che il *tweet* sia positivo o negativo. Una valorizzazione di entrambi i valori a "0" indica un *tweet* con polarità neutra mentre una valorizzazione di entrambi ad "1" ne indica uno con polarità mista (sia negativa sia positiva)

Di seguito alcuni esempi:

"idtwitter","subj", *"pos", "neg"*,"iro","top","text"

"193201249095651328","1", *"0", "1"*,"0","1","#Grillo Mi fa paura la gente che urla. Ne abbiamo già visti almeno un paio, ed è finita com'è finita. Niente urla per me, grazie."

"190814691994513409","1", *"1", "0"*,"" "0","@theblackproject grazie per il remix, bellissimo!"

"396633044293287936" "0", *"0", "0"*,"0","0","@Pietersenvgdr cerca di parlarci di più, così dovrà salutarti per forza."

"189677525717360640","1", *"1", "1"*,"0","0","@naripolpetta Si sta scrivendo da sola, praticamente. Anche se non so quando la finirò. Ma la finirò."

La tabella 7.1 mostra la composizione del *training set* di EVALITA 2014 utilizzato per la valutazione dei sistemi secondo l'etichettattura manuale sulle diverse classi previste dagli organizzatori per la classificazione del sentimento.

Pol. positiva	**Pol. neutra**	**Pol. negativa**	**Pol. mista**
861	1414	1471	293

Tabella 7.1: Composizione del *training set* del *task* SENTIPOLC di EVALITA 2014

7.3 Risultati del sistema basato sul *lexicon* (FICLIT+CS@UniBO System)

La tabella 7.2 mostra i risultati del sistema sul *training set* del task SENTPOLC di EVALITA 2014 per le due implementazioni del sistema descritte nel capitolo 5.

L'implementazione del sistema che usa alberi sintattici è stata analizzata sia con l'utilizzo di intensificatori e negatori sia con l'esclusione di uno o di entrambi.

I risultati, quindi, mostrano le prestazioni del sistema con:

- la sola somma di polarità a priori dei termini del *lexicon* (*Solo somme*)
- l'uso sia di intensificatori sia di negatori, con propagazione di polarità lungo alberi sintattici (*Parsing con soli intensificatori*)
- l'uso di intensificatori e non di negatori, con propagazione di polarità lungo alberi sintattici (*Parsing con soli intensificatori*)
- l'uso di negatori e non di intensificatori, con propagazione di polarità lungo alberi sintattici (*Parsing con soli negatori*)

Come riportato nel par. 7.2 , il *training set* di EVALITA 2014 comprende *tweet* di polarità mista che sono stati esclusi in questa analisi poiché il sistema implementato attua una classificazioni su sole tre classi: "positiva", "negativa", "neutra".

	Solo somme	**Parsing completo**	**Parsing con soli negatori**	**Parsing con soli intensificatori**
F-score	0,510	0,512	0,523	0,529

Tabella 7.2: F-score delle diverse implementazioni di *FICLIT+CS@Unibo System* sul *training set* di EVALITA 2014

Per tutte le implementazioni precedentemente descritte, durante la fase di progettazione e studio del sistema, si è, inoltre, analizzata la possibilità di escludere alcune categorie di *POS-tag* dal processo di classificazione se risultanti inefficienti. Come si può osservare dalla tabella 7.3, che mostra i risultati con diverse combinazioni di *POS-tag* (N=nomi, V=verbi, A=aggettivi, R=avverbi), i migliori risultati, però, si sono ottenuti utilizzando tutte le parti del discorso.

I risultati, infatti, dimostrano come l'utilizzo di tutte le parti del discorso si riveli estremamente importante nell'implementazione del sistema che

POS tag	**Solo somme**	**Parsing completo**	**Parsing con soli negatori**	**Parsing con soli intensificatori**
AR	0,482	0,462	0,465	0,475
A	0,458	0,454	0,460	0,453
NAR	0,513	0,490	0,493	0,507
NA	0,496	0,492	0,497	0,491
NR	0,480	0,458	0,460	0,474
N	0,467	0,464	0,466	0,465
NVAR	*0,510*	*0,512*	*0,523*	*0,529*
NVA	0,518	0,514	0,518	0,514
NVR	0,496	0,480	0,481	0,492
NV	0,492	0,491	0,492	0,491
R	0,438	0,418	0,411	0,437
VAR	0,498	0,485	0,486	0,494
VA	0,482	0,479	0,482	0,479
VR	0,455	0,443	0,441	0,452
V	0,442	0,442	0,442	0,442

