UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

LISA: un tagger semantico *leggero* per l'italiano

Candidato: *Ludovica Pannitto*

Relatore: *Alessandro Lenci*

Correlatore: *Felice Dell'Orletta*

Anno Accademico 2015-2016

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Il presente lavoro ha come oggetto lo sviluppo di un tagger semantico per i nomi italiani. A partire da un'analisi del task e della letteratura scientifica prodotta sul tema (capitolo 3), è stata sviluppata un'ontologia presso il Laboratorio di Linguistica Computazionale dell'Università di Pisa[1], descritta nel capitolo 4. Il lavoro è stato poi svolto in due fasi: una fase di annotazione, descritta nel capitolo 5, per la creazione di una risorsa linguistica annotata al livello di classi semantiche dei nomi, e una fase di creazione di un modello di classificazione, denominato LISA - *Light Italian Semantic Analyzer*, descritta nel capitolo 6, tramite l'uso di tecniche di Machine Learning.

[1]CoLing Lab - Università di Pisa, `http://colinglab.humnet.unipi.it/`

Capitolo 2

Trattamento Automatico del Linguaggio

Con l'espressione Trattamento Automatico del Linguaggio - TAL (o Natural Language Processing - NLP) - intendiamo lo sviluppo di sistemi che siano in grado di fare uso di conoscenza linguistica al fine di comprendere o produrre testi [1] in linguaggio naturale. Si tratta quindi di sviluppare modelli formali di funzionamento del linguaggio naturale che approssimino una o più competenze del parlante nativo di una data lingua. Così come la nostra competenza linguistica di parlanti entra in gioco su più livelli di analisi di un dato linguistico[2], il trattamento automatico del linguaggio deve occuparsi di modellare la conoscenza linguistica in riferimento a tutti i livelli di analisi descrittiva delineati dalla teoria linguistica di riferimento.

[1] come in Lenci et al. 2005, p. 24, con testi intendiamo «qualsiasi prodotto dell'attività linguistica dei parlanti, elaborato o trascritto come sequenza di caratteri»

[2] in Lenci et al. 2005, p. 23, «prodotti del linguaggio che sono oggetto di un processo di analisi»

2.1 Analisi linguistica e pipeline di annotazione

Al fine di permettere che un sistema formale sia in grado di modellare la struttura interna del linguaggio risulta necessario che la conoscenza linguistica a cui parlanti accedono per la decodifica di un testo sia resa esplicita. Il processo di esplicitazione delle informazioni avviene attraverso l'annotazione linguistica, che consiste, appunto, nella codifica delle informazione associate al dato testuale permettendone il suo arricchimento attraverso l'acquisizione di metadati.

Il processo di annotazione avviene tipicamente in relazione ai tradizionali livelli di descrizione linguistica (morfologico, sintattico, semantico), ma, a seconda del tipo di informazione da identificare, i requisiti variano e con essi varia anche il livello di annotazione richiesta. Ci aspettiamo dal modello una competenza che approssimi la competenza di un parlante nativo, che è in grado di tollerare un certo grado di rumore di vario genere nella comunicazione: perché ciò accada è necessario che l'annotazione sia in grado di trattare input mal formati, produca risultati accurati, sia generalizzabile rispetto al dominio e alla lingua e sia incrementale, ovvero permetta una progressiva identificazione delle strutture linguistiche. In questo modo i livelli precedenti di analisi costituiscono l'input per i livelli successivi nel processo di estrazione di conoscenza.

L'analisi linguistica automatica viene così realizzata attraverso una sequenza di fasi di elaborazione su più livelli: a partire dal testo si procede alla divisione in periodi, alla tokenizzazione ed eventualmente alla normalizzazione del testo, alla lemmatizzazione e all'analisi morfosintattica e sintattica, sulla base della quale si può poi procedere ai livelli di analisi più complessi.

Il procedimento base può essere riassunto da questa pipeline di analisi:

Normalizzazione fase di pre-trattamento in cui si cerca di ridurre la variabilità del testo con l'applicazione di alcune procedure standard (ad esempio riduzione degli spazi multipli, aggiunta dello spazio dopo l'apostrofo, trasformazione degli apostrofi in accenti, normalizzazione di numeri, date etc.)

Sentence splitting fase di segmentazione del testo in frasi

Tokenizzazione fase di segmentazione di ciascuna frase in unità ortografiche

Lemmatizzazione e analisi morfologica riconduzione delle forme flesse a un insieme di possibili esponenti lessicali (o lemmi) e corrispondenti interpretazioni morfologiche

Part of Speech tagging fase di selezione della corretta interpretazione morfologica

Chunking fase di aggregazione dei token in chunk sintattici[3]

Analisi sintattica fase di identificazione delle relazioni sintattiche, a costituenti o a dipendenze[4].

[3]per una definizione di chunk sintattico, Abney 1991

[4]nel modello a costituenti, la rappresentazione è basata sull'identificazione di costituenti sintattici e di relazioni di incassamento gerarchico: vengono marcate relazioni *parte-tutto* e il ruolo sintattico svolto da un costituente è completamente determinato dalla sua posizione nell'albero gerachico. Nel modello a dipendenze, invece, ad essere marcate sono relazioni di tipo *parte-parte*. La rappresentazione fornisce una descrizione della frase in termini di relazioni binarie di dipendenza tra token.

Tabella 2.1: Sentence Splitting

ID	Sentence
1	Il danno non poteva essere sottovalutato.
2	Il sig. Rossi decise perciò di chiamare l'avvocato.

Analisi semantica fase di identificazione nel testo di istanze di classi semantiche definite in un'ontologia[5]

Esempio

Consideriamo il testo[6]:

> *Il danno non poteva essere sottovalutato. Il sig. Rossi decise perciò di chiamare l'avvocato.*

Il primo passaggio consiste nell'individuare di quali frasi si compone il testo (come nella Tabella 2.1), successivamente le frasi vengono tokenizzate e ogni token viene analizzato morfologicamente (nella Tabella 2.2 viene mostrato il processo sulla frase 2). Infine viene selezionata l'informazione morfosintattica corretta, e a partire da questa l'analisi prosegue con la fase di chunking e la fase di annotazione sintattica (come mostrato nella tabella 2.3)

[5]per una definizione di ontologia, capitolo 3.1

[6]L'analisi qui mostrata è stata effettuata con la pipeline di annotazione LinguA, si veda DellOrletta 2009, Attardi e Dell'Orletta 2009, Attardi et al. 2009

Tabella 2.2: Tokenizzazione e Analisi Morfologica

id	**form**	**lemma**	**pos**	**feats**
1	Il	Il	RD	MS
2	danno	danno,dare	S;V	MS;P31P
3	non	non	B	NULL
4	poteva	potere	V	S3II
5	essere	essere	V	F
6	sottovalutato	sottovalutare	V	MSPR

Tabella 2.3: PoS Tagging, Chunking e Analisi Sintattica

id	**form**	**lemma**	**pos**	**feats**	**b/i**	**chunk type**	**chunk role**	**head**	**dep**
1	Il	Il	RD	MS	B	N_C	DET	2	DET
2	danno	danno	S	MS	I	N_C	POTGOV	6	SUBJ_PASS
3	non	non	B	NULL	B	BE_C	PREMODIF	6	NEG
4	poteva	potere	V	S3II	I	BE_C	MOD	6	MODAL
5	essere	essere	V	F	I	BE_C	AUX	6	AUX
6	sottovalutato	sottovalutare	V	MSPR	I	BE_C	POTGOV	0	ROOT

2.2 Analisi semantica, polimorfismo e ambiguità semantico lessicale

L'analisi semantica, di cui questo lavoro tratta, consiste nel mettere in relazione il piano del significato con la struttura sintattica o morfosintattica di porzioni di testo. Si tratta, quindi, di un processo che si interfaccia con l'ambito della rappresentazione della conoscenza espressa da enunciati in linguaggio naturale.

Task quali risoluzione delle anafore, individuazione dello scope dei quantificatori, word sense disambiguation o role labeling possono risultare complessi anche se sottoposti ad un parlante nativo, proprio perché richiedono spesso l'accesso, da parte del parlante, a conoscenze che travalicano la struttura inerente al testo e chiamano in gioco informazioni sul contesto comunicativo.

Ulteriore problema risulta la presenza di ambiguità che, a livello semantico, si realizza in senso stretto in vari fenomeni, che possiamo suddividere in fenomeni di omonimia[7] e di polisemia[8].

Per quanto attiene poi al fenomeno polisemico è rilevabile che non tutti i fenomeni

[7]l'omonimia è il caso in cui due lessemi diversi si trovano a condividere la forma ortografica, ad esempio nel caso di *cane* - mammifero e *cane* - componente di una pistola. In questo caso i due sensi ascrivibili alla stessa unità ortografica non sono semanticamente collegati

[8]la polisemia è il caso in cui i sensi sono in qualche modo collegati. Consideriamo ad esempio il verbo *piantare* nelle due frasi:

(1) a. Gianni ha *piantato* sua moglie

b. Gianni ha *piantato* i fiori in giardino

hanno la stessa regolarità: alcuni tipi di polisemia appaiono infatti più regolari di altri, come ad esempio il passaggio da contenente a contenuto nei casi (esempi tratti da Murphy 2010, p. 90):

(1) a. riempire la scatola/bottiglia/valigia

b. rovesciare la scatola/bottiglia/valigia sul pavimento

Il significato di una parola è inoltre influenzato dalla sua collocazione all'interno di una MultiWord Expression (MWE) o di una struttura idiomatica, le cui definizioni si trovano in realtà collocate lungo un continuum che prende in esame le preferenze combinatorie delle parole. Mentre infatti alcune potenzialità combinatorie sono determinate da tratti morfosintattici e semantici generali delle parole stesse, in virtù della loro appartenenza a classi astratte (ad esempio considerando la classe dei nomi e quella degli aggettivi, sarà naturale trovare nomi concreti legati ad aggettivi che indicano qualità sensibili come colore o forma, piuttosto che ad aggettivi che indicano qualità morali), altre combinazioni lessicali si basano su legami non prevedibili[9]. In altre parole, analizzando i rapporti che le parole instaurano sull'asse sintagmatico, alcune espressioni polirematiche risultano fortemente lessicalizzate (MultiWord Expression), idiomatiche o idiosincratiche (strutture idiomatiche), e si trovano in questo modo a perdere la proprietà di composizionalità acquisendo rigidità strutturale nei confronti delle modificazioni a cui le parole sono normalmente sottoposte, ad esempio per mezzo di aggettivi e avverbi o nel passaggio dalla forma attiva alla forma passiva. Si confrontino i casi:

(2) a. Il re ha tirato le cuoia.

[9]pensiamo a *notte fonda* o *luna di miele*

b. * Le cuoia sono state tirate dal re.

(3) a. Luca ha tirato la palla.

b. La palla è stata tirata da Luca.

La capacità delle parole di associarsi in collocazioni più forti di altre è inoltre una proprietà dipendente dalla varietà di lingua che stiamo considerando.

2.3 Strumenti per l'analisi del linguaggio

Le tecnologie sviluppate nel tempo in ambito linguistico computazionale nascono all'incrocio tra metodi statistico-matematici utilizzati nel campo dello studio dei testi letterari e il mondo dell'Intelligenza Artificiale. Dagli anni '80 la prevalenza del paradigma empirista rispetto a quello formalista ha portato in primo piano i modelli statistici e il Machine Learning, definitivamente suggellati dalla crescente quantità di dati di cui negli ultimi anni disponiamo.

2.3.1 Machine Learning

L'apprendimento automatico (o Machine Learning) è una branca dell'Intelligenza Artificiale che si occupa della realizzazione di algoritmi che, dall'osservazione dei dati, sono in grado di sviluppare conoscenza e utilizzarla nell'analisi di nuovi dati. Seguendo la definizione di Mitchell[10]:

[10]Mitchell 1997, p. 2

> *un programma apprende da una certa esperienza E se: nel rispetto di una classe di compiti T, con una misura di prestazione P, la prestazione P misurata nello svolgere il compito T è migliorata dall'esperienza E*

La filosofia di sistemi automatici di questo tipo consiste dunque nell'inferire, da una serie di esempi (che costituiscono l'«esperienza E»), un modello statistico dei dati linguistici di cui ci stiamo occupando, che sia in grado di dare risposte soddisfacenti, in termini di misure statistiche di accuratezza[11], nel caso di nuovi dati.

Task supervised e unsupervised

Per gli algoritmi di apprendimento automatico i task si dividono essenzialmente in due tipologie, entrambe rilevanti nel campo della linguistica computazionale: task di apprendimento supervisionato e task di apprendimento non supervisionato.

Un task supervisionato consiste nel trovare un'approssimazione $h(\cdot)$ di una funzione $f(\cdot)$ non nota, che associa i dati di input ai rispettivi target. $h(\cdot)$ è costruita basandosi su un insieme di dati $\langle \vec{x}, f(\vec{x}) \rangle$ per i quali si conosce il valore della funzione $f(\cdot)$. $\vec{x}$ è un vettore di n feature che rappresentano un'istanza del problema. Partendo quindi da una serie di m casi di esempio $\{\langle \vec{x}_1, f(\vec{x}_1) \rangle, ..., \langle \vec{x}_m, f(\vec{x}_m) \rangle\}$ dove $f(\cdot)$ è la funzione target, l'algoritmo di machine learning supervisionato crea un'approssimazione $h(\cdot)$ di $f(\cdot)$ tale che, dato un nuovo input x^* mai visto prima, restituisce $h(x*)$ che è il valore predetta per x^*. Tipici casi di algoritmi supervisionati sono quelli che si occupano della *classificazione* di istanze o di problemi di *regressione*.

[11] un breve riepilogo delle misure di accuratezza rilevanti per questa trattazione è presentato nel capitolo 6.1.2

Nel primo caso, dato un insieme C di classi predefinite, $f : X \to C$ associa ad ogni istanza $\vec{x} \in X$ la sua classe $c \in C$. Nel secondo caso la funzione $f : X \to Y$ associa ad ogni istanza $\vec{x} \in X$ un valore in un certo dominio e il problema consiste nel trovare la relazione funzionale che lega ogni coppia (x_i, y_i)

Per quanto riguarda invece gli algoritmi non supervisionati, non esiste alcuna funzione $f(\cdot)$ da approssimare, ma lo scopo è quello di mostrare proprietà intrinseche nei dati. I task non supervisionati possono essere usati ad esempio per trovare raggruppamenti naturali degli input, per ridurne la dimensionalità (ovvero per renderne possibile una rappresentazione con un numero minore di feature) o per trovare elementi anomali nei dati di ingresso. Tra gli algoritmi non supervisionati citiamo invece il caso del *clustering*, che consiste nel raggruppare elementi seguendo relazioni di similitudine tra i dati.

Capitolo 3

SuperSense Tagging

3.1 Ontologie semantico lessicali

Una ontologia è una rappresentazione formale, condivisa ed esplicita della concettualizzazione di un dominio di interesse. Da sempre le ontologie sono impiegate nel campo dell'Intelligenza Artificiale (e, di conseguenza, della linguistica computazionale), perché, stabilendo relazioni tra le entità in gioco, risultano particolarmente adatte al ragionamento induttivo e alla classificazione, grazie al principio di ereditarietà e alla loro frequente struttura gerarchica. Un particolare tipo di ontologia è costituito dalle Reti Semantiche[1]: formalmente un grafo[2] in cui i nodi sono etichettati con i concetti e gli archi con relazioni semantiche tra i concetti. Il significato di un concetto viene in questo modo a essere determinato dalle relazioni che stabilisce

[1] introdotte in Ross 1968, Woods 1975

[2] una coppia $G = (V, E)$ con V insieme dei nodi, $E \subseteq V \times V$ insieme di archi

con altri concetti nella rete, e le sue proprietà semantiche si possono inferire dalla struttura formale della rappresentazione.

3.2 WordNet

Un esempio di rete semantica rilevante per la nostra trattazione è WordNet[3], un database semantico-lessicale basato sul modello relazionale, sviluppato per la lingua inglese a Princeton a partire dagli anni '80. E' composto da quattro reti semantiche principali, una per ogni classe aperta presente nella risorsa (Nomi, Verbi, Aggettivi, Avverbi). Le word forms sono organizzate secondo la relazione di sinonimia[4]: più parole che condividono alcuni contesti vengono considerate sinonimi e raggruppate in un synset, che individua in questo modo un concetto lessicalizzato.

