L'ADATTAMENTO DI UN PARSER DI ITALIANO L1: PROBLEMI E PROSPETTIVE

Sadegh Astaneh* e Francesca Frontini**

1. Introduzione

Il lavoro qui presentato costituisce la parte applicativa del progetto PIL2 condotto presso il Dipartimento di Linguistica dell'Università di Pavia (la cui parte teorica è introdotta da Andorno e Rastelli in questo volume). Verrà illustrata una procedura per l'annotazione e l'interrogazione di corpora L2, effettuata in modo semi-automatico unendo le potenzialità del parser Italian NLP (realizzato da ILC-CNR) ad una annotazione manuale su più livelli.

2. L'input

L'input è costituito da un testo in italiano L2, sia esso la trascrizione di parlato, sia esso l'originale di una produzione scritta. Nel primo caso il testo potrà contenere un'annotazione specifica per la trascrizione del parlato. Attualmente il sistema PIL2 è stato applicato a dati provenienti da due corpora:

- il **Corpus Chini** è una raccolta di trascrizioni di registrazioni effettuate presso l'Università di Pavia intorno al 1995/1996; gli informanti, tutti studenti universitari in soggiorno di studio a Pavia, sono di varie L1 (tedesco, spagnolo, cinese) e le conversazioni hanno temi diversi (Frog Story, Portafogli, Modern Times, Conversazione Libera); la trascrizione delle registrazioni è stata fatta utilizzando

*Università Statale di Milano
**Università degli Studi di Pavia - Dipartimento di Linguistica Teorica e Applicata

il formato chat (per informazioni più dettagliate su questo sistema di annotazione si veda MacWhinney, 2000).

– **ISA** è un corpus di apprendimento formato da testi scritti dagli studenti americani che frequentano un programma universitario americano con sede a Milano (I.E.S.). Ogni testo è composto da 80-120 parole. Il compito richiesto agli studenti è descrivere una singola scena del film "Pane e tulipani" (Silvio Soldini, Italia, 2000). Il corpus è stato raccolto tra il 2003 e il 2005 e contiene circa 600 file per un totale di 70 mila occorrenze. La pulizia dei dati - per quanto concerne le condizioni esterne dell'esperimento - è garantita dal fatto che la prova avviene in aula sotto il controllo del docente, senza vocabolario, senza proofreading. Gli studenti sono per l'80% anglofoni (per una descrizione dettagiata si veda Rastelli, 2005; 2007).

Il sistema di trattamento e annotazione di dati L2 è stato sviluppato a partire da una porzione del Corpus Chini, (tedesconfoni che raccontano Frog Story) e successivamente applicato anche al Copus ISA. Qui viene riportato uno frammento di trascrizione dal Corpus Chini:

(1) *IT0: oggi è il 13 dicembre 1995.
*IT0: il signor A ha guardato attentamente?
*IT0: ecco il signor A ha visto i fogli della frog story -, io non li vedo-, lei vuole raccontarmi -, guardandoli -, raccontare la storia guardandoli o così.
*ANT: no senza guardare [% volume basso].
*IT0: senza guardare -, comunque se ogni tanto: le viene qualche dubbio li può guardare -, ecco # okay.
*ANT: questa storia si tratta di un -, un ragazzo -' che possessa due -, un -, due -, due animali -, un -, due animali -, un un un cane e un -, un -, una rana.
*IT0: sì.
*ANT: il ragazzo è molto felice con i due -, i su -, i suoi animali e # ehm ## ad un ## lui ha il rane in un -, una caraffa di vetro -' per evitare di -, sfugge -, di -, di essere sfuggiato il -, la rana -' e ad un -, un giorno il ragazzo si sveglia e deve deve # vèdere che la rane è sfugato.