Tabella 7.3: F-score, secondo diversi raggruppamenti di *POS-tag*, delle varie implementazioni di FICLIT+CS@Unibo sul *training set* di EVALITA 2014

usa, per la classificazione, le sole somme di polarità seppur l'utilizzo di soli nomi, verbi e aggettivi (NVA) consenta di ottenere un *F-score* leggermente migliore (51.8% *vs* 51%). Dai risultati ottenuti dall sistema che usa il *parser* sintattico e la propagazione di polarità, invece, si evince come l'uso di soli intensificatori consenta di ottenere, a prescindere dai *POS-tag* utilizzati, una migliore classificazione sia rispetto l'utilizzo di soli negatori sia rispetto l'utilizzo combinato dei due tipi di *polarity shifter*. L'uso combinato sia di negatori sia di intensificatori sembrebbe, infatti, non comportare un miglioramento di *performance* rispetto all'uso di uno solo dei due "modificatori di polarità".

Posizione	Combined F-score	Pol. Pos. F-score	Pol. Neg. F-score
1	0.6771	0.6752	0.6789
2	0.6347	0.6196	0.6498
3	0.6312	0.6352	0.6271
4	0.6299	0.6277	0.6321
-	<u>0.6062</u>	<u>0.5941</u>	<u>0.6184</u>
5	0.6049	0.6079	0.6019
6	0.6026	0.6153	0.5899
7	**0.5980**	**0.5940**	**0.6019**
8	0.5626	0.5556	0.5695
9	0.5342	0.5293	0.5390
10	0.5181	0.5021	0.5341
11	0.5086	0.5159	0.5013
12	*0.3718*	*0.3977*	*0.3459*

Tabella 7.4: Risultati ufficiali *task* SENTIPOLC di EVALITA 2014 sul *test set* fornito dagli organizzatori. In grassetto i risultati ufficiali del sistema FICLIT+CS@Unibo con la propagazione di polarità che utilizza sia intensificatori sia negatori, in sottolineato i risultati del sistema che utilizza sole somme di polarità, in corsivo la *baseline*.

La Tabella 7.4, infine, riassume i risultati ufficiali di EVALITA 2014 per il *task* SENTIPOLC. I risultati sono stati ottenuti calcolando, sul *test set* della campagna di valutazione, il "*combined F-score*" fornito dagli organizzatori che, rispetto ai risultati mostrati in precedenza, considera anche le polarità miste, assegnando un F-score di "0.5" qualora un *tweet* misto sia etichettato dal sistema solo positivo o solo negativo, "0" se etichettato neutro ed "1" se correttamente classificato con polarità mista.

FICLIT+CS@Unibo System, realizzato in soli tre mesi per poter partecipare alla campagna di valutazione, si è posizionato settimo (con un "*com-*

bined F-score" del 59.8 %) su undici partecipanti. Come si può osservare, i risultati ottenuti con la versione del sistema che combina i dati del lessico annotato con l'algoritmo di propagazione della polarità lungo gli alberi delle dipendenze sintattiche non superano quelli ottenuti usando la sola somma delle polarità riportate nel *lexicon*. Il sistema, in quest'ultima versione, si posiziona quinto con un "*combined F-score*" che supera il 60%.

Il processo di propagazione della polarità presenta molte problematiche ma opportunamente analizzato e affinato potrebbe portare a risultati più affidabili. Inoltre poiché buona parte dei risultati ottenuti con il processo di propagazione dipendono fortemente dai lemmi e dalle annotazioni presenti nel lessico, è ragionevole pensare come l'aggiunta di nuovi lemmi ed un processo di annotazione più accurato possano consentire una migliore classificazione.

7.4 Risultati del sistema basato sull'apprendimento automatico

La tabella 7.5 riporta i risultati del sistema basato sull'apprendimento automatico sul *training set* del task SENTPOLC di EVALITA 2014 (vedi par. 7.2) al quale, come per i risultati mostrati per il precedente sistema, sono stati esclusi i *tweet* con polarità mista.