Le altre relazioni semantiche che sono definite sulla rete non sussistono dunque tra parole ma tra concetti (o sensi di parole). In questo modo possiamo ridefinire omonimia e polisemia in termini di relazioni instaurate nella rete: *cane* animale domestico e *cane* parte della pistola sono due omonimi, in quanto la stessa forma lessicale si trova in due synset che tra loro non instaurano alcun tipo di relazione. Su Wordnet i due synset hanno rispettivamente id 024071-n (*mammifero domestico, dei canidi, molto comune, diffuso in tutto il mondo, con attitudini varie a seconda della razza*) e 03481824-n (*nelle armi da fuoco, barretta d'acciaio cilindrica che percuote il detonatore provocando l'accensione della carica di lancio e l'escursione del proiettile*).

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

[4] ricalcando la definizione Leibniziana, in Fellbaum 1998 viene proposta questa definizione di sinonimia: "*two expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value*"

La relazione che sussiste invece tra *chiave* - oggetto concreto - e *chiave* - soluzione - è una relazione di polisemia. I due synset hanno rispettivamente id 03613294-n (*strumento metallico atto ad aprire e a chiudere una serratura*) e 05794057-n (*tutto ciò che serve a capire e a svelare*).

L'accesso a WordNet può avvenire sia per lemma, avendo così in risposta l'insieme dei synset a cui tale lemma appartiene, che per concetto. In altre parole, WordNet adempie a una duplice funzione: di dizionario e di thesaurus.

Per le varie categorie lessicali sono definiti diversi tipi di relazione. La rete nominale è quella che evidenzia di più l'aspetto gerarchico dell'organizzazione dei concetti, in quanto è costruita principalmente attraverso la relazione di iperonimia, così definita:

(4) Y è iperonimo di X se è vera la proposizione *X è un tipo di Y*

Emerge in questo modo un diagramma ad albero. Come evidenziato in Miller 1998, sarebbe teoricamente possibile raccogliere l'intero albero sotto un'unica radice, semanticamente un synset vuoto $\{\Lambda\}$, ma la giustificazione lessicale a questo espediente sarebbe tenue a causa dell'impossibilità di instaurare relazioni semantiche informative tra concetti troppo astratti.

La gerarchia dei nomi in WordNet è dunque organizzata in varie sottogerarchie, con differenti *unique beginners*, il termine utilizzato per indicare il più alto e più inclusivo grado di una tassonomia, che si riferiscono a campi semantici diversi. L'insieme di *unique beginners* è stato definito in 25 classi lessicografiche (riportate nella tabella 3.1, nella quale sono riportate anche ulteriori relazioni che riducono le 25 categorie originali, in neretto, a 11 classi).

Tabella 3.1: Rappresentazione delle relazioni che riducono le 25 classi lessicografiche a 11 *unique beginners*. Le classi originali sono riportate in grassetto, gli *unique beginners* sono da identificati dal pedice «u»

entity$_u$	organism	**animal**	
		person	
		plant	
	object	**artifact**	
		natural object	**body**
		substance	**food**
	abstraction$_u$	**attribute**	
		quantity	
		relation	**communication**
		time	
	psychological feature$_u$	**cognition**	
		feeling	
		motivation	
		natural phenomenon$_u$	**process**
		activity$_u$	
		event$_u$	
		group$_u$	
		location$_u$	
		possession$_u$	
		shape$_u$	
		state$_u$	

3.3 SuperSense Tagging

NER e WSD

Riportiamo qui la definizione di due task semantici rilevanti per la trattazione successiva:

Il task di **Word Sense Disambiguation** (WSD) consiste nello scegliere il senso giusto in un contesto per una parola ambigua. La difficoltà del task è stata equiparata in letteratura alla risoluzione di problemi centrali dell'Intelligenza Artificiale, quali ad esempio il test di Turing[5] Come evidenziato in Navigli 2009, le difficoltà di approccio al task sono molteplici: sono fattori influenti la granularità del tagset, l'approccio alla rappresentazione di un word-sense, il dominio di interesse in quanto la difficoltà della disambiguazione nel passaggio da un dominio ristretto a un dominio generalista aumenta sensibilmente, e la forte dipendenza da fonti di conoscenza, senza le quali il task diventa difficile anche per un umano.

Formalmente, considerata una sequenza $(w_1, ..., w_n)$ di token, il task consisite nell'identificare il mapping $A : A(i) \subseteq Senses_D(w_i)$, dove $Senses_D(w_i)$ è l'insieme finito o numerabile dei sensi del token w_i in un dato dizionario D e $A(i)$ è l'insieme dei sensi di w_i appropriati nel contesto, tipicamente $|A(i)| = 1$.

Il task di word-sense disambiguation è tipicamente trattato come un task di classificazione in cui i word-sense sono le classi e un metodo automatico di classificazione viene utilizzato per assegnare ad ogni token una o più classi a seconda del contesto

[5]criterio elaborato in Turing 1950 per stabilire se una macchina sia in grado di dimostrare un'intelligenza tale da essere indistinguibile dall'intelligenza umana.

e delle fonti di conoscenza esterne.

Il task di **Named Entity Recognition** (NER) consiste invece nell'identificazione di una parola o sequenza di parole in un testo, che rappresentino una particolare entità di interesse rispetto al dominio di riferimento.

Nasce nell'ambito dell'Information Extraction e richiede l'identificazione di nomi propri nel testo e la loro classificazione secondo un set di categorie di interesse, ad esempio persona, entità geo-politica, entità temporali o monetarie, ma a seconda del dominio possiamo incontrare nomi di farmaci, riferimenti bibliografici....

Il modello ontologico di riferimento, per natura del task, risulta spesso linguisticamente troppo scarno, lasciando sottospecificate categorie potenzialmente informative. Inoltre il task è spesso ristretto al riconoscimento e alla classificazione di sequenze di nomi propri.

3.3.1 SST

Il SuperSense Tagging (SST) consiste nell'annotare ogni entità in un contesto con la categoria giusta in riferimento ad una certa tassonomia. Può essere, quindi, considerato a metà strada tra un task di Word Sense Disambiguation (WSD) e un task di Named Entity Recognition (NER). Rispetto al primo, le categorie sono meno specifiche, ma estendono comunque quelle utilizzate per il NER. La tassonomia inoltre non è ristretta ad entità di dominio ma classifica anche nomi comuni.

Sebbene la definizione delle categorie sia in generale più chiara della definizione di *word-sense*, fenomeni linguistici come la *metonimia* introducono notevoli casi di

ambiguità. A questo proposito si confrontino ad esempio:

(5) a. Il *tavolo tre* chiede il conto

b. Il cameriere fa accomodare i clienti al *tavolo tre*

(6) a. Quest'estate andrò in vacanza in *Italia*

b. L'*Italia* ha perso i mondiali

L'ontologia di riferimento più largamente utilizzata, e introdotta in Ciaramita e Johnson 2003, si basa sulle classi lessicografiche di WordNet (26 per i nomi, 15 per i verbi, 3 per gli aggettivi e 1 per gli avverbi). Il set di categorie risultanti è piuttosto ridotto, e ciò permette di generalizzare rispetto al livello dei synset[6], redendo il task trattabile con metodi di machine learning di cui disponiamo allo stato attuale. Inoltre, le classi sono facilmente riconoscibili senza risultare troppo astratte.

3.4 Stato dell'arte

Il task viene per la prima volta proposto in Ciaramita e Johnson 2003, come estensione della Named Entity Classification. Viene qui utilizzato un set di 26 etichette semantiche, derivato dalle classi lessicografiche di WordNet (riportate in Tabella 3.1), che gli autori definiscono *supersensi*.

L'algoritmo utilizzato è un Multiclass Averaged Perceptron[7] con feature standard usate in task di word-sense classification e named-entity recognition (ad esempio collocazioni, spelling, feature contestuali...).

[6] i synset dei sostantivi sono oltre 75000 in WordNet 1.71

[7] come definito in Crammer e Singer 2003

Tabella 3.2: Composizione Training Data Ciaramita e Johnson 2003

Corpus	**n. istanze**
Bliip	787186
WordNet's definitions	66841
Example Sentences	6147
Tot	**860174**

Viene usato come training set l'insieme dei sostantivi etichettati in WordNet 1.6 e il modello viene valutato sull'insieme dei nuovi sostantivi inseriti in WordNet 1.71 . Il training set (si veda la Tabella 3.2) è costruito combinando istanze provenienti dal corpus Bllip[8] da cui sono state rimosse istanze ambigue, istanze provenienti dalle frasi di esempio e dalle glosse presenti in WordNet.

Sono stati creati due test set per la valutazione, il primo a partire dal corpus Bliip, questa volta considerando le istanze non ambigue presenti in WordNet 1.71, il secondo utilizzando 20394 token del training set.

Sono stati dunque creati tre modelli e testati su entrambi i test set, utilizzando per il train il 55% delle istanze provenienti da Bllip (AP-B-55), il 65% delle istanze provenienti da bliip (AP-B-65), e un insieme identico al primo con l'aggiunta dei dati provenienti da WordNet (AP-B-55+WN). Come riportato dagli autori, i risultati, riportati nella Tabella 3.3, mostrano che l'aggiunta delle istanze provenienti dalle glosse e dagli esempi di WordNet permette un sensibile miglioramento nelle performance, e che ciò non dipende solo dalla maggior quantità di dati disponibili, ma anche dalla miglior qualità dei dati e dalla presenza di istanze disambiguate di

[8]Charniak et al. 2000

Tabella 3.3: Risultati Ciaramita e Johnson 2003

	WordNet 1.6		WordNet 1.71	
Modello	**token**	**type**	**token**	**type**
Baseline	24.1%	21.0%	20.0%	27.8%
AP-B-55	47.4%	47.7%	35.9%	50.7%
AP-B-65	47.9%	48.3%	36.1%	50.8%
AP-B-55+WN	36.9%	52.9%	52.3%	53.4%

lemmi polisemici, che erano invece stati eliminati dal corpus Bllip.

Restando nell'ambito degli algoritmi supervisionati, in Ciaramita e Altun 2006 lo stesso task viene affrontato come un problema di labeling sequenziale[9], utilizzando un Hidden Markov Model (HMM), come descritto in Collins 2002. Il tagging viene qui esteso ai verbi, sempre utilizzando le classi lessicografiche di WordNet.

Sono stati utilizzati tre dataset distinti: due di questi (SEM e SEMv) provenienti da Semcor [10], un sottoinsieme del Brown corpus[11] manualmente annotato con gli id dei synset di WordNet aggiornati alla versione 2.0 . In SEM sono annotati nomi, verbi e aggettivi, mentre in SEMv solo i verbi. Il terzo dataset proviene da quello fornito per un task di Senseval-3[12] (SE3), che si proponeva di valutare la performance di disambiguazione su tutte le classi aperte. Si compone di porzioni del Wall Street Journal e del Brown Corpus. Il labeling è stato portato nel formato IOB, arrivando

[9]un tipico task di labeling sequenziale è ad esempio il PoS Tagging: in questi casi si cerca di ottimizzare il tagging di una sequenza di parole, piuttosto che di una singola parola

[10]Miller et al. 1993

[11]Kuera, Francis et al. 1967

[12]Snyder e Palmer 2004

Tabella 3.4: Risultati Ciaramita e Altun 2006

	Semcor			Senseval - 3		
Modello	**Recall**	**Precision**	**F-measure** [σ]	**Recall**	**Precision**	**F-measure** [σ]
Baseline	69.25%	63.9%	66.47%	68.65%	60.10%	64.09%
SuperSense Tagger	77.71%	76.65%	77.18% [0.45]	73.74%	67.6%	70.54% [0.21]

così a un numero totale di 83 etichette. I risultati riportati in Tabella 3.4, ottenuti tramite 5-fold cross validation, sono sensibilmente migliori rispetto a quelli ottenuti da algoritmi paragonabili nel task di Senseval-3, e incoraggianti rispetto a risultati nel capo del NER.

Il task è stato affrontato anche con approcci non supervisionati, riportiamo qui lo studio presentato in Curran 2005. I dati di training sono costituiti da un corpus di 2 miliardi di parole, parsati superficialmente a livello sintattico in costituenti non ricorsivi[13], composto come riportato nella Tabella 3.5: su questo è stata operata una normalizzazione per ridurre il rumore, escludendo frasi troppo corte (meno di 3 token) o troppo lunghe (più di 100), e che includessero più di 5 numeri o 4 parentesi.

Per la classificazione viene utilizzata la similarità semantica in uno spazio vettoriale, basandosi sull'ipotesi distribuzionale[14], secondo la quale parole simili appaiono in contesti simili: estraendo automaticamente dei sinonimi per il target, si assegna a questo il supersenso più probabile tra quelli dei sinonimi. Nel caso non sia possibile reperire dei sinonimi a causa della bassa frequenza del token, gli autori adoperano un classificatore a regole che si basa su informazione morfologica considerando *artifact* la classe di default. Il sistema è stato testato sullo stesso test set utilizzato

[13]per una definizione più completa di *shallow-parsing* si veda Martin e Jurafsky 2000, p. 577
[14]Harris 1970

Tabella 3.5: Composizione Training Data Curran 2005

Corpus	**Token**
British National Corpus	114M
Reuters Corpus Volume I	207M
Continuous Speech Recognition III	226M
North American News Text Corpus	559M
North American News Text Corpus Supplement	507M
ACQUAINT Corpus	491M

Tabella 3.6: Risultati Curran 2005

Modello	**WN 1.6**	**WN 1.71**
Ciaramta & Johnson baseline	21%	28%
Ciaramita & Johnson perceptron	53%	53%
Semantic based results	68%	63%

in Ciaramita e Johnson 2003. I risultati, riportati nella Tabella 3.6, mostrano un miglioramento rispetto a quanto fatto in Ciaramita e Johnson 2003.

Il task è stato inizialmente ripreso per la lingua italiana in Picca et al. 2008, come adattamento del SuperSense Tagger presentato in Ciaramita e Altun 2006. Il modello è stato allenato su MultiSemCor [15], un corpus di 116 documenti tradotti manualmente in italiano dai corrispettivi inglesi presenti in SemCor. Il corpus è stato allineato automaticamente a quello inglese, e in questo modo sono stati inferiti

[15]Bentivogli et al. 2004

Tabella 3.7: Risultati Picca et al. 2008

Recall	**Precision**	**F-measure**
0.6357	0.6225	0.6290

i supersensi, estratti da MultiWordNet[16]. I risultati, riportati in Tabella 3.7, sono stati valutati con una cross-validation sullo stesso corpus adottato per il training e risultano peggiori del corrispondente sistema per la lingua inglese: ciò può essere causato dalla peggior qualità delle risorse[17], la minor quantità di risorse a disposizione[18] e dall'esclusione dell'euristica del primo senso dalle feature[19].

Il task è stato poi proposto a EVALITA 2011[20]. La competizione riguardava il tagging di tutte le classi aperte (nomi, aggettivi, verbi e avverbi) con riferimento alle classi lessicografiche di WordNet, 45 in totale, descritte nell'appendice A. Il task si componeva di due subtask: il primo (*closed subtask*) mirava alla valutazione di sistemi creati a partire dal solo corpus fornito come training, il secondo (*open subtask*) permetteva l'uso di risorse esterne.

[16] `http://multiwordnet.fbk.eu`

[17] gli autori notano che circa il 14% dei trasferimenti di senso è errato

[18] rispetto alla versione inglese, le parole etichettate a disposizione degli autori sono meno della metà, 92000 rispetto alle oltre 200000 inglesi

[19] l'ordinamento dei sensi nella WordNet italiana, a differenza di quanto accade nella versione utilizzata per la lingua inglese, non riflette la frequenza dei sensi

[20] EVALITA, `http://www.evalita.it/`, è una campagna di valutazione di strumenti di Trattamento Automatico del Linguaggio e vocali per l'italiano. Promossa a partire dal 2007 dall'Associazione Italiana di Linguistica Computazionale - AILC - congiuntamente con l'Associazione Italiana per l'Intelligenza Artificiale - AI*IA - e l'Associazione Italiana Scienze della Voce - AISV -, propone un framework attraverso il quale diversi sistemi e approcci possano essere correttamente confrontati.

Tabella 3.8: Descrizione campi token Evalita 2011

Nome campo	**Descrizione**
Form	Word form or punctuation symbol
Lemma	Word lemma or punctuation symbol
PoS	Part-of-Speech tag, with morphological features, based on the TANL tagset
SuperSense Tag	SuperSense Tag in IOB notation

Il dataset fornito è stato creato a partire dalla Italian Syntactic Semantic Treebank[21], attraverso un processo di correzione automatica e revisione manuale. Il corpus risultante, ISST-SST, parte del progetto SemaWiki (Text Analytics and Natural Language processing - TANL) [22], è composto da circa 300000 token, annotati fino al livello di analisi morfologica (ogni token consiste di 4 campi, come descritti in Tabella 3.8). Di questi, circa 276000 riservati alla fase di development e training, mentre i dati di test sono composti da circa 30000 token provenienti da ISST-SST e altri 20000 da una porzione della Wikipedia italiana.