3. Trattamento del parlato

Per poter processare ed etichettare un trascrizione di parlato L2 (come i testi presenti nel Corpus Chini) sono necessarie alcune operazioni preliminari:

1 distinguere le stringhe di parlanti nativi da quelle di parlanti non nativi
2 per il parlato, distinguere i segni di annotazione dal testo vero e proprio

Inoltre è necessaria una decisione teorica sul trattamento di tutti quei fenomeni, tipici del parlato, ma presenti in maniera enfatizzata nel parlato L2, quali ripetizioni, riformulazioni, frasi sospese. A questo proposito si è scelto di escludere dal parsing i seguenti elementi (si veda Andorno & Rastelli in questo volume):

- punteggiatura, tranne quella eventualmente utile al parser (alcuni punti o virgole per separare elementi che potrebbero essere altrimenti interpretati come chunk);
- parole in lingua straniera non integrate in chunk con parole in IL (code switching):

(2) lui dice che # brot vuole il pane — brot ['salta']

- ripetizioni di parole o gruppi di parola o loro frammenti immediatamente seguiti da ripetizione / integrazione:

(3) a. ripetizioni e integrazioni esatte
- il cara -, caravan — cara ['salta']
- di di di un amico — di ['salta'] di ['salta']

b. ripetizioni e integrazioni con cambio di forma:
- di una un amico — una ['salta']
- mentre dorm sta dormendo — dorm ['salta']

c. ripetizioni e integrazioni di sequenze di parole:
- della mia della mia amica — della ['salta'] mia ['salta']
- perché duran perché durante le feste — perché ['salta'] duran ['salta']

d. ripetizioni e integrazioni di sequenze di parole con cambio di forma:
- del mio della mia amica — del ['salta'] mio ['salta']
- perché det perché sta dicendo — perché ['salta'] det ['salta']

- frammenti e forme devianti non riconducibili a una sicura forma di TL

(4)
- ha s visto che c'era la rana — s ['salta']
- lui arriv pensa di andare — arriv ['salta']

Concretamente, è stato sviluppato algoritmo che permette di effettuare al tempo stesso sia la tokenizzazione del testo che, per le trascrizioni in formato chat, una prima separazione tra contenuto linguistico e segni di annotazione. Il risultato è un file in formato tabellare, che contiene un token per riga. Nella prima colonna viene riportato il testo originale, mentre nella seconda colonna vengono ripetuti automaticamente soltanto i token linguistici. L'annotatore manuale provvederà ad eliminare dalla seconda colonna anche i turni del parlante, la punteggiatura superflua, le ripetizioni e i frammenti secondo le regole sopra riportate (Tab. 1 colonna 1 e 2). A questo punto il testo è pronto per le fasi di normalizzazione e annotazione vere e proprie; nel caso di input di partenza costituito da dati di italiano L2 scritto, privo di segnali di trascrizione, l'unica operazione a questo livello è quella di tokenizzazione.

4. Annotazione manuale

La fase di annotazione manuale vera e propria viene svolta utilizzando le altre colonne della tabella 1.

- colonna "Cambia": contiene l'eventuale **forma tendenziale**. Per ogni forma deviante o anomala in L2 viene individuata la forma L1 più vicina (per quanto riguarda il concetto di forma target, si rimanda ad Andorno - Rastelli in questo volume). Tale operazione naturalmente non cancella il riferimento alla forma originale (source), come vedremo illustrando i criteri di annotazione, ma consente di sfruttare al meglio le potenzialità di analisi dei dati trattati da parte di un parser Italiano L1.
- colonne "Ins prima" e "Ins dopo": contengono eventuali token aggiunti, che servono per favorire la successiva corretta analisi sintattica dei dati.
- colonna "Conv": contiene un'eventuale annotazione conversazionale (false partenze, ripetizioni, ...)
- colonna "T_categoria" e "T_chunk": consente di effettuare l'analisi morfologica e sintattica manualmente (bypassando l'esito del parser).
- colonna "T_flag_verbo": contiene gli specifici flag verbali ("intr_av" - verbo intransitivo con avere; "intr_es" - verbo intransitivo con essere; "tr" - verbo transitivo; "imp" - impersonale; "pass" - passivo; "due_aux" - doppio ausiliare; "exist" - esistenziale; "mod" - modale; "lvc" - light verb construction; "pron" - pronominale)

Token	Leggi	Cambia	Ins prima	Ins dopo	T_conv	T_categoria	T_chunk	T_flag_verbo
ANT								
:								
questa	questa							
storia	storia							
si	si							
tratta	tratta							pron, intr_es
di	di							
un					fs, rip_e			
-								
,								
un	un							
ragazzo	ragazzo							
-								
che	che							
possessa		possiede						tr

due					fs, rip_e			
-								
,								
un					fs, rip_e			
-								
,								
due					fs, rip_e			
-								
,								
due					fs, rip_e			
animali					fs, rip_e			
-								
,								
un					fs			
-								
,								
due		due						
animali		animali						
-								
,								
un						fs		