I risultati mostrano accuratezza, precision, *recall* ed *F-score* ottenuta da una *5-fold cross validation* secondo tre diverse combinazioni delle *feature* descritte nel paragrafo 6.1:

- Tutte le *feature* (*Full*)
- Solo *feature* dipendenti dal *lexicon* (*SoloLexicon*)
- Solo *feature* lessicali e morfosintattiche (*NoLexicon*)

Si può osservare come i risultati migliori si ottengano combiando sia le *feature* dipendenti sia quelle indipendenti del *lexicon*. Le sole *feature* lessicali e morfosintattiche considerate nella realizzazione del sistema non consentono,

	Full	*SoloLexicon*	*NoLexicon*
Precisione (*combined*)	0.572	0.561	0.345
Recall (*combined*)	0.546	0.528	0.438
F-score (*combined*)	**0.548**	0.531	0.381
Accuratezza	0.580	0.562	0.507

Tabella 7.5: Valutazione sul *training set* di EVALITA 2014 del sistema basato sull'apprendimento automatico secondo tre diverse combinazioni delle *feature* utilizzate.

invece, una buona classificazione del sentimento. Ciò è dovuto, in particolare, all'inefficacia o, piuttosto, all'insufficenza delle stesse a rappresentare le caratteristiche lessico-sintattiche dei *tweet* positivi del *training set*. L'algoritmo di SVM, in questo caso, non riesce a classificare correttamente i *tweet* positivi, generando un modello capace di etichettare solamente *tweet* negativi o neutrali (vedi tabella 7.6). Ovviamente tale difficoltà pregiudica l'intera classificazione con sole *feature* lessicali e morfosintattiche causando un *"combined F-score"* del 38.1%.

NoLexicon	**Positivo**	**Neutro**	**Negativo**
Precisione	0.0	0.563	0.471
Recall	*0.0*	0.574	0.739
F-score	0.0	0.568	0.576

Tabella 7.6: Valutazione (sul *training set* di EVALITA 2014), per ogni etichetta di classificazione, del sistema basato sull'apprendimento automatico che usa solo *feature* lessicali e morfosintattiche.

Diversamente, le sole *feature* basate sul *lexicon* si rivelano particolarmente efficaci per la *Twitter sentiment polarity classification* raggiungendo, da sole, un *"combined F-score"* di 0.531. La tabella 7.7 mostra i risultati ottentuti da tale sistema divisi per singola etichetta di classificazione.

SoloLexicon	**Positivo**	**Neutro**	**Negativo**
Precisione	0.557	0.530	0.595
Recall	*0.310*	0.615	0.659
F-score	0.398	0.570	0.625

Tabella 7.7: Valutazione (sul *training set* di EVALITA 2014), per ogni etichetta di classificazione, del sistema basato sull'apprendimento automatico che usa solo *feature* basate sul *lexicon*.

Full	**Positivo**	**Neutro**	**Negativo**
Precisione	0.547	0.566	0.604
Recall	*0.325*	0.626	0.686
F-score	0.408	0.595	0.642

Tabella 7.8: Valutazione (sul *training set* di EVALITA 2014), per ogni etichetta di classificazione, del sistema basato sull'apprendimento automatico che usa tutte le feature.

La tabella 7.8, infine, mostra i risultati, sempre distinti per ogni etichetta di classificazione, del sistema che utilizza tutte le *feature* descritte nel par. 6.1.

Dai risultati fin qui mostrati, si osserva come la *recall* per la classificazione dei *tweet* positivi (indicata in corsivo nelle tabelle precedenti) sia estremamente inferiore (quasi 50% in meno) rispetto alla *recall* di quelli neutri e negativi, a prescindere dalla combinazione delle *feature* utilizzata. Ciò conferma la necessità di aggiungere ulteriori *feature* al sistema per cercare di modellizzare meglio i *tweet* positivi e, quindi, consentire all'algoritmo di SVM di generare un modello più efficente per la classificazione.