Alla competizione hanno partecipato due team, uno dell'Università di Pisa e uno dell'Università di Bari. Il modello dell'Università di Pisa, descritto in Attardi et al. 2013, si basa su un Maximum Entropy Classifier per il chunking e un algoritmo di programmazione dinamica per il tagging. Il modello è derivato da precedenti esperimenti presentati in Attardi et al. 2010, nel quale si miglioravano i risultati mostrati in Picca et al. 2008, ampliando e migliorando la qualità della risorsa di partenza. Il modello dell'Università di Bari, descritto in Basile 2013, utilizza invece

[21] Montemagni et al. 2003

[22] `http://medialab.di.unipi.it/wiki/SemaWiki`, portato avanti in collaborazione tra l'Università di Pisa e l'ILC-CNR, Attardi et al. 2008

Tabella 3.9: Risultati Evalita 2011

	Accuracy	**Precision**	**Recall**	**F1-measure**	**F1 SST**	**F1 Wikipedia**
UniPi	88.50%	76.82%	79.76%	78.27%	78.23%	78.36%
UniBa	86.96%	74.85%	75.83%	75.34%	76.29%	73.38%

una Support Vector Machine. Si riportano nella Tabella 3.9 i risultati ottenuti dai due team nel closed subtask della competizione.

Un dato interessante si può trarre dal confronto dei risultati dei modelli sviluppati sulla lingua italiana sulle singole classi di sostantivi: le classi identificate in Picca et al. 2008 come le più semplici da identificare, *noun.body*, *noun.person* e *noun.time*, si confermano tali nei risultati presentati a EVALITA 2011 (i risultati sono riportati in Tabella 3.10). Le classe indicate come più difficili da individuare erano invece *noun.event*, *noun.feeling* e *noun.shape*. In Attardi et al. 2013 vengono invece indicate *noun.animal*, *noun.plant* e *noun.food*: confrontandolo con il miglioramento del team dell'Università di Bari nell'*open subtask* (come riportato in Basile 2013), si può argomentare, come fatto dagli autori stessi, che ciò, considerata la composizione del test set, indichi la rilevanza delle risorse esterne.

3.5 Light Semantic Tagging

Il nostro task prende spunto dall'*open subtask* proposto a EVALITA 2011, con alcune sostanziali differenze:

- Ci restringiamo alla sola categoria dei sostantivi

Tabella 3.10: Sono riportati i valori di f1-measure ottenuti in Picca et al. 2008, Attardi et al. 2013 e Basile 2013 per ogni classe semantica. In grassetto sono evidenziate, per ogni modello, le classi che hanno ottenuto migliori e peggiori risultati.

	Picca et al. 2008	**Attardi et al. 2013**	**Basile 2013**
noun.person	**0,7876**	61,73	
noun.time	**0,7684**	**83,61**	
noun.body	**0,7099**	**85,25**	
noun.artifact	0,6600	63,68	63.79
noun.phenomenon	0,6479	82,61	
noun.possession	0,6451	75,88	78.29
noun.motive	0,6405	72,41	
noun.animal	0,6242	**50,00**	
noun.group	0,6090	59,46	66.11
noun.substance	0,6019	57,14	
noun.plant	0,5897	**37,84**	
noun.object	0,5884	64,46	
noun.food	0,5858	**28,57**	
noun.quantity	0,5849	81,96	
noun.process	0,5806	76,19	
noun.cognition	0,5726	75,44	
noun.communication	0,5724	72,03	74.17
noun.act	0,5713	**85,37**	**83.99**
noun.state	0,5637	80,34	
noun.attribute	0,5600	82,09	
noun.location	0,5488	65,70	
noun.relation	0,4357	67,25	
noun.feeling	0,4178	78,79	
noun.event	**0,4087**	79,59	
noun.shape	**0,3699**	66,67	

- Il corpus è stato annotato automaticamente, dunque senza revisione manuale a differenza della risorsa fornita per EVALITA 2011, fino al livello di parsing sintattico a dipendenze, aggiungendo dunque un livello di analisi
- L'ontologia di riferimento è diversa: il task di SuperSense Tagging si è consolidato come task di classificazione rispetto ai supersensi derivati da WordNet. Nel nostro task le categorie semantiche sono state sviluppate a partire dalla Light Semantic Ontology (si veda capitolo 4.1)

Il task è stato affrontato come un task supervisionato di classificazione multiclasse, per la costruzione del modello è stata utilizzata una Support Vector Machine con kernel lineare (si veda capitolo 6.1).

Capitolo 4

Light Semantic Ontology

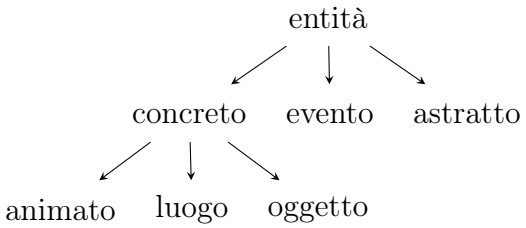
Si descrive l'ontologia *Light Semantic Ontology* e il tagset da essa derivato.

4.1 Light Semantic Ontology

I sistemi fin qui citati, come già detto, hanno utilizzato le classi lessicografiche di WordNet come supersensi. La necessità di una classificazione minimale e linguisticamente plausibile, insieme con la constatazione che molti dei supersensi derivati dalle classi lessicografiche di WordNet sono problematici, ad esempio la distinzione nel dominio degli astratti o degli eventi, ha portato ad elaborare, presso il Laboratorio di Linguistca Computazionale dell'Università di Pisa, la *Light Semantic Ontology*, che si basa su una principale divisione delle entità in tre tipi: entità concrete, astratte ed eventi[1], come mostrato in Figura 4.1. L'ontologia riguarda esclusivamente i

[1]lo scheletro principale è mutuato dalla classificazione proposta da Lyons 1977

Figura 4.1: Light Semantic Onthology



sostantivi e la classificazione non è da intendersi assoluta del tipo di entità del mondo reale, ma relativa rispetto all'argomento linguistico realizzato dall'entità[2] in un determinato contesto.

Entità concrete

Rientrano in questa categoria i concetti che denotano entità percepibili con i sensi e localizzate nello spazio e nel tempo. Le entità concrete si suddividono a loro volta in:

- Animati: se dotati di un qualche ruolo agentivo.
- Luoghi: se, nel contesto linguistico, descrivono entità in cui ha luogo un evento.
- Oggetti: se una entità concreta, ma non rientra nelle due precedenti categorie.

[2]ad esempio il lemma *vita* è classificato come *abstract* qualora esso sia trattato come un'entità per la quale si lotta o un valore assoluto, ma è classificato come *event* qualora si riferisca allo stato di un essere o all'insieme delle attività che ha compiuto una persona.

Eventi

Le entità di tipo evento sono denotate da concetti che esprimono stati, proprietà o processi. Sono entità che accadono nel tempo.

Entità astratte

La categoria delle entità astratte raccoglie i concetti che non rientrano nelle categorie presentate precedentemente, svolgendo il ruolo di classe residuale, seguendo l'approccio di Lyons 1977. Sono generalmente entità coinvolte in una comunicazione, elementi cognitivi, domini o misure.

4.2 Il tagset

A partire dalla *Light Semantic Onthology* è stato sviluppato un tagset, utilizzato poi per annotazione del corpus, così composto:

O(ther) sostantivi privi di contenuto semantico perché utilizzati in costruzioni funzionali, ad esempio *da* ***parte*** *di*, *in* ***modo*** *che* o lo stesso *ad* ***esempio***. Allo stesso modo sono stati trattati casi in cui il filler è stato erroneamente identificato come sostantivo nella fase di PoS Tagging.

Animate sostantivi utilizzati in contesti agentivi, ad esempio in grado di *volere*, *sbagliare*, *morire*...

Location sostantivi che esprimono collocazioni spaziali relativamente fisse e indicano qualcosa in cui ha luogo un evento. Possono essere sostituiti da *qui*, *un altro posto*.

Object sostantivi che indicano oggetti concreti (ad esempio artefatti) o sostanze, in generale che denotano entità percepibili attraverso i sensi.

Event sostantivi che esprimono entità che accadono nel tempo[3].

Abstract entità che si riferiscono a concetti astratti, come sentimenti, ideali, ...

All'atto pratico sono state adottate le seguenti strategie:

- I tag sono stati utilizzati in formato IOB[4].
- Sono state considerate MultiWord Expression solo le sequenze rigidamente non composizionali.
- I gruppi di entità sono stati etichettati come le entità di cui è formato il gruppo. Ad esempio consideriamo il sostantivo *classe*: a seconda del contesto in cui viene utilizzato, si trova ad indicare una classe di entità ontologicamente diverse, e possono dunque essergli attribuiti tag diversi a seconda della categoria delle entità che raggruppa.

[3]rispetto alle entità concrete, che invece *esistono*. Possiamo applicare il test x era ancora qui oggi $\rightarrow$ $x \in$ entità concrete, x è risuccessa oggi, ha avuto luogo ieri $\rightarrow$ $x \in$ eventi

[4]Inside, Outside, Beginning. Il prefisso B- indica che il filler è l'inizio di un chunk, il prefisso I- indica che il filler è interno al chunk e il tag O indica che il filler non appartiene a nessun chunk.

- entità che specificano un NP[5] che ne descrive il tipo, tendono ad ereditare il tipo della testa dell'NP padre. Ad esempio in *stato di choc*, sebbene *choc* funga da argomento generale astratto, eredita il tipo da *stato* e diventa di tipo EVENT, ossia indica appunto uno stato.

4.3 Casi notevoli

Si riportano di seguito alcuni esempi che mettono in luce i casi di ambiguità riscontrati.

Casi di omonimia sono evidenziati dal seguente esempio:

(7) a. Il *caccia*$_{\text{OBJECT}}$ è troppo veloce e il radar troppo poco potente; [...].

b. La ragazza è stata rilasciata dopo cinque ore e si è ripresentata a casa sconvolta, nel pieno della *caccia*$_{\text{EVENT}}$ ai sequestratori.

Situazioni di polisemia invece si ravvisano in casi del genere:

(8) a. Nel 1993 _ cito sempre dal rapporto Anee _ sono stati venduti 2,7 milioni di *lettori*$_{\text{OBJECT}}$ di cd-rom negli Stati Uniti, 900 mila in Asia e 400 mila in Europa.

b. Ho visto subito forte l' Argentina, come sanno i miei *lettori*$_{\text{ANIMATE}}$.

Una serie di fenomeni di alternanza di senso sono poi ricorrenti e permettono di identificare alcuni pattern notevoli. Un caso diffuso è il passaggio al supersenso

[5] *Noun Phrase*

ANIMATE nel caso in cui il lemma, propriamente indicante eventi o località o oggetti, si trovi a indicare l'insieme di agenti che rispettivamente partecipano, risiedono o muovono le entità sopra definite.

Riportiamo il caso del lemma *Lombardia* nei due esempi:

(9) a. Lunga catena di vittime, *Lombardia*$_{\text{ANIMATE}}$ in lutto

b. Ho telefonato anche a istituti della *Lombardia*$_{\text{LOCATION}}$ e dell'Emilia Romagna

Similmente si confronti l'esempio 7b con la seguente:

(10) a. Questo ente ha ereditato tutte le attività del gruppo assicurativo non compatibili con una gestione privatistica: [...] la gestione [...] del fondo di garanzia vittime della strada , del fondo di solidarietà per le vittime dell' estorsione , e del fondo di garanzia per le vittime della *caccia*$_{\text{ANIMATE}}$.

O si consideri il caso del lemma *colonna*:

(11) a. Frequenti erano i tavoli ovali e ancor più le scrivanie, specie per signora, complete di poggiapiedi imbottito, con un' alzatina elicoidale con cassettini e due *colonne*$_{\text{OBJECT}}$ laterali che potevano contenere delle piante.

b. Al confine con la Bosnia settentrionale si è formata una *colonna*$_{\text{ANIMATE}}$ lunga 10 chilometri di profughi in marcia, in auto e a piedi.

Un fenomeno simile si ravvisa in casi in cui lemmi che indicano entità percepibili con i sensi, dunque oggetti, quali *libro*, *pellicola* o simili, si denota il contenuto di questi ultimi, chiaramente di senso astratto. Si confrontino:

(12) a. Il *libro*$_{\text{OBJECT}}$ era di cento pagine e costava sei lire.

b. La scelta del titolo d'un *libro*$_{\text{ABSTRACT}}$ è spesso motivo di angustie per l'autore.

Un ultima categoria va considerata per quei contesti metaforici del tutto astratti e più o meno frequenti nell'uso in cui vengono impiegati lemmi che, pur conservando in funzione metaforica parte del loro significato concreto, vengono classificati come astratti:

(13) a. Poco dopo li fucilarono davanti a una folla delirante: alcune salme vennero gettate nel *fiume*$_{\text{LOCATION}}$, altre fatte a pezzi e vendute sui banchi del mercato.

b. Tutti caduti nel grande calderone della maxitangente miliardaria di Enimont o travolti dal *fiume*$_{\text{ABSTRACT}}$ dei finanziamenti illeciti partiti dal gruppo Ferruzzi.

L'annotazione di contesti metaforici non risulta sempre chiara, a causa anche del fatto che alcune metafore risultano più comuni di altre, e che l'identificazione di un uso metaforico dipende dalla conoscenza di un contesto più ampio. Consideriamo l'esempio:

(14) Poi il pallino è tornato nelle *mani* di Tronchetti Provera.

In casi come questo le informazioni contenute nella porzione di contesto a noi noto sono troppo scarne per capire se il lemma *mani* sia usato metaforicamente e sia quindi il caso di attribuirgli il tag ABSTRACT, come è probabile che sia, o se il *pallino* sia fisicamente tornato nelle mani di qualcuno, e il token sia dunque da taggare come OBJECT.

Se infatti casi come *fare le valigie* sono comuni in vari registri linguistici e il lemma *valigia* viene facilmente percepito come astratto, sebbene mantenga un comportamento tipico di un oggetto concreto, meno chiari risultano casi come *semaforo*, per cui riportiamo gli esempi:

(15) a. Miloud Belkhaoua, 14 anni, si guadagna da vivere facendo il lavavetri ai *semafori*$_{\text{LOCATION}}$ di Casalpalocco.

b. ... il regolamento dovrebbe ricevere il *semaforo*$_{\text{ABSTRACT}}$ verde ai primi di giugno.

Capitolo 5

Annotazione

Si descrive il processo di annotazione portato avanti, analizzando l'agreement sul tagset e il livello di ambiguità risultante sulla risorsa.

5.1 Intercoder agreement

L'annotazione manuale porta con sé un certo grado di soggettività che può inficiare il grado di coerenza della risorsa annotata. Bisogna quindi definire in che misura i dati annotati sono affidabili rispetto al task, in termini di quanto gli annotatori si accordano sulle categorie da assegnare. Questo è un primo passo verso la valutazione della validità del tagset: è chiaro che molti altri fattori possono essere coinvolti (livello di istruzione, pregiudizi personali).

Prima di procedere con una breve trattazione, introduciamo la notazione utilizzata.

Consideriamo I l'insieme delle istanze, K l'insieme delle categorie, C l'insieme degli annotatori, di cardinalità rispettivamente i, k, c.

Consideriamo inoltre l'*observed agreement* A_o, l'*expected agreement* A_e, $P(\cdot)$ la probabilità di un evento e $\hat{P}(\cdot)$ la stessa probabilità, approssimata a partire dai dati.

La più semplice delle misure di accordo (Scott 1955) è l'*observed agreement* (o percentuale di agreement), ovvero il numero di elementi su cui gli annotatori si accordano, diviso per il numero totale di items.

Considerata agr_i la funzione di agreement su ogni item, l'*observed agreement* si trova quindi ad essere espresso dalla seguente formula:

$$A_o = \frac{1}{\mathbf{i}} \sum_{i \in I} agr_i \tag{5.1}$$

Definiamo[1] qui agr_i nel caso generale in cui sono coinvolti un numero arbitrario di annotatori[2]:

[1]Qui e nelle successive formule consideriamo $\mathbf{n}_{ik}$ il numero di coder che hanno assegnato l'item i alla categoria k, $\mathbf{n}_{ck}$ il numero di item assegnati dal coder c alla categoria k, $\mathbf{n}_k$ il numero totale di items assegnati da tutti i coder alla categoria k

[2]o pairwise agreement, definito in Fleiss 1971. Nel caso di due annotatori la formula può essere riscritta in questo modo:

$$agr_i = \begin{cases} 1 & \text{se i due coder assegnano l'item } i \text{ alla stessa categoria } k \\ 0 & \text{altrimenti} \end{cases}$$

$$agr_i = \frac{1}{\binom{\mathbf{c}}{2}} \sum_{k \in K} \binom{\mathbf{n}_{ik}}{2} = \frac{1}{\mathbf{c}\,(\mathbf{c}-1)} \sum_{k \in K} \mathbf{n}_{ik}\,(\mathbf{n}_{ik}-1) \tag{5.2}$$

Con la misura così definita l'*observed agreement* risulta quindi:

$$A_o = \frac{1}{\mathbf{ic}\,(\mathbf{c}-1)} \sum_{k \in K} \sum_{i \in I} \mathbf{n}_{ik}\,(\mathbf{n}_{ik}-1) \tag{5.3}$$

Considerato da solo, tuttavia, l'*observed agreement* è soggetto a fattori di distorsione: intanto parte della misurazione può essere influenzata dal caso, e la percentuale di annotazione casuale è fortemente dipendente dal task. Inoltre, a parità di fenomeno, questa misura privilegia tagset con un minor numero di categorie. Bisogna poi considerare il caso (come il nostro) in cui una categoria è nettamente prevalente rispetto alle altre, e l'*observed agreement* non tiene in considerazione la probabilità inerente di ogni categoria.