Tabella 1 - Tokenizzazione e annotazione manuale

5. Parsing

Italian NLP è un analizzatore morfosintattico realizzato da ILC-CNR di Pisa. Il parser è stato sviluppato per essere applicato a testi di italiano L1 e ottiene performance molto elevate con la lingua scritta. L'algoritmo su cui si basa il parser prevede che il testo opportunamente tokenizzato sia inviato ad un analizzatore morfologico, che collegato ad un dizionario macchina, è in grado di riconoscere forma possibile e relativo lemma per ogni token. Nel caso di forme ambigue, l'analizzatore fornisce le varie alternative; la disambiguazione avviene successivamente, in sede di analisi sintattica. Il parser infatti utilizza in ingresso i dati dell'analisi morfologica e, basandosi su un set di regole (automi a stati finiti) è in grado di ricostruire delle unità sintattiche minime, attribuendo al tempo stesso i token in ingresso ad una specifica categoria lessicale.

In Federici *et al.* (1996) Italian NLP viene descritto come un parser leggero, che permette di segmentare il testo in una sequenza non strutturata di unità testuali

organizzate sintatticamente, chiamate **chunk**. Un chunk viene definito come unità testuale di token adiacenti.

Rispetto alla tradizionale accezione di sintagma utilizzata nelle grammatiche formali, nel text chunking:

- sono identificate solo le relazioni non ambigue
- le relazioni sono solo tra token adiacenti: non sono permessi chunk discontinui
- i chunk non sono ricorsivi
- eventuali relazioni sintattiche tra i diversi chunk non vengono specificate
- sono possibili chunk non identificati

Si consideri il seguente esempio di chunking (i chunk sono individuati dalle parentesi):

(6) (Gianni) (ha mangiato) (la minestra) (calda).

Tale frase può avere due interpretazioni ('John ate the hot soup' o 'John ate the soup hot'), quindi il token "calda" costituisce un chunk a sé stante. Inoltre i vari chunk sono tutti allo stesso livello, senza gli incassamenti tipici dell'analisi sintattica ricorsiva, e non vengono specificate le relazioni tra le unità di livello superiore, come quella che intercorre tra "Gianni" e "ha mangiato", o tra "ha mangiato" e "la minestra".

Le riflessioni teoriche sviluppate nell'ambito del progetto PIL2 (si vedano Andorno Rastelli in questo volume) hanno portato a pensare che un un'analisi sintattica ed un parsing "leggeri" siano più adeguati al trattamento della L2. In molti casi la natura dell'interlingua è tale che non è opportuno spingersi troppo oltre nell'inferire legami di reggenza e relazioni tra parole non prossime le une alle altre.

Il parser è stato sviluppato per essere utilizzato su L1 e per una varietà scritta, tuttavia la normalizzazione preventiva permette di ottenere risultati target-like anche per trascrizioni di interviste in L2 (si veda il prossimo par. 6). Per ogni token, l'output dell'analizzatore riporta informazioni su lemma, PoS, analisi morfologica, e appartenenza o meno ad un chunk. Qui di seguito vediamo un frammento di testo L2 normalizzato, analizzato con Italian NLP

(7) {questa storia si tratta di un ragazzo che possiede due animali un cane e una rana . }

CHUNKING				**TOKEN**	**ANALISI MORFOLOGICA**
B-	N_C	@FS	DET	questa	QUESTO#D@FS
I-	N_C	@FS	POTGOV	storia	STORIA#S@FS
B-	N_C	@NN3	POTGOV	si	SI#PQ@NN3
B-	N_C	@FS	POTGOV	tratta	TRATTA#S@FS
B-	P_C	@MS	PREP	di	DI#E
I-	P_C	@MS	DET	un	UN#RI@MS
I-	P_C	@MS	POTGOV	ragazzo	RAGAZZO#S@MS
B-	CHE_C		POTGOV	che	CHE#P@NN CHE#CS

					CHE#D@NN CHE#CC
B-	FV_C	@S3	POTGOV	possiede	POSSEDERE#V@S3IP
B-	N_C	@MP	PREMODIF	due	DUE#N@NN
I-	N_C	@MP	POTGOV	animali	ANIMALE#S@MP
B-	NA_C	@MS	DET	un	UN#RI@MS
I-	NA_C	@MS	POTGOV	cane	CANE#S@MS
B-	COORD_C		CONJTYPE	e	E#CC
B-	N_C	@FS	DET	una	UNA#RI@FS
I-	N_C	@FS	POTGOV	rana	RANA#S@FS
B-	PUNC_C		PUNCTYPE	.	.#@