Il sistema basato sull'apprendimento automatico non ha partecipato al *task* SENTPOLC di EVALITA 2014 poiché realizzato successivamente alla campagna di valutazione. Di seguito si mostrano comunque i risultati ottenu-

ti dal sistema sul *test set* di SENTIPOLC utilizzando il calcolo del *"combined F-score"* fornito dagli organizzazatori.

A partire dal *training set* di EVALITA 2014 (vedi par. 7.2) ed utilizzando sia le *feature* dipendenti sia le *feature* indipendenti dal lessico (vedi par. 6.1), si è generato un modello generalizzato per la classificazione dei *tweet* per la lingua italiana tramite LIBSVM. Il modello così costruito è stato successivamente impiegato per la classificazione dei *tweet* dell'intero *test set*. Quest'ultimo, infatti, è stato trasformato in una lista di vettori di *feature* (uno per ogni *tweet*) che opportunamente fornita, insieme al modello generalizzato precedentemente ottenuto, all'algoritmo di "*prediction*" di LIBSVM ha consentito la classificazione.

La tabella 7.9 mostra il *"combined F-score"*, che considera anche le polarità miste assegnando un *F-score* di "0.5" qualora un *tweet* misto sia etichettato dal sistema solo positivo o solo negativo, "0" se etichettato neutro ed "1" se correttamente classificato con polarità mista), oltre agli *F-score* ottenuti dal sistema per i *tweet* positivi e negativi.

Combined F-score	**Pol. Pos. F-score**	**Pol. Neg. F-score**
0.6207	0.5846	0.6568

Tabella 7.9: Risultati del sistema basato sull'apprendimento automatico sul *test set* del *task* SENTIPOLC di EVALITA 2014.

Come già rilevato dai risultati sul *training set*, il sistema basato sull'apprendimento automatico, anche sul *test set*, si comporta meglio con i *tweet* negativi (*"combined F-score"* del 65.68%) piuttosto che con quelli positivi (*"combined F-score"* del 58.46%).

Con un *"combined F-score"* complessivo del 62.07%, il sistema si sarebbe classificato in quinta posizione (vedi tabella 7.10) tra gli undici parteci-

panti alla campagna di valutazione sulla *sentiment polarity classification* di EVALITA 2014.

Posizione	**Combined F-score**	**Pol. Pos. F-score**	**Pol. Neg. F-score**
1	0.6771	0.6752	0.6789
2	0.6347	0.6196	0.6498
3	0.6312	0.6352	0.6271
4	0.6299	0.6277	0.6321
*	**0.6207**	**0.5846**	**0.6568**
5	0.6049	0.6079	0.6019
6	0.6026	0.6153	0.5899
7	0.5980	0.5940	0.6019
8	0.5626	0.5556	0.5695
9	0.5342	0.5293	0.5390
10	0.5181	0.5021	0.5341
11	0.5086	0.5159	0.5013
12	*0.3718*	*0.3977*	*0.3459*

Tabella 7.10: Classifica *task* SENTIPOLC di EVALITA 2014 con, in aggiunta, il sistema basato sull'apprendimento automatico. In grassetto i risultati del sistema basato sull'apprendimento automatico, in corsivo la *baseline*.

7.5 Confronto tra i risultati dei due sistemi

Si presenta, in tabella 7.11, uno specchietto riassuntivo degli *F-score* ottenuti, dai due sistemi realizzati per questo lavoro di tesi, sul *training set* di EVALITA 2014, così come già presentati nei precedenti paragrafi.

Dal confronto si osserva come il sistema basato sull'apprendimento automatico riporti un *F-score* combinato (sulle tre etichette della classificazione)

Sistema	Tipo implementazione	F-score
Sistema basato su apprendimento supervisionato	*Full*	**0.548**
	Solo lexicon	0.531
	No lexicon	0.381
Sistema FICLIT+CS@Unibo	*Solo somme*	0.510
	Parsing completo	0.512
	Parsing solo negatori	0.523
	Parsing solo intensificatori	0.529

Tabella 7.11: Confronto risultati dei due sistemi realizzati sul *training set* di EVALITA 2014 (con esclusione dei *tweet* etichettati con polarità mista).

migliore rispetto al sistema FICLIT+CS@Unibo sia nell'implementazione che usa tutte le *feature* sia in quella che utilizza le sole *feature* basate sul *lexicon*.