Tenendo in considerazione la percentuale di agreement avvenuto per caso, definiamo il coder agreement come segue:

$$S, \pi, \kappa = \frac{A_o - A_e}{1 - A_e} \tag{5.4}$$

A_e è l'*expected agreement*, ovvero l'agreement che ci aspetteremmo se gli annotatori annotassero alla cieca. $1-A_e$ è quindi la quantità di miglioramento ottenibile. A_o-A_e ci dice quindi quanto agreement c'è effettivamente stato.

In generale possiamo assumere che venga assegnato un solo tag a ogni filler e che gli annotatori annotino indipendentemente, ottenendo, per una coppia di annotatori:

$$A_e^{i,j} = \sum_{kinK} P\left(k|c_i\right) \cdot P\left(k|c_j\right) \tag{5.5}$$

Il modo in cui calcolare $P\left(k|c_i\right)$ dà comunque luogo a diverse misure, che ricapitoliamo brevemente:

- possiamo assumere che la distribuzione di categorie sia uniforme per tutti gli annotatori (coefficente S, Bennett et al. 1954[3])
- possiamo assumere che la distribuzione delle categoria sia non uniforme, ma la stessa per tutti gli annotatori (coefficente π, Scott 1955)
- possiamo assumere che ogni annotatore operi secondo una personale distribuzione di probabilità sulle categorie, possibilmente diversa da quella degli altri (coefficente κ, Kohen 1960)

5.2 Il corpus

Il corpus preso in considerazione è stato quello fornito per EVALITA 2011, dunque circa 300000 token provenienti da ISST-SST e dalla Wikipedia Italiana.

A differenza di quanto fatto per EVALITA, qui il corpus è stato annotato automaticamente fino al livello sintattico con gli strumenti messi a disposizione dalla pipeline TANL[4].

[3]anche noto in letteratura come G, κ_n, C, RE (Zwick 1988, Hsu e Field 2003)

[4]Text Analytics and Natural Language - `http://medialab.di.unipi.it/wiki/Tanl`

Sono dunque state eliminate porzioni inconsistenti come frasi non concluse o intere porzioni semanticamente poco significative, come quelle mostrate in Tabella 5.1, giungendo così alla composizione mostrata in Tabella 5.2.

Nessuna ulteriore operazione di preprocessing è stata portata avanti sul corpus, che non costituisce dunque un gold standard.

In riferimento alle categorie previste dal tagset TANL per le parti del discorso[5], i sostantivi presenti nel corpus si suddividono in 4 classi come mostrato in Tabella 5.3.

5.2.1 Annotazione

Il lavoro di annotazione del corpus utilizzato per il lavoro presentato qui ha coinvolto tre annotatori, durante l'attività di tirocinio curriculare svolta presso il CoLing Lab.

L'affidabilità dell'annotazione è stata testata calcolando l'intercoder agreement (si veda capitolo 5.1) tra gli annotatori. Dopo una fase di addestramento su esempi significativi, ai tre annotatori è stata sottoposta separatamente la stessa porzione di corpus: 10 documenti scelti a caso dalla porzione di corpus riservata al training, per un totale di 1630 sostantivi.

I risultati mostrano un accordo soddisfacente (simple agreement - Fleiss' κ: 76.1%, con $p < 0.001$) su tutte le classi presenti nell'ontologia, in riferimento alla scala presentata in Landis e Koch 1977.

[5] `http://medialab.di.unipi.it/wiki/Tanl_POS_Tagset`

Tabella 5.1: Esempio di porzione di corpus eliminata

1	"	"	F	FB	_	2	punc
2	Il	Il	S	SP	_	0	ROOT
3	Lido	Lido	S	SP	_	2	concat
4	"	"	F	FB	_	5	punc
5	,	,	F	FF	_	2	con
6	"	"	F	FB	_	7	punc
7	The	The	S	SP	_	2	con
8	Beach	Beach	S	SP	_	7	concat
9	"	"	F	FB	_	10	punc
10	,	,	F	FF	_	2	con
11	"	"	F	FB	_	12	punc
12	La	il	R	RD	num=s|gen=f	13	det
13	Mirage	Mirage	S	SP	_	2	conj
14	"	"	F	FB	_	15	punc
15	,	,	F	FF	_	2	con
16	"	"	F	FB	_	17	punc
17	The	The	S	SP	_	2	conj
18	Palace	Palace	S	SP	_	17	concat
19	"	"	F	FB	_	17	punc
20	.	.	F	FS	_	2	punc

Tabella 5.2: Composizione Corpus

Documenti	487
Frasi	12486
Token	314132
Sostantivi	85963

Tabella 5.3: Composizione Sostantivi

PoS	**Descrizione**	**Token**
S	Sostantivi	64303
SP	Nomi Propri	20759
SW	Sostantivi Stranieri	643
SA	Abbreviazioni	258

L'attività di annotazione è dunque stata portata avanti dagli annotatori indipendentemente e su porzioni disgiunte di corpus e la risorsa finale risulta dall'unione di queste.

5.3 Valutazioni sulla composizione del corpus annotato

La distribuzione dei sostantivi nelle varie classi derivate dall'ontologia presentata in 4.1 risulta come in Tabella 5.5.

Procedendo a un'analisi più approfondita che prenda in considerazione l'ambiguità

Tabella 5.4: Dettaglio agreement per classe

	Fleiss'-κ	**p-value**
ABSTRACT	0.716	0.000
ANIMATE	0.902	0.000
EVENT	0.655	0.000
LOCATION	0.833	0.000
OBJECT	0.654	0.000

Tabella 5.5: Distribuzione istanze nelle classi del tagset

Classe	**Token**	**Lemmi**
ABSTRACT	34785	5219
ANIMATE	23662	6255
EVENT	9362	2087
LOCATION	8029	2203
OBJECT	5248	1847
O	4877	1544

Tabella 5.6: Copertura lemmi non ambigui

Classe	**Lemmi**	**Token**	**Lemmi / tot_Lemmi**	**Token / tot_Token**
ANIMATE	4970	13866	0,327	0,161
ABSTRACT	2975	11154	0,523	0,291
LOCATION	1394	2939	0,615	0,325
OBJECT	1162	2317	0,691	0,352
EVENT	904	2160	0,751	0,377
O	759	1069	0,801	0,390

rispetto al lemma, la distribuzione è riportata in Tabella 5.6. I lemmi non ambigui rappresentano dunque l'80.1% del totale, coprendo però solo il 38.9% dei token.

I restanti 3020 lemmi risultano ambigui, ma in misura diversa.

Si nota infatti che una parte di questi (1936, il 12,7% del totale) sono in realtà ambigui solo se si considerano le istanze provenienti da documenti diversi: limitatamente ad una plausibile finestra di contesto quale può essere un documento, il livello di ambiguità decresce dunque fortemente. Ciò suggerisce che l'uso di informazione globale potrebbe migliorare nettamente le capacità di disambiguazione e che l'informazione di appartenenza a un dato dominio o sottodominio è estremamente rilevante per il tipo di task.

Un altro aspetto da considerare per una valutazione sui livelli di ambiguità risultanti dall'annotazione è la preponderanza, in termini di frequenza, di un dato senso rispetto agli altri.

Definiamo:

$S_l = \{s \mid \exists\, t$ token con lemma l e categoria $s\}$ l'insieme dei sensi tra i quali un lemma risulta ambiguo
$occ\,(l, s) = |\{t \mid\ t.lemma = l\ e\ t.categoria = s\}|$ il numero di occorrenze di un senso di un lemma nel corpus
$s^\star = s \in S_l \mid \forall\, s' \in S_l\ occ\,(l, s) \geq occ\,(l, s')$ il senso più frequente di un lemma nel corpus

Una misura di ambiguità per un lemma può essere definita come segue:

$$A\,(l) = \frac{1}{|S_l| - 1} \sum_{s \in S_l \setminus \{s^\star\}} \frac{occ\,(l, s)}{occ\,(l, s^\star)} \tag{5.6}$$

Così definita $A\,(l) \to 1$ se le occorrenze dei vari sensi del lemma risultano bilanciate, mentre $A\,(l) \to 0$ nel caso in cui uno dei sensi risulti nettamente prevalere sugli altri.

Raggruppiamo dunque i lemmi in esame in gruppi così definiti come nella formula 5.7, ottenendo quanto riportato in Tabella 5.7.

$$G(a, b) = |\{l \mid a < A\,(l) \leq b\}|\ \text{ con } a, b \in [0, 1] \tag{5.7}$$

I dati mostrano che nella maggior parte dei casi esiste un senso che ha il doppio o più delle occorrenze degli altri, e che quando ciò non è vero le occorrenze si distribuiscono in porzioni di testo disgiunte, che spesso implica sottodomini diversi.

Tabella 5.7: In tabella è mostrato il valore della funzione 5.7 negli intervalli indicati, sia considerando l'ambiguità globale che restringendoci al caso di lemmi ambigui nella stessa porzione o documento del corpus. L'ultima colonna mostra che la differenza tra i valori della funzione nei due casi sopra menzionati.

		nello stesso documento		globale		
a	**b**	**G(a,b)**	%	**G(a,b)**	%	Δ
0	0,1	163	0,215	341	0,153	0,06
0,1	0,2	145	0,191	357	0,160	0,03
0,2	0,3	111	0,146	253	0,113	0,03
0,3	0,4	74	0,097	283	0,127	0,03
0,4	0,5	107	0,141	389	0,174	0,03
0,5	0,6	29	0,038	65	0,029	0,01
0,6	0,7	36	0,047	87	0,039	0,01
0,7	0,8	29	0,038	66	0,030	0,01
0,8	0,9	19	0,025	28	0,013	0,01
0,9	1	46	0,061	364	0,163	**0,10**

Capitolo 6

Classificazione

Si descrive il processo di classificazione: l'algoritmo utilizzato, le features estratte e il loro peso nel modello finale.

6.1 Support Vector Machine

L'algoritmo scelto per la creazione del modello è una Support Vector Machine (SVM): introdotte in Cortes e Vapnik 1995 estendono i modelli lineari guadagnando capacità di generalizzazione alla luce del principio di Minimizzazione del Rischio Strutturale derivato dalla teoria Statistica dell'Apprendimento[1]: come accade nei modelli lineari, infatti, a partire da una rappresentazione vettoriale dei dati in uno spazio m-dimensionale, si cerca il miglior iperpiano di decisione (di equazione $\vec{w} \cdot \vec{x} + b = 0$)

[1] Vapnik e Vapnik 1998

che permetta di dividere lo spazio in due regioni, ognuna delle quali contiene istanze di una sola delle due classi in esame, secondo la relazione:

$$\begin{cases} \vec{w} \cdot \vec{x}_i + b > 0 \text{ se } y_i = 1 \\ \vec{w} \cdot \vec{x}_i + b < 0 \text{ se } y_i = -1 \end{cases} \tag{6.1}$$

Quello che distingue la SVM da un modello lineare è il principio secondo il quale, nel cercare la coppia $(\vec{w}, b)$ di coefficienti che rispetti l'equazione 6.1, invece di cercare di minimizzare il rischio empirico tramite la relazione

$$\vec{w}^{\star} = \underset{w}{\operatorname{argmin}} \, E\left(\vec{w}\right) \tag{6.2}$$

cerchiamo piuttosto di trovare la coppia $(\vec{w}, b)$ tale per cui sia massimo il margine tra l'iperpiano costruito e i vettori delle due classi più prossimi a questo, i così detti vettori di supporto.

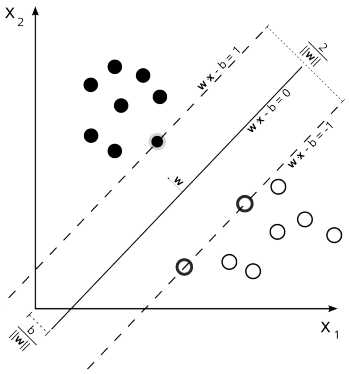


Figura 6.1: Iperpiano costruito dalla SVM

È possibile ridursi alla situazione in cui, definito l'insieme dei vettori di supporto, questi giacciano sui due iperpiani $\vec{w} \cdot \vec{x} + b = 1$ e $\vec{w} \cdot \vec{x} + b = -1$. In questa situazione, il margine da massimizzare è il doppio della distanza di questi punti dall'iperpiano,

e la sua dimensione è dunque $\frac{2}{\|\vec{w}\|}$. A questo fine è necessario minimizzare $\|\vec{w}\|$ con $y_i\,(\vec{w}\cdot\vec{x}_i+b)\geq 1$ per ogni $(\vec{x}_i, y_i)$ esempio di training.

In fase di test, per stabilire la classe di $\vec{x}^\star$, si calcola il segno di $\vec{w}\cdot\vec{x}^\star+b$: se risulta <0 gli si assegna la classe -1, altrimenti la classe 1. Per questo motivo, più ampio è il margine, più probabile sarà classificare correttamente le nuove istanze.

È inoltre possibile, al fine di aumentare la flessibilità del modello riducendo l'overfitting, ammettere qualche errore in fase di training introducendo N variabili di errore z_i non negative, una per ogni istanza di input.

Se $z_i > 0$, significa che stiamo tollerando un errore sull' i-esimo dato in input.

La quantità da minimizzare diventa quindi $\|\vec{w}\|+C\sum_i z_i$ con i vincoli $(\vec{w}\cdot\vec{x}_i+b)\,y_i \geq 1-z_i$. C è un iperparametro che stabilisce il contributo degli z_i nella funzione da minimizzare.

Per effettuare una classificazione che coinvolga più di due classi, si possono utilizza due algoritmi, denominati OVA (One Versus All) e AVA (All Versus All). Nel primo caso ogni classe viene confrontata con la totalità delle altre, nel secondo ogni classi viene confrontata con ognuna delle altre.

6.1.1 scikit-learn

L'implementazione dell'algoritmo utilizzata in questo lavoro è quella fornita dalla libreria scikit-learn[2], una libreria open source di Machine Learning in Python, che

[2]Pedregosa et al. 2011

mette a disposizione funzionalità per preprocessing e classificazione dei dati. Per lo sviluppo del modello abbiamo utilizzato la classe `LinearSVC`, che implementa una SVM con kernel lineare, basandosi su `liblinear`[3]. Per gestire la classificazione multiclasse `LinearSVC` utilizza l'approccio OVA. Per tutte le prove qui considerate, non sono state apportate modifiche agli iperparametri di default della classe.

6.1.2 Valutazione dei sistemi di apprendimento automatico

Per valutare la bontà del modello creato, è possibile utilizzare misure statistiche. Riportiamo qui una breve descrizione delle principali, utilizzate in fase di selezione e testing dei nostri modelli. Per semplicità facciamo riferimento al caso di una classificazione binaria.

Sottoponendo al modello N istanze di cui conosciamo l'appartenenza, possiamo valutare la bontà della classificazione considerando, in termini descritti dalla matrice di confusione in Tabella 6.1:

Accuracy Il numero di istanze classificate correttamente rispetto al numero totale di istanze.

$$Accuracy = \frac{\sum veri\ positivi + \sum veri\ negativi}{\sum istanze} \tag{6.3}$$

Precision Il numero di istanze della classe A correttamente classificate rispetto al numero totale di istanze della classe A

$$Precision = \frac{\sum veri\ positivi}{\sum veri\ positivi + \sum falsi\ positivi} \tag{6.4}$$

[3]Fan et al. 2008

Recall Il numero di istanze correttamente assegnate alla classe A, rispetto al numero totale di istanze assegnate ad A

$$Recall = \frac{\sum veri\ positivi}{\sum veri\ positivi + \sum falsi\ negativi} \tag{6.5}$$

Le misure sopra descritte misurano tipologie di errori diversi e nessuna è dunque sufficiente da sola a descrivere la bontà del sistema.

A seconda del tipo di task, potrebbe inoltre essere preferibile ammettere, ad esempio, un valore di *precision* inferiore per guadagnarne in *recall*[4], o viceversa.Si definisce dunque la **F-measure** come la media armonica tra *precision* e *recall*.

$$F_1 score = 2 \cdot \frac{precision \cdot recall}{precision\ +\ recall} \tag{6.6}$$

Tabella 6.1: La tabella mostra un esempio di classificazione binaria di un insieme di elementi. Sulle colonne si trovano gli elementi appartenenti alla classe corrispondente, su ogni riga gli elementi che vengono assegnati alla classe relativa. Considerando la classe A, sulla prima riga troviamo i *veri positivi*, ovvero elementi di classe A correttamente classificati, e i *falsi positivi*, ovvero elementi di classe B classificati come appartenenti alla classe A. Sulla seconda riga troviamo i *falsi negativi*, ovvero elementi di classe A che sono stati erroneamente classificati, e i *veri negativi*, ovvero elementi di classe B assegnati alla classe B.

	A	**B**
A	veri positivi	falsi positivi
B	falsi negativi	veri negativi

[4]Pensiamo ad esempio al caso di allarme antincendio, in cui preferiamo venire avvertiti anche per un falso allarme, piuttosto che rischiare di non essere evacuati in casi di reale pericolo.

6.2 Rappresentazione dei dati di training

Prima di prendere in considerazione le features estratte per ogni token, seguiamo l'impostazione delineata in Agirre e Stevenson 2007 nel presentare le fonti di conoscenza che abbiamo preso in considerazione per la costruzione di *LISA*. Come suggerito dagli autori, una descrizione in termini di risorse più che di features, fornisce infatti una visione più astratta del problema, e ci permette di confrontarci con la discussione portata avanti in quella sede. La divisione del contenuto del resto di questo paragrafo ricalca dunque la discussione citata.

6.2.1 Knowledge sources

Gli autori suddividono le fonti di conoscenza in tre classi, secondo modelli già elaborati in Hirst 1992, McCroy 1992, Agirre e Martinez 2001:

sintattica che raccoglie tutte le risorse che hanno a che fare con il ruolo di una parola nella struttura grammaticale della frase.

semantica che raccoglie tutte le risorse che riguardano proprietà delle entità cui le parole si riferiscono.

pragmatica che raccoglie le risorse che catturano il ruolo della parola in una finestra più ampia, quale quella del discorso. Qui non ulteriormente citate perché non utilizzate.

Dove non specificato, l'informazione è stata tratta dall'annotazione automatica del corpus.

Risorse sintattiche

Part of Speech indica la categoria grammaticale della parola. Sebbene il nostro task sia ristretto alla sola categoria dei sostantivi, l'informazione sulla Part of Speech suddivide la categoria in quattro sottocategorie, come già menzionato, isolando nomi propri, abbreviazioni e parole straniere. Dal momento che la risorsa utilizzata è annotata automaticamente, si è utilizzato il corpus di Repubblica[5] come ulteriore fonte di Part of Speech

Informazione Morfologica nel caso dei sostantivi, indica genere e numero: possiamo constatare che alcune forme morfologiche si associano più facilmente ad alcuni sensi piuttosto che ad altri, e dunque l'informazione morfologica risulta valida per la disambiguazione. Ad esempio l'informazione sul numero può dare indicazioni sull'uso personificato di un concetto astratto.

Risorse semantiche

Associazioni sintagmatiche descrivono associazioni tra parole della frase relativamente a relazioni di dipendenza sintattica. Tramite LexIt[6] è stato possibile accedere a informazioni sulla forza di associazione della relazione sintattica considerata

Preferenze combinatorie (Selectional preferences) tramite un mapping dei supersensi di WordNet nei supersensi della nostra ontologia, abbiamo ricavato il grado di preferenza di un certo filler per ogni classe semantica, relativamente alle relazioni sintattiche instaurate

[5] `http://sslmit.unibo.it/repubblica`

[6] `http://lexit.fileli.unipi.it/`

6.2.2 Features

Il set di features prese in considerazione deriva dunque in parte da quelle utilizzate nei vari modelli presentati nel capitolo 3.4, e in parte dalle considerazioni teoriche derivanti da quanto sopra citato. Ne presentiamo ora i raggruppamenti, per una lista più dettagliata si veda l'Appendice B.

Features del token corrente

Lemma lemma del filler

Part of Speech un valore tra {S, SP, SA, SW} a seconda della PoS del filler

Morfologia sia per il genere che per il numero sono stati considerati tutti i valori possibili (maschile, femminile o neutro nel primo caso, e singolare, plurale o neutro nel secondo) e da questi sono state tratte due feature categoriche.

WordShape codifica la forma grafica del token, ovvero assume il valore 1 se il target contiene una maiuscola non in prima posizione, o contiene un numero o un carattere non alfabetico.

Lemma-PoS si è presa in considerazione la frequenza dell'associazione, nel corpus di Repubblica, del lemma target con la PoS assegnatagli dal tagger automatico: l'informazione è stata codificata in una feature numerica (logaritmo della frequenza dell'associazione) e una feature booleana (frequenza relativa[7] dell'associazione maggiore di una soglia fissata a 0.1)

[7] consideriamo frequenza relativa il valore ottenuto dividendo il numero di occorrenze del lemma in associazione alla PoS presa in considerazione per la frequenza assoluta del lemma

Feature dell'aspetto ortografico gruppo di cinque features booleane che codificano informazioni tipiche degli algoritmi di Named Entity Recognition: nello specifico, se il filler è in prima posizione, se è in prima posizione nella frase e capitalizzato, se è in prima posizione ma non è capitalizzato, se è composto da due o più parti congiunte da trattino, se la form è capitalizzata.

Features locali

Pattern locali - Part of Speech è stata presa in considerazione l'informazione sulla Part of Speech (sia nella versione *coarse grained* che in quella *fine grained*) dei token linearmente successivo o precedente rispetto al token in esame

Pattern locali - Forma ortografica una feature booleana codifica se il token si trova in una porzione racchiusa da virgolette[8], altre tre feature codificano se il token si trova in una particolare sequenza di capitalizzazioni

Dipendenze sintattiche le informazioni sulle dipendenze sintattiche tra token sono state codificate in un nutrito gruppo di features, che può essere ulteriormente suddiviso come segue:

Determinanti e modificatori numerali un gruppo di feature registra la presenza di un determinante definito o indefinito e di un modificatore numerale del filler

Modificatori aggettivali gli aggettivi sono stati clusterizzati in accordo alla loro collocazione in WordNet ed è stata registrata la loro presenza come modificatore del filler in esame

[8] come virgolette sono stati presi in considerazione i caratteri ", « e »

Testa e dipendenti per ogni filler, sono state considerate le sue relazioni di dipendenza, codificandole in un gruppo di feature che tiene conto del tipo di relazione che questo instaura con la testa e con i suoi dipendenti, eventualmente della preposizione tramite cui la dipendenza si attua, dei lemmi con cui instaura le suddette relazioni e delle rispettive parti del discorso

Preferenze di selezione rispetto alle relazioni sintattiche di un token con la propria testa, si è considerata la forza di associazione della relazione, in termini di *log-likelihood*[9].

Features globali

Preferenza di supersensi Per ogni filler, è stata stimata la probabilità di appartenere a un dato supersenso attraverso la frequenza con cui la testa sintattica del filler seleziona il supersenso in esame.

[9]Nel nostro caso consideriamo un evento che coinvolge due token, u e v, nel corpus. Consideriamo Log-Likelihood Ratio (Dunning 1993) la misura così definita:

$$LLR(u,v) = f(u,v) \times \log \frac{p(u,v)}{p(u) \times p(v)} \tag{6.7}$$

6.3 Feature selection

Il corpus annotato è stato suddiviso ricalcando la suddivisione tra training set e test set della risorsa fornita per EVALITA 2011[10].

Tra le features sopra menzionate sono poi stati effettuati dei raggruppamenti significativi dal punto di vista dell'informazione fornita e tramite questi è stata portata avanti una fase di *model selection* che ha permesso la selezione dei gruppi più promettenti di features. Questi modelli sono stati poi testati sul test set per una valutazione delle prestazioni.

Un nucleo base di features è stato individuato in Attardi et al. 2013, considerando il set di features lì utilizzato ad eccezione dell'informazione lessicale. Allargando questo nucleo più gruppi sono stati formati come riassunto Tabella 6.3.

I sette modelli risultanti sono stati creati con una Support Vector Machine con Kernel Lineare e valutati sul training set tramite una 10-fold cross validation.

I grafici 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8 mostrano i risultati ottenuti su ogni classe del tagset, in termini di precision, recall e f1-measure.

Notiamo che, rispetto a quanto riportato in 3.4, la classe dei nomi astratti sembra qui la più semplice da classificare: ciò è sicuramente da attribuirsi alla maggior frequenza di questa classe rispetto alle altre.

L'informazione morfologica fornisce sicuramente un contributo al sistema in ge-

[10]poiché il test set comprende, come già detto, una porzione proveniente da Wikipedia, l'omogeneità della divisione con quanto fatto a EVALITA 2011 permette un confronto con i risultati della competizione

nerale, migliorando la precisione nel riconoscimento di ogni classe. Le feature del gruppo *ner* si mostrano più che valide per il riconoscimento della classe OTHER. I due gruppi appena citati, inoltre, riequilibrano in generale i valori di f-measure tra le classi. Ciò è particolarmente evidente guardando alla classe degli ANIMATE in cui lo scarto tra la versione base e la versione comprendente informazione morfologica è di 4 punti percentuali: se pensiamo alla gerarchia presente nell'ontologia, non dimenticando la frequenza delle varie classi nel corpus (riportata nella Tabella **??**), lo spostamento di alcune istanze sulla classe ANIMATE sembra indicare un maggior riconoscimento della dicotomia astratto/concreto, a prescindere dal valore di animatezza.

L'informazione sintattica permette di distinguere, seppur in minima percentuale, le classi a bassa frequenza (OBJECT, LOCATION, EVENT), dimostrandosi dunque valida, anche se non sufficiente, per l'obiettivo da noi considerato.

Abbiamo poi voluto valutare l'assetto del nostro miglior modello rispetto a quello presentato in Attardi et al. 2013 (le differenze tra i due insiemi di features sono riportate in Tabella 6.2). Nonostante ciò non costituisca in alcun modo un confronto con il modello sviluppato presso l'Università di Pisa in occasione di EVALITA 2011, considerate le macroscopiche differenze a livello di tagset tanto quanto di algoritmo utilizzato, ci ha comunque permesso di collocare gli score ottenuti dal nostro modello entro un panorama più definito. Da quanto riportato nei grafici 6.11, 6.13, 6.9, si vede che la presenza dei lemmi nell'insieme delle feature del modello *unipi* è fondamentale per la riuscita della classificazione. Alla luce di ciò, abbiamo riportato nei grafici 6.12, 6.14, 6.10 un confronto tra il nostro modello e un modello costruito tramite il solo uso dei lemmi: l'andamento si mostra del tutto sovrapponibile a quello ottenuto

dal modello *unipi*, confermando la rilevanza dell'informazione lessicale in un task come il nostro.

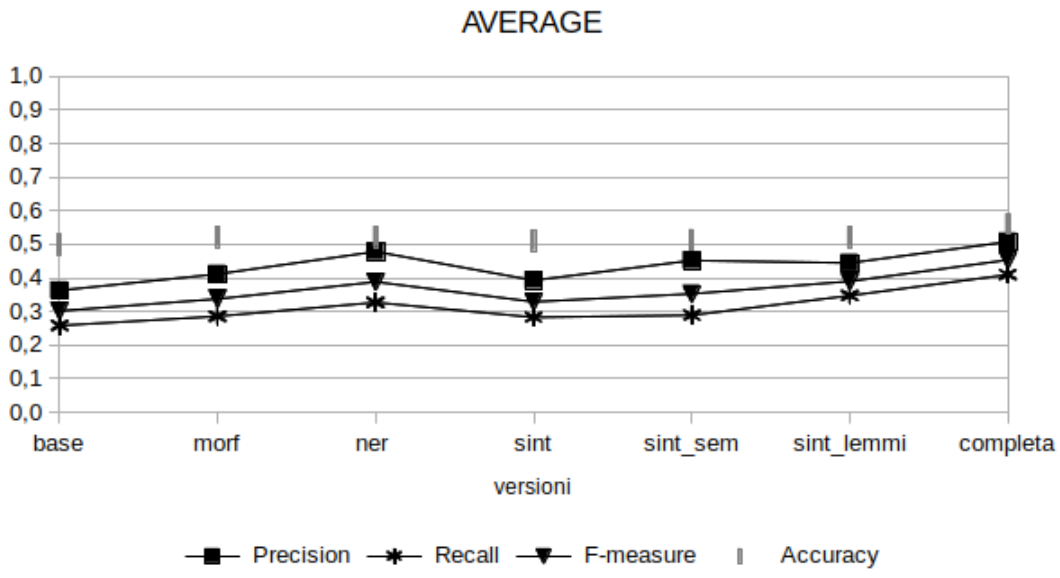


Figura 6.2: Valori di accuratezza medi ottenuti da ogni modello

Tabella 6.2: Composizione versioni *completa* e *unipi*

	completa	**unipi**
form	_	x
lemma	_	x
morfologia (genere, numero)	x	_
PoS	-1 0 1	0 1
CPoS	-1 1	-1
FirstWordCap	x	x
FirstWordNoCap	x	x
Hyphen	x	x
FrequenzaLemmaPoS_log	x	_
FrequenzaRelativaLemmaPoS	x	_
Capitalized	x	_
WordShape	x	_
FirstWord	x	_
SeqCap	x	x
CapNext	x	x
CapPrev	x	x
WithinQuotes	x	x
Informazione sintattica	x	_
Informazione semantica	x	_