Questi risultati trovano conferma anche osservando il *test set* di EVALITA 2014 utilizzando il calcolo del "*combined F-score*" fornito dagli organizzatori (tabella 7.12).

Sistema	Combined F-score	Pol. Pos. F-score	Pol. Neg. F-score
Sistema basato su apprendimento supervisionato	**0.6207**	0.5846	0.6568
Sistema FICLIT+CS@Unibo	0.5980	0.5940	0.6019

Tabella 7.12: Confronto risultati dei due sistemi realizzati sul *test set* di EVALITA 2014.

Si riassume, infine, la classifica di EVALITA 2014 per il task SENTIPOLC (tabella 7.13) considerando entrambi i sistemi realizzati in questo lavoro di

Posizione	**Combined F-score**	**Pol. Pos. F-score**	**Pol. Neg. F-score**
1	0.6771	0.6752	0.6789
2	0.6347	0.6196	0.6498
3	0.6312	0.6352	0.6271
4	0.6299	0.6277	0.6321
*****	***0.6207***	***0.5846***	***0.6568***
-	0.6062	0.5941	0.6184
5	0.6049	0.6079	0.6019
6	0.6026	0.6153	0.5899
7	**0.5980**	**0.5940**	**0.6019**
8	0.5626	0.5556	0.5695
9	0.5342	0.5293	0.5390
10	0.5181	0.5021	0.5341
11	0.5086	0.5159	0.5013
12	*0.3718*	*0.3977*	*0.3459*

Tabella 7.13: Classifica *task* SENTIPOLC di EVALITA 2014 che include entrambi i sistemi realizzati. In grassetto e corsivo i risultati del sistema basato sull'apprendimento, in grassetto i risultati ufficiali del sistema FICLIT+CS@Unibo con la propagazione di polarità che utilizza sia intensificatori sia negatori, in sottolineato i risultati di FICLIT+CS@Unibo con sole somme di polarità, in corsivo la *baseline*.

tesi. Si osserva come il sistema basato sull'apprendimento supervisionato si posizioni in quinta posizione con un "*combined F-score*" del 62.07%, facendo meglio del sistema FICLIT+CS@Unibo sia nella versione che utilizza sole somme di polarità (60.62%) sia in quella che usa il *parser* ed i *polarity shifter* (59.8%) classificatasi in settima posizione.

Conclusioni e sviluppi futuri

La *sentiment analysis* per la lingua inglese ha attratto l'interesse di molti ricercatori che, negli ultimi anni, hanno provato ad estendere il processo alla classificazione di testi anche in altre lingue: italiano, francese, tedesco, giapponese, cinese, arabo, etc... Ma, nonostante il crescente interesse per queste lingue, l'inferiore disponibilità di risorse online (es. testi, dizionari, lessici annotati) fa sì che l'inglese sia ancora la lingua più studiata nell'ambito dell'analisi del sentimento.

Le ricerche e le risorse *online* per la lingua italiana stanno aumentando fortemente negli ultimi anni e ciò è molto importante per ridurre il divario dall'inglese e, quanto finora presentato, vuole essere un altro piccolo passo verso questa direzione.

Questo lavoro di tesi ha presentato aspetti, tecniche e problematiche generali per la *sentiment analysis* con un focus particolare alla *sentiment polarity classification*.

Partendo da un'introduzione teorica sulle tecniche e le metodologie ad oggi in uso nel campo dell'analisi del sentimento, si è giunti alla presentazione di due sistemi per la per la *Twitter sentiment polarity classification* per la lingua italiana, basati su due approcci profondamente diversi.

Il primo approccio, realizzato in soli tre mesi, è stato presentato alla campagna di valutazione di EVALITA 2014 ed utilizza tecniche basate sul lessico e sulla polarità a priori di parole.

Il secondo, invece, utilizza tecniche di apprendimento supervisionato tra-

mite vettori di *feature* lessicali, morfosintattiche e dipendenti dal lessico realizzate per "catturare" le caratteristiche salienti dei *tweet*.