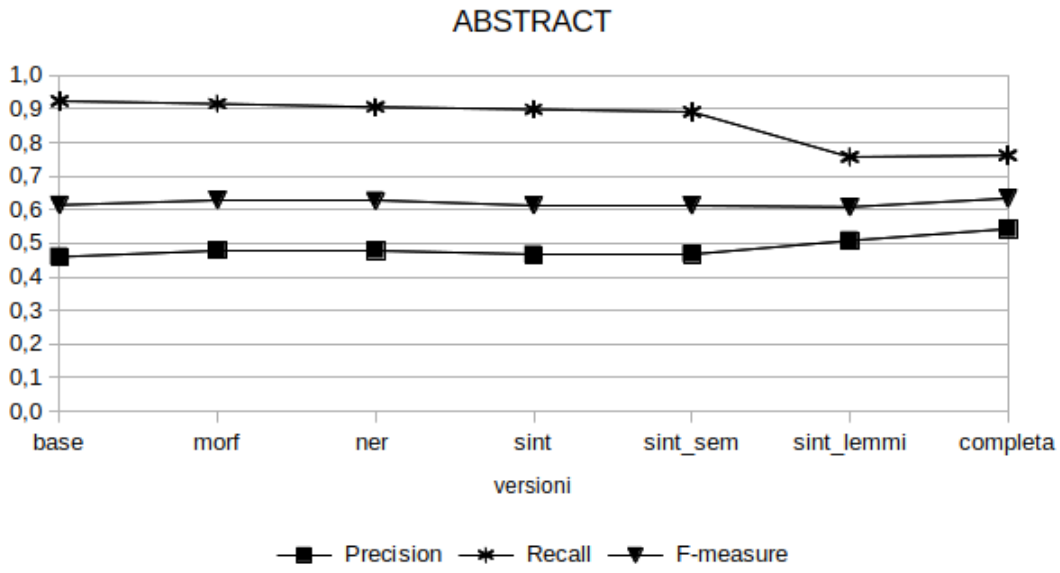


Figura 6.3: Valori di accuratezza per la classe **Abstract**

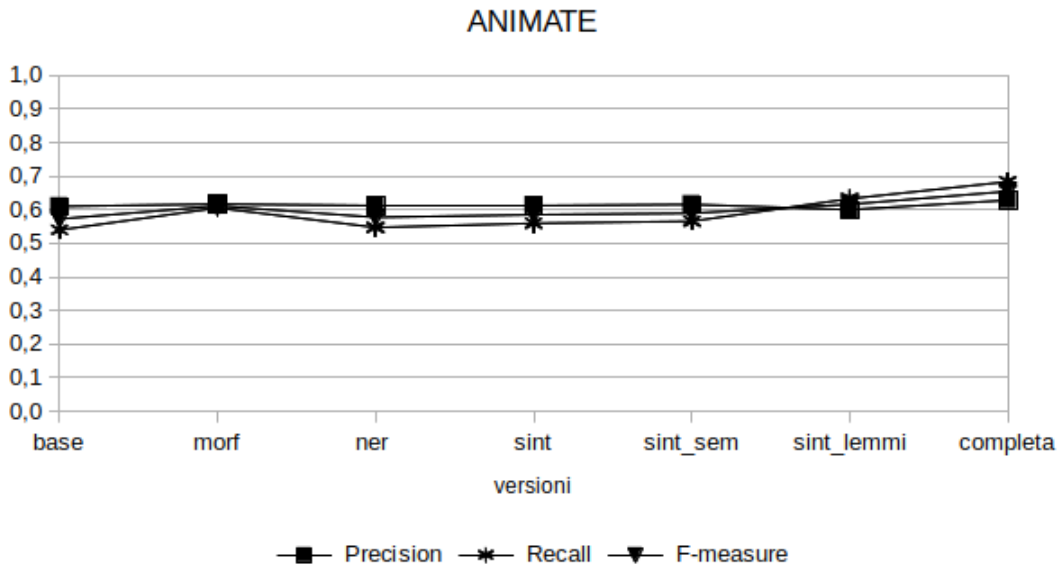


Figura 6.4: Valori di accuratezza per la classe **Animate**

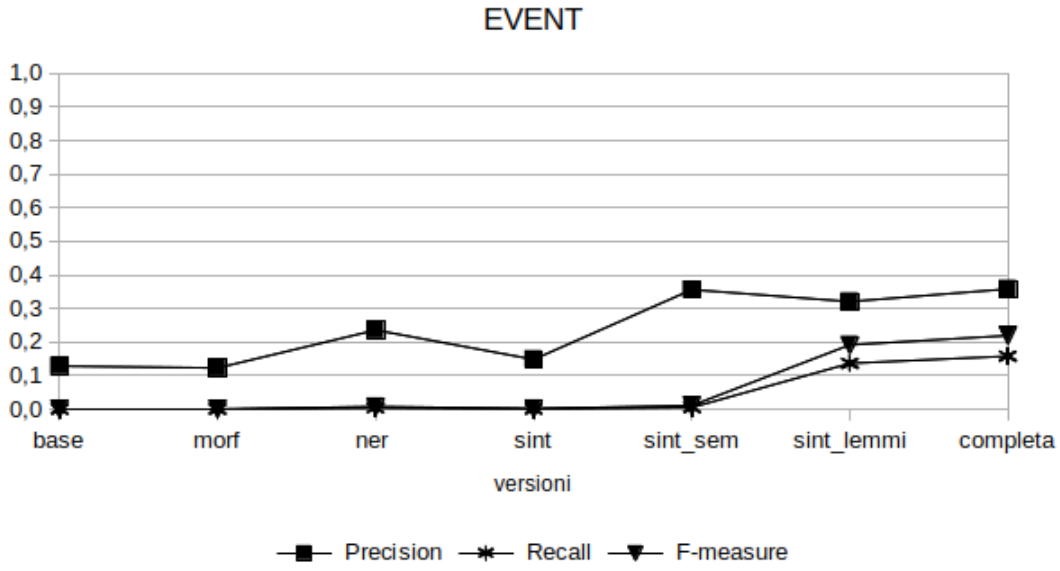


Figura 6.5: Valori di accuratezza per la classe **Event**

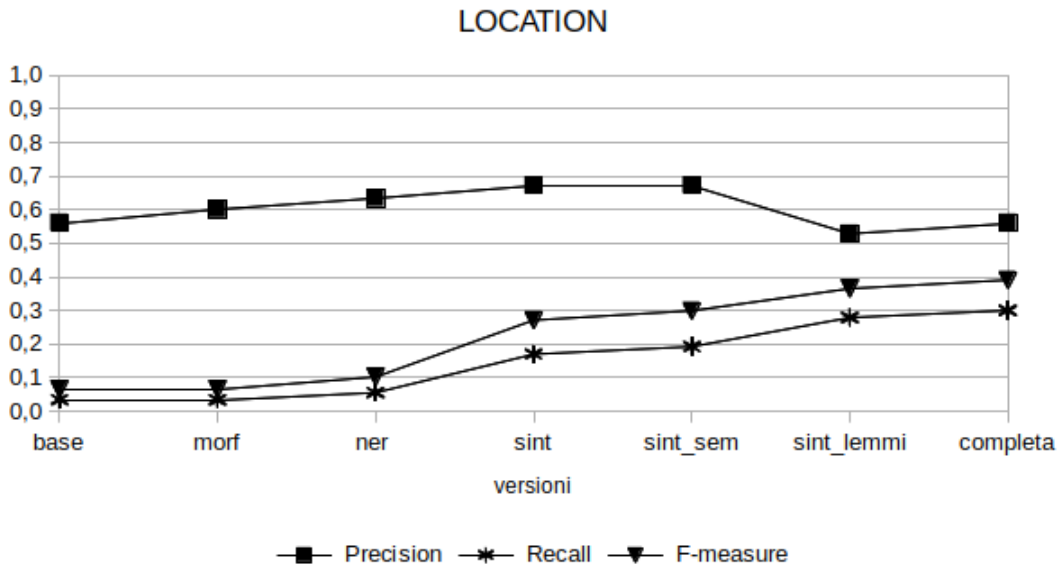


Figura 6.6: Valori di accuratezza per la classe **Location**

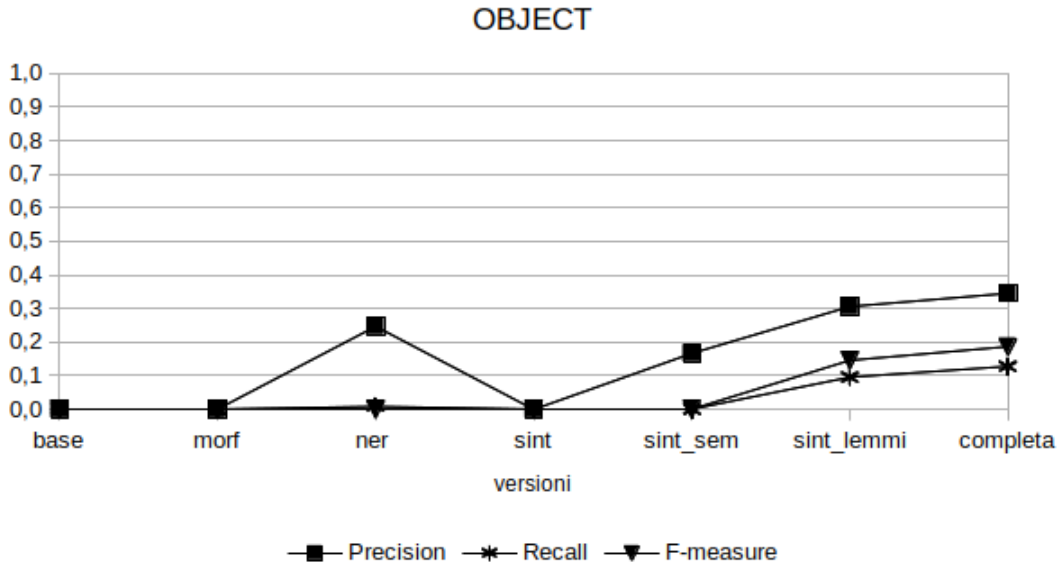


Figura 6.7: Valori di accuratezza per la classe **Object**

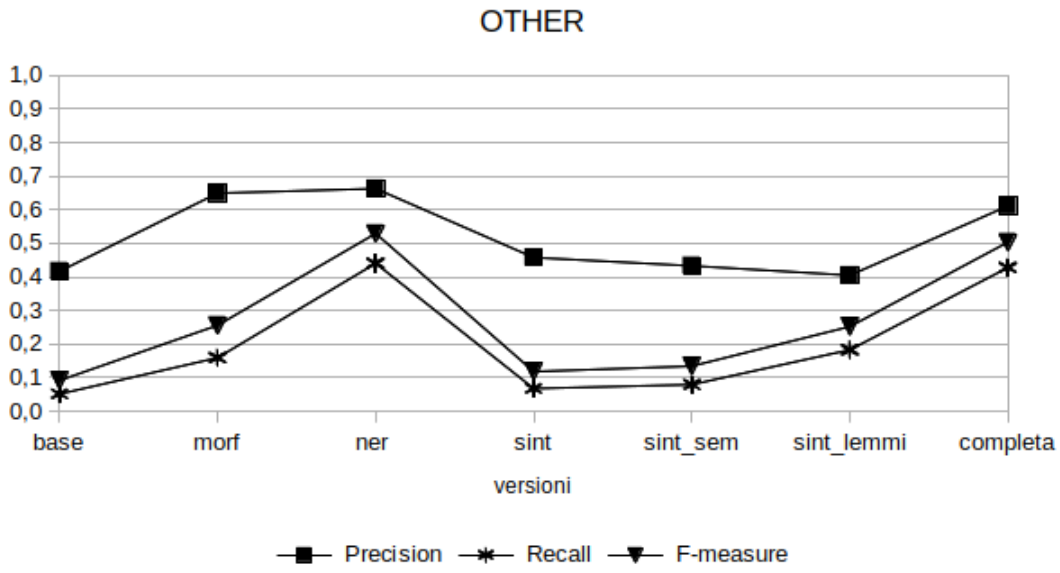


Figura 6.8: Valori di accuratezza per la classe **Other**

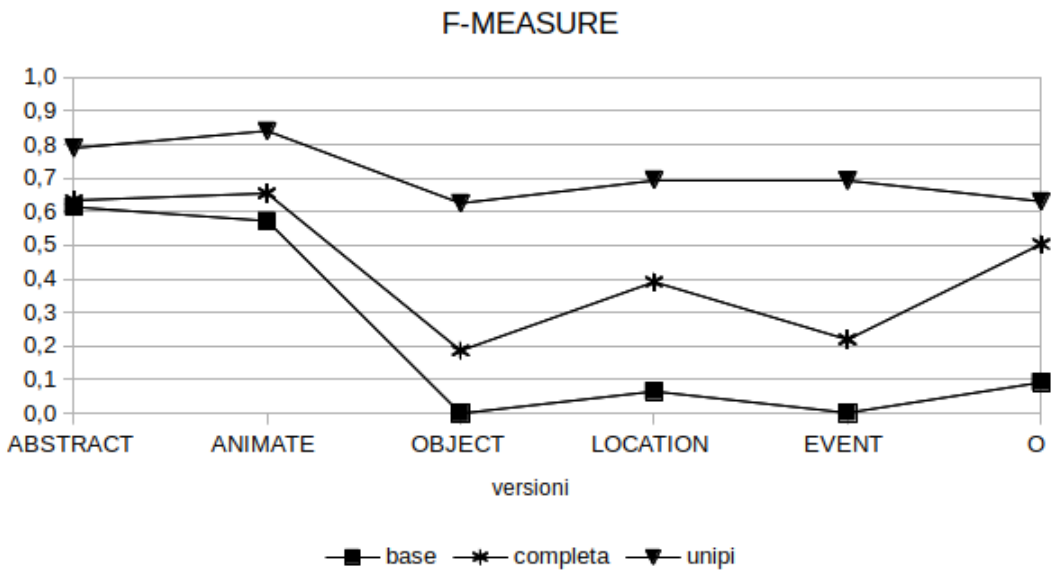


Figura 6.9: Confronto tra le versioni *base*, *completa* e *unipi* (F-Measure)

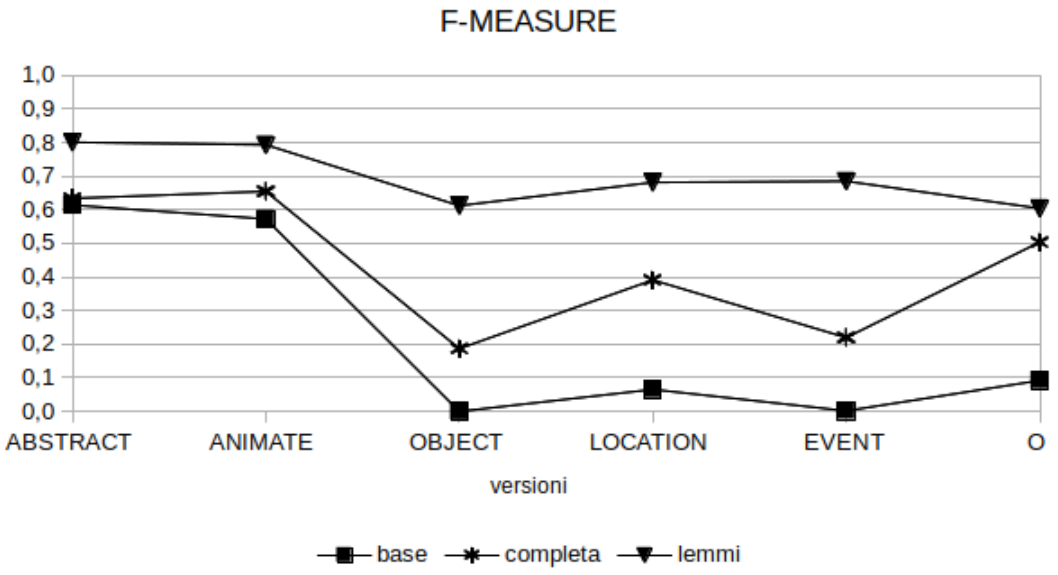


Figura 6.10: Confronto tra le versioni *base*, *completa* e *lemmi* (F-Measure)

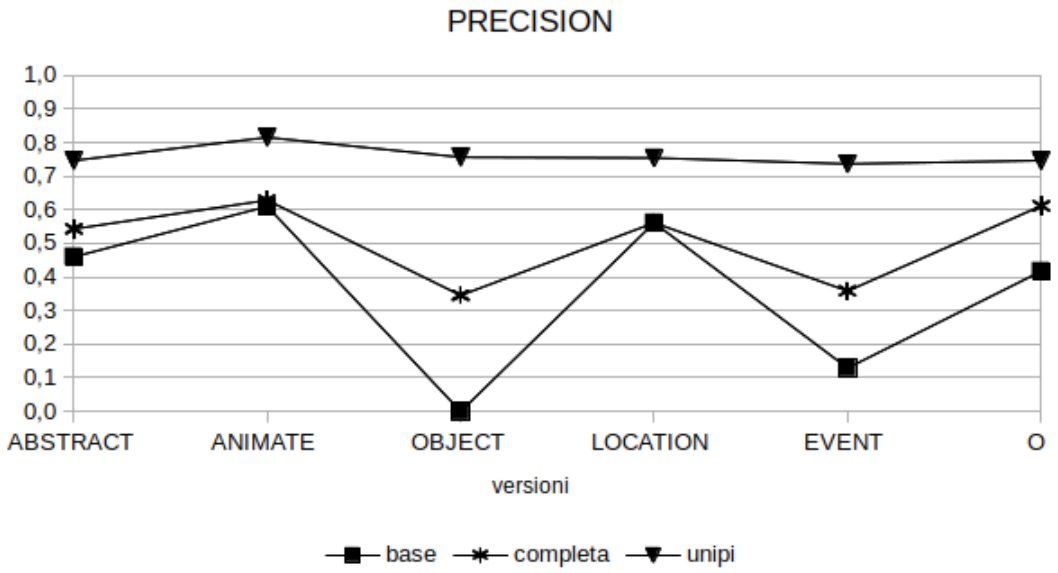


Figura 6.11: Confronto tra le versioni *base*, *completa* e *unipi* (Precision)

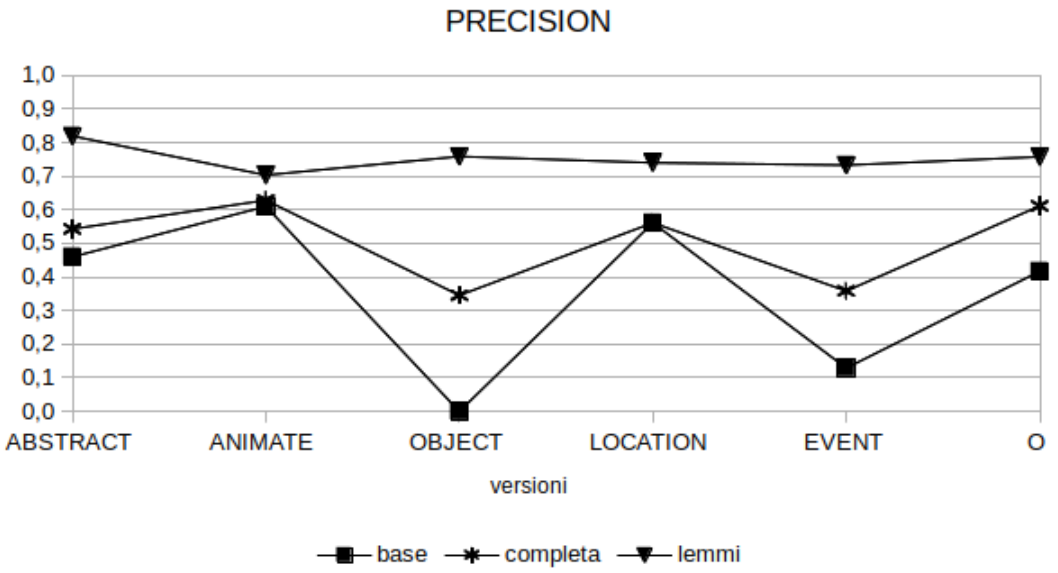


Figura 6.12: Confronto tra le versioni *base*, *completa* e *lemmi* (Precision)

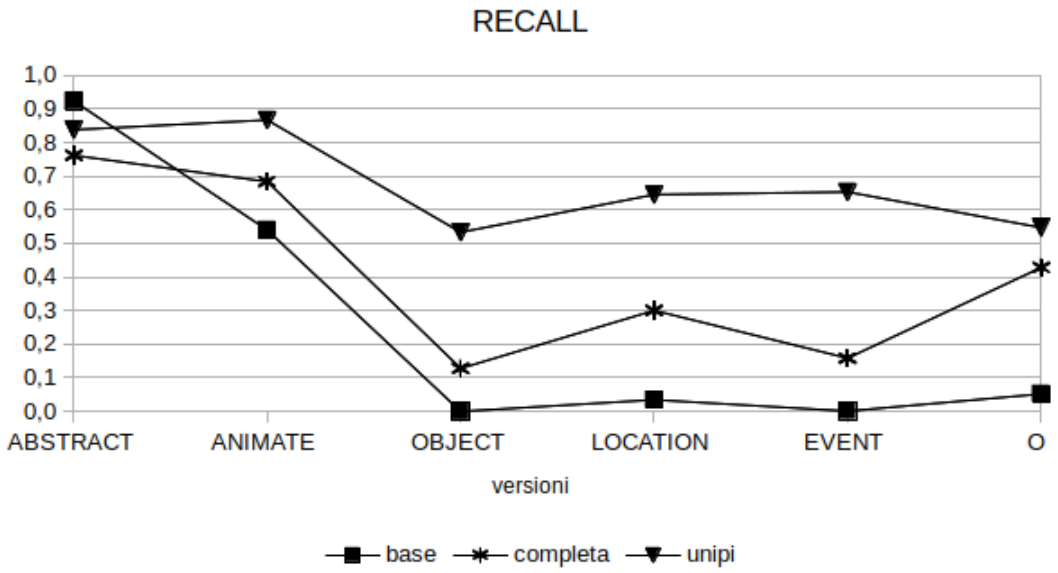


Figura 6.13: Confronto tra le versioni *base*, *completa* e *unipi* (Recall)

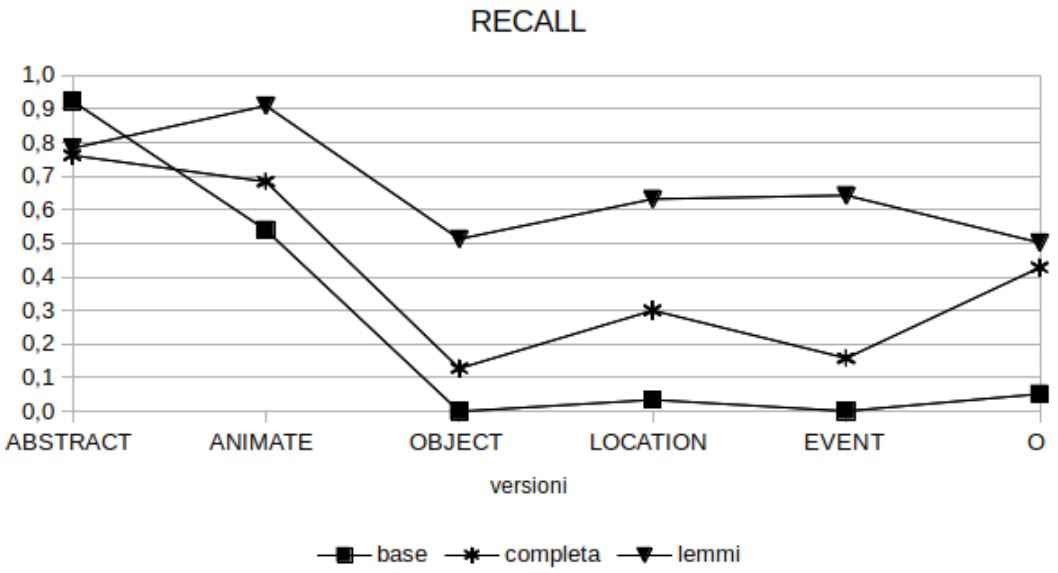


Figura 6.14: Confronto tra le versioni *base*, *completa* e *lemmi* (Recall)

Tabella 6.3: Riepilogo feature usate per ogni gruppo

	base	ner	morfologia	sintassi	sintassi_semantica	sintassi_lemmi	completa
morfologia (genere, numero)			x				x
PoS	0,1	-1,0,1	0,1	0,1	0,1	0,1	-1,0,1
CPoS	-1	-1,1	-1	-1	-1	-1	-1,1
FirstWordCap	x		x	x	x	x	
FirstWordNoCap	x		x	x	x	x	
Hyphen	x	x	x	x	x	x	x
FrequenzaLemmaPoSlog		x					x
FrequenzaRelativaLemmaPoS		x					x
Capitalized		x					x
WordShape		x					x
FirstWord		x					x
SeqCap	x	x	x	x	x	x	x
CapNext	x	x	x	x	x	x	x
CapPrev	x	x	x	x	x	x	x
WithinQuotes	x	x	x	x	x	x	x
ModAdj (pre, post)				x	x	x	x
ModAdjclusters						x	x
ModNum				x	x	x	x
Det (def, indef)				x	x	x	x
DipInv_tipo				x	x	x	x
DipInv_lemmi						x	x
DipInv_PoS					x		x
DipInv_preposizione					x		x
DipInv_forzaassociazioni					x		x
DipInv_classeassociazioni					x		x
Dip				x	x	x	x
DipPoS					x		x
Dippreposizione					x		x
Diplemmi						x	x

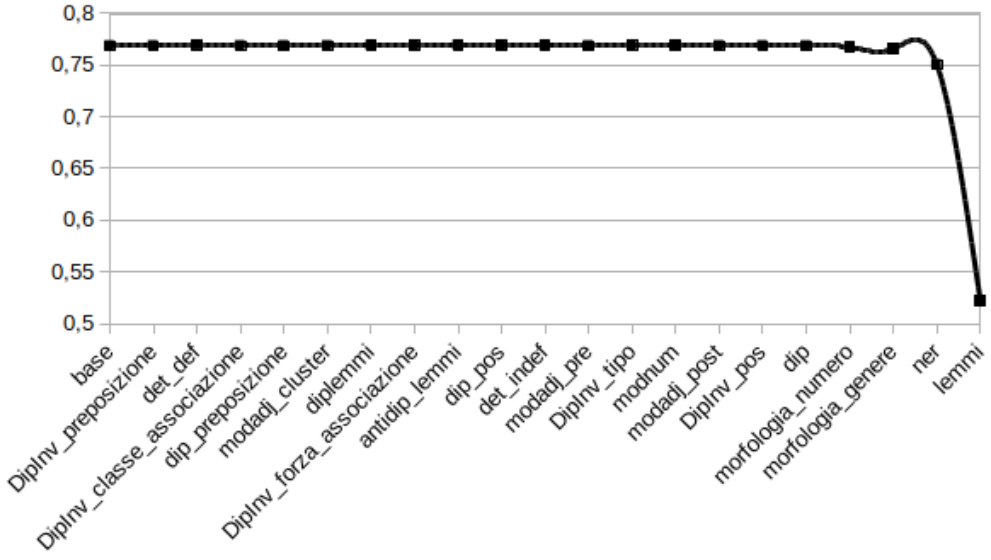
6.4 Recursive Feature Elimination

Sul modello più completo, sia nella versione con i lemmi sia in quella senza, è stata operata una Recursive Feature Elimination per valutare il peso delle features sul modello finale. A questo proposito, sono stati considerati i gruppi di features booleane presenti nella versione di partenza e a partire da questi gruppi è stato sviluppato l'algoritmo qui descritto:

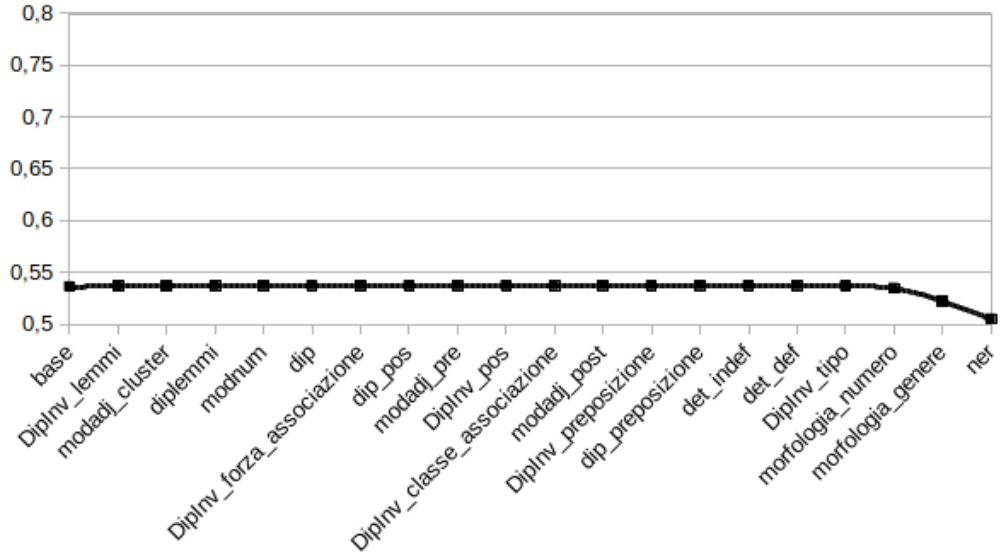
```
while len(gruppi_features)>1:
        risultati={}
        for f in gruppi_features:
                modello = crea_modello (gruppi_features - f)
                risultati[f] = valuta (modello)
        feature_peggiore = argmax(risultati)
        gruppi_features = gruppi_features - feature_peggiore
```

La valutazione di ogni modello è stata portata avanti confrontando le medie di accuracy su una 5-fold cross validation.

La scala di rilevanza delle features che i due modelli, con e senza lemmi, generano mostra qualche differenza, come si vede in Figura 6.15. Ciò è dovuto in parte al forte peso che l'informazione sui lemmi porta al modello. Ci sembra dunque significativa la coincidenza delle due classifiche sul gruppo che contiene le features morfologiche e sul gruppo *ner*. Per quel che riguarda poi l'informazione sintattica, le features che codificano informazioni lessicali risultano meno significative di quelle che codificano informazione legata al contesto (es. preposizioni o tipo di testa).



(a) Valori di accuracy ottenuti eliminando progressivamente gruppi di feature dalla versione completa comprendente i lemmi



(b) Valori di accuracy ottenuti eliminando progressivamente gruppi di feature dalla versione completa *non* comprendente i lemmi

Figura 6.15: Recursive Feature Elimination

6.5 Valutazione del modello

Il modello creato con l'assetto di features che ha registrato la miglior performance nella fase di selezione dei modelli è stato valutato sulla porzione di test del corpus. Lo stesso è stato fatto per il modello *base* e il modello *unipi*.

I risultati, riportati in Tabella 6.4, se confrontati con i valori di f-measure valutati con la 10-fold cross validation sul train, rivelano una maggiore stabilità dell'informazione non lessicale da noi selezionata rispetto all'insieme di features su cui il modello *unipi* si basa.

Tabella 6.4: Risultati riportati in fase di testing. Le misurazioni riportate sono in termini di f1-measure.

versione	**Average**	Abstract	Animate	Object	Location	Event	Other
base	**0.261**	0.834	0.594	0	0.047	0.092	0
completa	**0.396**	0.742	0.583	0.143	0.349	0.404	0.154
unipi	**0.553**	0.777	0.766	0.313	0.540	0.390	0.530

Capitolo 7

Conclusioni

La performance del nostro modello non risulta sicuramente confrontabile con nessuno dei modelli citati nel capitolo 3.4. Le differenze tra i modelli presi in considerazione e il nostro si pongono infatti su più livelli.

La differente granularità dei tagset (insieme con la restrizione del task all'ambito dei sostantivi), oltre a non permettere un confronto diretto tra le classi, porta con sé effetti sulla valutazione dell'accuratezza di difficile previsione. Differenze nella qualità delle risorse utilizzate, a partire dall'annotazione del corpus, influiscono inoltre non solo sulla performance finale, ma sull'intero processo di estrazione e selezione delle features.

Gli algoritmi utilizzati sono differenti e, nel nostro caso, l'attenzione posta alla scelta dell'assetto dell'algoritmo è stata minima, riservandoci di esplorare maggiormente in un secondo momento l'influenza dei parametri numerici sulla performance.

L'ontologia di riferimento è stata sviluppata per lo specifico task e non le si possono attribuire le stesse capacità di sistematizzazione che si assumono per le classi lessicografiche di WordNet. Lo dimostra il fatto che le classi risultanti siano fortemente sbilanciate, parametro da tenere in considerazione nel momento in cui ci proponiamo di trovare una plausibile rappresentazione dei dati, che sia in grado di rendere evidenti quelle che anche per l'annotatore umano risultano spesso sottili differenze.

Confrontandoci con quanto riportato in Agirre e Stevenson 2007, i risultati emersi dalle valutazioni dei modelli e della rilevanza delle features sembrano in linea con le argomentazioni degli autori. Informazioni su Part of Speech e Morfologia, insieme a pattern di sottocategorizzazione anche identificati semplicemente in n-grammi di parti del discorso, sono indicate come le più rilevanti, e così si confermano nel nostro modello. Per quel che riguarda l'informazione sintattica e semantica, come collocazioni o associazioni semantiche, queste sembrano riuscire a fornire un supporto informativo solo se provenienti da risorse che godono di un alto livello di accuratezza. Nell'analisi svolta in Stevenson e Wilks 2001 si mostra inoltre come tali informazioni siano estremamente rilevanti per la disambiguazione di verbi, ma molto meno influenti nel caso della disambiguazione di sostantivi.

La scarsa accuratezza dei nostri dati, e la loro relativa sparsità (consideriamo che il parsing sintattico effettuato su alcune porzioni del linguaggio giornalistico, ad esempio titoli o frasi molto brevi le cui dipendenze sono in realtà da recuperare in una finestra di testo più ampia, non rispecchia accuratamente la struttura sintattica del discorso, e a volte è del tutto manchevole), potrebbero quindi essere sufficienti a spiegare la scarsa influenza che le features che codificano tale informazione hanno avuto nel panorama generale dei modelli da noi sviluppati.

Un'ultima considerazione è da farsi sull'interazione tra le fonti di informazione e l'algoritmo utilizzato: come già discusso nel capitolo 5.3, restringendosi al campo del singolo documento il livello di ambiguità cala notevolmente. Da questo punto di vista è ragionevole pensare che, nel caso di multiple occorrenze di uno stesso lemma all'interno di un documento, l'informazione estratta per ogni token possa essere utile alla disambiguazione di tutti i contesti. Inoltre, dal confronto tra gli annotatori, è emerso che è piuttosto rilevante considerare la sequenza di tag assegnati, al di là della classificazione del singolo token. Un algoritmo come la SVM, che codifica l'informazione separatamente per ogni istanza e, nell'implementazione da noi utilizzata, non fa riferimento alla sequenza di tag assegnati, non è in grado di approssimare questo tipo di informazione.

7.1 Sviluppi futuri

Molti sono gli spunti che emergono per la prosecuzione del lavoro, che risulta sicuramente, in questo stadio, incompleto tanto dal punto di vista teorico quanto da quello computazionale.

Alcune riflessioni di carattere tecnico sorgono dall'esperienza fatta con l'uso delle risorse: al di là della già citata sperimentazione di più assetti per l'algoritmo utilizzato, molta attenzione può essere ancora posta ad alcuni elementi come l'uso di filtri o di una funzione di smoothing sulle frequenze degli elementi considerati (a partire dagli stessi lemmi o dalle coppie lemma-PoS).

Dal punto di vista algoritmico, sarebbe interessante riprodurre le dicotomie presenti nell'ontologia, dividendo dunque inizialmente le entità da ciò che riteniamo

non semanticamente classificabile, e poi le varie tipologie all'interno delle entità, ottenendo così la classificazione di un'istanza da una composizione di classificazioni, eventualmente ottenute con algoritmi diversi.