Per lo sviluppo dei sistemi si è utilizzato il linguaggio C++ ed alcuni annotatori sintattici opportunamente integrati nei sistemi di classificazione che, dato un qualsiasi *tweet*, restituiscono l'etichetta della classificazione: "1" se positivo, "-1"' se negativo, "0" se neutro.

Come in molti altri sistemi per la *Twitter sentiment polarity classification*, anche questi sistemi prevedono una fase di *pre-processing*, il *tweet* viene modificato rimuovendo e/o sostituendo tutto ciò che si è ritenuto superfluo per la classificazione (trasformazione delle *emoticon*, trasformazione indirizzi internet, etc...) Inoltre, per entrambi i sistemi si sono utilizzati alberi di dipendenze sintattici per analizzare le dipendenze fra le parole dei *tweet* e costruire un apposito algoritmo di propagazione della polarità (per l'approccio basato sul lessico) o una più corretta e accurata costruzione delle *feature* (per l'approccio basato sull'apprendimento automatico).

Terminata la realizzazione dei sistemi, infine, è stata misurata la loro "capacità di classificare" utilizzando il *training* ed il *test set* di EVALITA 2014 attraverso diverse metriche di misurazione dei risultati.

Nonostante i più che buoni risultati ottenuti, entrambi i sistemi presentati in questo lavoro di tesi possono essere notevolmente migliorati sia affinando e perfezionando gli strumenti utilizzati sia arricchendoli con altre tecniche e nuovi approcci.

Il lessico del sistema FICLIT+CS@Unibo può essere perfezionato ed esteso così come la propagazione della polarità lungo gli alberi delle dipendenze sintattiche può essere analizzata e realizzata con più precisione.

È possibile inserire un algoritmo di selezione delle *feature* nel sistema basato sull'approccio *machine learning* o utilizzare altri tipi di classificatori, oltre alle *Support Vector Machine*. Ulteriori miglioramenti dei risultati della classificazione si possono ottenere facendo precedere alla *sentiment polarity classification* un processo di *subjectivity classification* al fine di identificare

tweet che presentano solamente fatti da quelli che, invece, esprimono sentimenti ed opinioni. È possibile, inoltre, pensare di aggiungere alcune tecniche per la *irony detection* allo scopo di riconoscere e classificare con più precisione i *tweet* ironici o quelli utilizzano metafore.

Un esperimento molto interessante è la realizzazione di un sistema "misto" che effettua una classificazione di polarità dei *tweet* sfruttando e combinando i risultati sia dell'approccio "a regole" basato sul lessico sia di quello basato sull'apprendimento automatico stocastico/statistico.

Bibliografia

Baccianella S., Esuli A. and Sebastiani F. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC 2010*, pp. 2200-2204

Barbosa L., Feng J. 2010. Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36-44

Basile P., Novielli N. 2014. UNIBA at EVALITA 2014-SENTIPOLC Task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014*, pp. 58-63

Basile V., Nissim M. 2013. Sentiment Analysis on Italian Tweets. In *Proceedings of the 4th ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 100-107

Calzolari N., Magnini B., Soria C., Speranza M. 2009. La lingua italiana nell'era digitale. *META-NET Collana Libri bianchi*

Chang C., Lin C. 2011. LIBSVM: A Library for Support Vector Machines. In *Journal ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), No. 27

Cimino A., Cresci S., Dell'Orletta F., Tesconi M. 2014. Linguistically-motivated and Lexicon Features for Sentiment Analysis of Italian Twee-

ts. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014*, pp. 81-86

De Mauro T. 2000. Il dizionario della lingua italiana. *Paravia*

Deshmukh S.N., Shirbhate A. G. 2016. Feature Extraction for Sentiment Classification on Twitter Data. In *International Journal of Science and Research (IJSR)*, Volume 5, Issue 2

Dhande L. L., Patnaik G. K. 2014. Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier. In *Classifier, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, Volume 3, Issue 4, ISSN 2278-6856

Di Gennaro P., Rossi A., Tamburini F., 2014. The FICLIT+CS@UniBO System at the EVALITA 2014 Sentiment Polarity Classification Task. In *Proceedings of the Fourth International Workshop EVALITA 2014*, Pisa University Press, pp. 93-97, ISBN 978-886741-472-7