Un'analisi più approfondita degli errori commessi in fase di classificazione porterebbe senza dubbio a una maggiore consapevolezza sul funzionamento del sistema e ci permetterebbe di comprenderne meglio le debolezze e i punti di forza. Per quel che riguarda l'informazione fornita al sistema, altre strade restano comunque percorribili. Soluzioni banali come l'aggiunta di un dizionario di costrutti tipici potrebbero ad esempio migliorare sensibilmente il riconoscimento dei costrutti funzionali o altre strutture fisse. In Finlayson e Kulkarni 2011 si mostra inoltre come il riconoscimento preliminare di MultiWord Expressions migliori la disambiguazione: nel corpus da noi considerato le MWE sono state riconosciute in fase di annotazione, ma questa informazione non è stata inclusa in alcun modo nei modelli presentati. Come già accennato, in Agirre e Stevenson 2007 si suggeriscono altre rilevanti fonti di informazione, citiamo tra le altre esempio l'uso di informazione globale riguardante il dominio o di pattern argomentali, e ciò suggerisce come molto lavoro di *feature engineering* possa ancora essere fatto. Un altro ambito non toccato da questo lavoro ma trattato in letteratura[1] è l'introduzione di informazione tratta da rappresentazioni delle parole create in modo non supervisionato, ad esempio tramite *word embeddings* o *cluster*. A sostegno di ciò notiamo come dalla valutazione del nostro modello sia emerso che l'informazione lessicale sia estremamente rilevante ai fini del task, ma, nella forma in cui è stata inclusa nei modelli qui esaminati, troppo soggetta al fenomeno della sparsità dei dati.

[1] si veda ad esempio Turian et al. 2010

Appendice A

Descrizione dei tag utilizzati a EVALITA 2011

0 - adj.all This tag is used for all simple adjectives, such as grande, bello, simpatico.

1 - adj.pert This tag is used for all adjectives that are related with nouns, such as scolastico, marittimo, soleggiato.

2 - adv.all This tag is used for all adverbs, such as anche, sempre, dove.

3 - noun.Tops This tag is used for those nouns that appear as super sense, such as animale, gruppo, tempo.

4 - noun.act This tag is used for all those nouns that denote an action, such as corsa, incontro, sciopero.

5 - noun.animal This tag is used for all nouns of animals, such as cane, gorilla, coniglio .

6 - noun.artifact This tag is used for all man - made objects, such as edificio, fontana, bomba .

7 - noun.attribute This tag is used for all nouns that denote attributes of people, such as serietà, eleganza, pigrizia.

8 - noun.body This tag is used for all body parts, such as braccio, occhio, cuore.

9 - noun.cognition This tag is used to identify all nouns related to cognitive (or mental) processes, such as pensiero, sogno, conoscenza.

10 - noun.communication This tag is used for all nouns that denote both objects that allow communication, such as libro, film, licenza, and nouns that denote comunicative processes, such as discussione, chiarimento, proposta.

11 - noun.event This tag is used to denote all nouns of event, such as trionfo or incidente.

12 - noun.feeling This tag is used to identify all emotions and feelings, such as delusione, paura, desiderio.

13 - noun.food This tag is used to denote both all nouns of food, such as miele, aranciata, pizza, and the meals in which they are consumed, such as cena or merenda.

14 - noun.group This tag is used to denote all the nouns that refer to associations or organization, groups or communities, such as church, ONU, Mediaset. This tag is also used to denote football team, such as Italia, Francia, Germania.

15 - noun.location This tag is used to denote nouns of cities or places, such as Pisa, Roma or via, piazza.

16 - noun.motive This tag is used to denote all those nouns that refer to a purpose, such as ragione, causa, motivo.

17 - noun.object Thi s tag is used to denote all natural objects, such as pietra, mare, montagna , but just if they are used as objects. For instance, in this sentence Abbiamo scalato la montagna più alta del mondo montagna is annoted as noun.object, but on the other hand, in this sentence Sono andato in montagna montagna is annoted as noun.location.

18 - noun.person This tag is used to denote first and last name of persons, such as Giovanni or Rossi.

19 - noun.phenomenon This tag is used for nouns that denote natural phenomenons, such as nebbia, fulmine, perturbazione.

20 - noun.plant This tag is used for all nouns of plants, such as pino, polline, basilico. Is also used for the nouns of vegetables, but not used in context of eating.

21 - noun.possession This tag is used for all nouns of pessession, such as finanziamento, tassa, and also for nouns of quantity of cash, such as miliardi.

22 - noun.process This tag is used to denote all nouns that express the gowing up of a process, such as declino, sviluppo, and also natural processes, such as tramonto.

23 - noun.quantity This tag is used for all nouns that denote a quantity or units, such as numbers or metri and dollari.

24 - noun.relation This tag is used to all nouns that refer to a part of something, such as per cento or parte, but also est or ovest , because they refer to a single part of something (for instance, a eastern part fo the world).

25 - noun.shape This tag is used to all all nouns that denote a objects that have a particolar shape, such as colonna, piano, curva.

26 - noun.state This tag is used to all nouns that denote a state of persons or situations, such as morte, crisi, pace.

27 - noun.substance This tag is used to all nouns that denote a substance, such as oro, gas, pasta.

28 - noun.time This tag is used to all nouns that express time, such as notte, settembre, ore .

29 - verb.body This tag is used to all verbs that express actions of body, such as dormire, respirare.

30 - verb.change This tag is used to all verbs that express a change of something, such as accendere, chiudere.

31 - verb.cognition This tag is used to all verbs that express actions that involve the mind, such as immaginare, dubitare, sperare.

32 - verb.communication This tag is used to all communication verbs, such as parlare, cantare, leggere.

33 - verb.competition This tag is used to all those verbs of both sports competition and hostility, such as vincere, gareggiare or espugnare, sparare.

34 - verb.consumption This tag is used to all verb that express action of eating, drinking or, more generally, consumption of something, such as mangiare, sorseggiare or fumare.

35 - verb.contact This tag is used for all verbs that denote contact, such as avvolgere, sfiorare.

36 - verb.creation This tag is used to all verbs that express action of creation or destruction ,such as costruire and distruggere , but also verbs about creative processes, such as dipingere or suonare.

37 - verb.emotion This tag is used to all verbs of emotion or feelings, such as esaltare, temere.

38 - verb.motion This tag is used for all verbs that express different type of moving, such as camminare, volare o muovere.

39 - verb.perception This tag is used to all verbs about perception, such as vedere, sentire.

40 - verb.possession This tag is used for all verbs that express Exchange of possessions, such as finanziare, pagare, investire.

41 - verb.social This tag is used for all verbs used to express social action, such as presentare, organizzare, emarginare.

42 - verb.stative This tag is used to all verbs that express a state that not change, such as rimanere, mantenere, esistere.

43 - verb.weather This tag is used to all verbs that express weather situations, such as piovere, nevicare, tuonare.

44 - adj.ppl This tag is used to denote all adjectives participials, i.e. those adjectives that have the same form of a partic ple, but they are not related with some verb, such as preoccupante.

Appendice B

Descrizione delle features

form form del token in esame

lemma lemma del token in esame

morfologia_genere un valore tra m,f,n che rappresenta il genere del token in esame (maschile, femminile, neutro)

morfologia_numero un valore tra s,p,n che rappresenta il numero del token in esame (singolare, plurale, neutro)

PoS Part of Speech Fine Grained del token in esame

FirstWordCap 1 se il token è in prima posizione e capitalizzato, 0 altrimenti

FirstWordNoCap 1 se il token è in prima posizione ma non capitalizzato, 0 altrimenti

Hyphen 1 se il token è formato da due parti congiunte da trattino, 0 altrimenti

FrequenzaLemmaPoS_log logaritmo naturale della frequenza - nel corpus di repubblica - della coppia lemma-PoS del token in esame

FrequenzaRelativaLemmaPoS 1 se il rapporto tra la frequenza - nel corpus di repubblica - della coppia lemma-PoS e la frequenza del lemma supera una soglia s (fissata a 0.1), 0 altrimenti

Capitalized 1 se la form è capitalizzata, 0 altrimenti

ContainsDigit presenza di cifre

ContainsPunct presenza di punteggiatura

Upper 1 se il token contiene una maiuscola non in prima posizione, 0 altrimenti

WordShape 1 se la form contiene una cifra, segni di punteggiatura o una maiuscola in prima posizione, 0 altrimenti

FirstWord 1 se il token è in prima posizione nella frase, 0 altrimenti

PoSPrev PoS del token precedente

PoSNext PoS del token successivo

CPoSPrev CPoS del token precedente

CPoSNext CPoS del token successivo

SeqCap 1 se il token in esame, il precedente e il successivo sono capitalizzati, 0 altrimenti

CapNext 1 se il token in esame e il successivo sono capitalizzati, 0 altrimenti

CapPrev 1 se il token in esame e il precedente sono capitalizzati, 0 altrimenti

WithinQuotes 1 se il token si trova in una sequenza tra virgolette, 0 altrimenti

ModAdj_pre_b presenza di aggettivi come modificatore postnominale del token

ModAdj_post_b numero di aggettivi presenti come modificatore del token

ModAdj_b presenza di aggettivi come modificatore del token

ModAdj_clusters elenco di cluster relativi ai lemmi presenti come modificatori aggettivali del token

ModNum 1 se il token è modificato da un numerale, 0 altrimenti

Det_def 1 se il token è modificato da un determinate definito, 0 altrimenti

Det_indef 1 se il token è modificato da un determinate indefinito, 0 altrimenti

DipInv_tipo presenza della dipendenza per tipo di dipendenza

DipInv_lemmi Lemma della testa del token

DipInv_PoS Pos della testa del token

DipInv_preposizione preposizione che lega il token alla sua testa

DipInv_forzaassociazioni Log-likelihood tra il token e la sua testa

DipInv_classeassociazioni Probabilità che la testa del token selezioni come dipendente un token di ciascuna classe

PresenzaDip 1 se il token presenta una dipendenza, 0 altrimenti

Dip_b presenza di dipendenza per tipo di dipendenza

Dip_PoS PoS dei token presenti in dipendenze con il token in esame

Dip_preposizione Preposizioni che leganao il token con le sue dipendenze

Dip_lemmi lemmi presenti in dipendenza con il token in esame

Bibliografia

Abney, Steven P (1991). *Parsing by chunks*. Springer.

Agirre, Eneko e David Martinez (2001). «Knowledge sources for word sense disambiguation». In: *International Conference on Text, Speech and Dialogue*. Springer, pp. 1–10.

Agirre, Eneko e Mark Stevenson (2007). «8 Knowledge Sources for WSD». In: *Word Sense Disambiguation*, p. 217.

Attardi, G et al. (2008). *Tanl (Text Analytics and Natural Language processing). Project Analisi di Testi per il Semantic Web e il Question Answering*.

Attardi, Giuseppe e Felice Dell'Orletta (2009). «Reverse revision and linear tree combination for dependency parsing». In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, pp. 261–264.

Attardi, Giuseppe et al. (2009). «Accurate dependency parsing with a stacked multilayer perceptron». In: *Proceedings of EVALITA* 9.

Attardi, Giuseppe et al. (2010). «A Resource and Tool for Super-sense Tagging of Italian Texts.» In: *LREC*.

Attardi, Giuseppe et al. (2013). «SuperSense Tagging with a Maximum Entropy Markov Model». In: *Evaluation of Natural Language and Speech Tools for Italian*. Springer, pp. 186–194.

Basile, Pierpaolo (2013). «Super-Sense Tagging using support vector machines and distributional features». In: *Evaluation of Natural Language and Speech Tools for Italian*. Springer, pp. 176–185.

Bennett, Edward M, R Alpert e AC Goldstein (1954). «Communications through limited-response questioning». In: *Public Opinion Quarterly* 18.3, pp. 303–308.

Bentivogli, Luisa, Pamela Forner e Emanuele Pianta (2004). «Evaluating cross-language annotation transfer in the multisemcor corpus». In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, p. 364.

Charniak, Eugene et al. (2000). «Bllip 1987-89 wsj corpus release 1». In: *Linguistic Data Consortium, Philadelphia* 36.

Ciaramita, Massimiliano e Yasemin Altun (2006). «Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger». In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 594–602.

Ciaramita, Massimiliano e Mark Johnson (2003). «Supersense tagging of unknown nouns in WordNet». In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pp. 168–175.

Collins, Michael (2002). «Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms». In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 1–8.

Cortes, Corinna e Vladimir Vapnik (1995). «Support-vector networks». In: *Machine learning* 20.3, pp. 273–297.

Crammer, Koby e Yoram Singer (2003). «Ultraconservative online algorithms for multiclass problems». In: *The Journal of Machine Learning Research* 3, pp. 951–991.

Curran, James R (2005). «Supersense tagging of unknown nouns using semantic similarity». In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 26–33.

DellOrletta, Felice (2009). «Ensemble system for Part-of-Speech tagging». In: *Proceedings of EVALITA* 9.

Dunning, Ted (1993). «Accurate methods for the statistics of surprise and coincidence». In: *Computational linguistics* 19.1, pp. 61–74.

Fan, Rong-En et al. (2008). «LIBLINEAR: A Library for Large Linear Classification». In: *Journal of Machine Learning Research* 9, pp. 1871–1874.

Fellbaum, Christiane (1998). *WordNet*. Wiley Online Library.

Finlayson, Mark Alan e Nidhi Kulkarni (2011). «Detecting multi-word expressions improves word sense disambiguation». In: *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*. Association for Computational Linguistics, pp. 20–24.

Fleiss, Joseph L (1971). «Measuring nominal scale agreement among many raters.» In: *Psychological bulletin* 76.5, p. 378.

Harris, Zellig S (1970). *Distributional structure*. Springer.

Hirst, Graeme (1992). *Semantic interpretation and the resolution of ambiguity*. Cambridge University Press.

Hsu, Louis M e Ronald Field (2003). «Interrater agreement measures: Comments on Kappan, Cohen's Kappa, Scott's π, and Aickin's α». In: *Understanding Statistics* 2.3, pp. 205–219.

Kohen, Jacob (1960). «A coefficient of agreement for nominal scale». In: *Educ Psychol Meas* 20, pp. 37–46.

Kuera, Henry, Winthrop Nelson Francis et al. (1967). «Computational Analysis of Present-Day American English». In:

Landis, J Richard e Gary G Koch (1977). «The measurement of observer agreement for categorical data». In: *biometrics*, pp. 159–174.

Lenci, Alessandro, Simonetta Montemagni e Vito Pirrelli (2005). *Testo e computer: elementi di linguistica computazionale*. Carocci.

Lyons, John (1977). «Semantics (vols I & II)». In: *Cambridge CUP*.

Martin, James H e Daniel Jurafsky (2000). «Speech and language processing». In: *International Edition* 710.

McCroy, SW (1992). «Using multiple knowledge sources for word sense disambiguation». In: *Computational Linguistics* 18, pp. 1–30.

Miller, George A (1998). «Nouns in wordnet». In: *WordNet: An electronic lexical database*, pp. 24–45.

Miller, George A et al. (1993). «A semantic concordance». In: *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, pp. 303–308.

Mitchell, Thomas M (1997). «Machine learning». In: *Boston et al.*

Montemagni, Simonetta et al. (2003). «Building the Italian syntactic-semantic treebank». In: *Treebanks*. Springer, pp. 189–210.

Murphy, M Lynne (2010). *Lexical meaning*. Cambridge University Press.

Navigli, Roberto (2009). «Word sense disambiguation: A survey». In: *ACM Computing Surveys (CSUR)* 41.2, p. 10.

Pedregosa, F. et al. (2011). «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Picca, Davide, Alfio Massimiliano Gliozzo e Massimiliano Ciaramita (2008). «Supersense Tagger for Italian.» In: *LREC*. Citeseer.

Ross, Quillian (1968). *Semantic memory, Semantic information processing.*

Scott, William A (1955). «Reliability of content analysis: The case of nominal scale coding». In: *Public opinion quarterly* 19.3, p. 321.

Snyder, Benjamin e Martha Palmer (2004). «The English all-words task». In: *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43.

Stevenson, Mark e Yorick Wilks (2001). «The interaction of knowledge sources in word sense disambiguation». In: *Computational Linguistics* 27.3, pp. 321–349.

Turian, Joseph, Lev Ratinov e Yoshua Bengio (2010). «Word representations: a simple and general method for semi-supervised learning». In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pp. 384–394.

Turing, Alan M (1950). «Computing machinery and intelligence». In: *Mind* 59.236, pp. 433–460.

Vapnik, Vladimir Naumovich e Vlamimir Vapnik (1998). *Statistical learning theory.* Vol. 1. Wiley New York.

Woods, William A (1975). «Whats in a link: Foundations for semantic networks». In: *Representation and understanding: Studies in cognitive science*, pp. 35–82.

Zwick, Rebecca (1988). «Another look at interrater agreement.» In: *Psychological bulletin* 103.3, p. 374.