Esuli A. and Sebastiani F. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC 2006*, pp. 417-422

Fellbaum C. 1998. WordNet. An Electronic Lexical Database *The MIT Press*

forumTAL - Trattamento Automatico della Lingua. 2009. Libro Bianco sul Trattamento Automatico della Lingua. *Fondazione Ugo Bordoni* http://forumtal.fub.it/LB_TAL_ITA.pdf

Go A., Bhayani R., Huang L. 2009. Twitter Sentiment Classification using Distant Supervision. *Technical report*, Stanford

Hatzivassiloglou V., McKeown K. 2004. Predicting the semantic orientation of adjectives. In *Proceedings of the Joint ACL/EACL Conference*, pp. 174-181

Kamps J., Marx M., Mokken R.J., de Rijke M. 2004. Using WordNet to measure semantic orientation of adjectives. In *Language Resources and Evaluation (LREC)*

Kemp S. 2016. Digital in 2016. *We are social* http://wearesocial.com/uk/special-reports/digital-in-2016

Kim S., Hovy E. 2004. Determining the sentiment og opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*

Kumar A., Sebastian T.M. 2012. Sentiment Analysis: A Perspective on its Past, Present and Future. In *Intelligent Systems and Applications (IJISA)*, Vol. 4, No. 10, pp.1-14

Kumar G., Goel P., Chaulan S. 2012. Opinion mining and summarization for customer reviews. In *International Journal of Engineering Science and Technology (IJEST)*, Vol. 4, No. 8

Lesmo L. 2007. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4), pp. 46-47

Lewis M, Simons G., Fennig C. 2016. Ethnologue: Languages of the World, Nineteenthedition. *SIL International* - https://www.ethnologue.com/statistics/size

Liu B. 2012. Sentiment Analyses and Opinion Mining. In *Synthesis Lectures on Human Language Technologies*, pp. 1-167

Liu B., Hu M., Cheng J. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th international world wide web conference (WWW-2005)*, ACM Press: 10-14

Liu B., Zhang L. 2012. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pp. 415-463.

Pang B., Lee. L. 2008. Opinion mining and sentiment analysis. In *Foundations and Trends in Information Retrieval 2(1-2)*, pp. 1-135

Pang B., Lee L., Vaithyanathan S. 2008. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Volume 10, pp 79-86

Pianta E., Bentivogli L. and Girardi C. 2002. MultiWordNet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet, Mysore*, pp. 21-25

Rossi A. 2014. Costruzione di un lessico e applicazione di grammatiche locali per la sentiment analysis. *Tesi magistrale*

Russel S., Norving P. 2005. Intelligenza artificiale Un approccio moderno - Volume 2. *Prentice Hall - Pearson Education Italia*

Sharma R., Nigam S., Jain E. 2014. Mining Of Movie Reviews At Document Level. In *International Journal on Information Theory (IJIT)*, Vol. 3, No. 3

Taboada M., Brooke J., Tofiloski M., Voll K., Stede M. 2011. Lexicon-based methods for sentiment analysis. In *Association for Computational Linguistics*, vol. 37, pp. 267-307

Tang H., Tan S., Cheng X. 2009. A survery on sentiment detection of reviews. In *Expert System with Applications 36: An Internation Journal*, pp. 10760-10773

Tu H., Hatzivassiloglou V. 2003. Towards answering opinion questions: Separating facts fom opinions and identifying the polarity of opinion sentence. In *Proceedings of the Conference on Emirical Methods in Natural Language Processing (EMNLP)*

Turney P.D. 2005. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 417-424

Xianghua F., Guo L., Yanyan G., Zhiqiang W. 2013. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. In *Knowledge-Based Systems*, Volume 36, pp. 186-195

Zang C., Zang J., Xu Z. 2009. A Semantic Approach for Knowledge Sharing in Collaborative Commerce Environment. In *Wireless Communications, Networking and Mobile Computing, 2008. 4th International Conference on*

Zhao Y., Dong S., Li L. 2014. Sentiment Analysis on News Comments Based on Supervised Learning Method. In *International Journal of Multimedia and Ubiquitous Engineering*, Vol.9, No.7 pp. 333-346