Ogni riga corrisponde a un token (quinta colonna). Nelle prime quattro colonne si trovano le indicazioni di chunking; B-/I- segnalano che il token apre un chunk (begin) o prosegue un chunk iniziato (intermediate); le altre colonne indicano il tipo di chunk e le relazioni di accordo e reggenza. L'analisi morfosintattica si trova invece nell'ultima colonna, che è così interpretabile:

(8)	lemma	PoS	analisi morfologica
	RAGAZZO	#S	@MS (=maschile singolare)
	POSSEDERE	#V	@S3IP (=sing. 3a p. indicativo presente)

Vi sono alcuni casi nei quali il parser non è in grado di disambiguare il token, dal momento che ciò richiederebbe una analisi sintattica delle dipendenze ad *ampio raggio*, che contrasta con la filosofia del parsing leggero e delle reggenze di prossimità. La forma "che", ad esempio, non è quasi mai inseribile in un chunk composto da un altro elemento ad essa prossimo, a destra o a sinistra, che con essa sia in qualche relazione sintattica e che permetta di decidere se essa abbia valore di pronome, determinatore o congiunzione. La forma viene dunque trattata come un chunk a sé stante (CHE_C), e analizzata in tutte e tre le sue possibili varianti.

(8)				
B-	CHE_C	che	CHE#P@NN CHE#CS CHE#D@NN CHE#CC

Infine il parser può fallire nell'analisi di un token. Il fallimento può avvenire già a livello di analisi morfologica, come per le forme non target like, ma anche a livello sintattico. Ricordiamo che Italian NLP effettua la disambiguazione delle forme a livello di chunking, quindi una forma che non può essere inserita in un chunk non viene riconosciuta. Ecco ad esempio l'analisi di una sequenza L2 non normalizzata:

(9) {questa storia si tratta di un , un ragazzo che possessa due , un due , due animali , ... }

B-	N_C	@FS	DET	questa	QUESTO#D@FS
I-	N_C	@FS	POTGOV	storia	STORIA#S@FS
B-	N_C	@NN3	POTGOV	si	SI#PQ@NN3
B-	N_C	@FS	POTGOV	tratta	TRATTA#S@FS

B-	**U_C**	FORM	di	
B-	**U_C**	FORM	un	
B-	PUNC_C	PUNCTYPE	,	,#@
B-	N_C @MS	DET	un	UN#RI@MS
I-	N_C @MS	POTGOV	ragazzo	RAGAZZO#S@MS
B-	CHE_C	POTGOV	che	CHE#P@NN CHE#CS CHE#D@NN CHE#CC
B-	**U_C**	FORM	possessa	
B-	N_C @NN	POTGOV	due	DUE#S@NN
B-	PUNC_C	PUNCTYPE	,	,#@
B-	N_C @MS	DET	un	UN#RI@MS
I-	N_C @MS	POTGOV	due	DUE#S@NN
B-	PUNC_C	PUNCTYPE	,	,#@
B-	N_C @MP	PREMODIF	due	DUE#N@NN
I-	N_C @MP	POTGOV	animali	ANIMALE#S@MP

...

Come si vede sono etichettati come U_C non solo i token devianti come "possessa", ma anche token come "di" e "un", che sono privi di testa sintattica (potential governor, o POTGOV) in quanto ripetizioni o false partenze.

6. Parsing di un testo L2

Prima di illustrare la procedura di annotazione del testo, è opportuno porsi qualche domanda sull'esito della procedura di parsing di un testo di parlato L2 con uno strumento tarato su testi nativi scritti.

È in effetti immaginabile che l'analisi di un testo L2 ponga diversi problemi ad un analizzatore, in particolare del tipo di quello utilizzato, che si basa su regole e non su procedure probabilistiche.

Data la mole del corpus analizzato, sarebbe stato impensabile verificare manualmente la correttezza dell'analisi token per token. D'altra parte la variabilità dei livelli di interlingua all'interno del corpus, che con ogni probabilità influenza anche le performance del software, non rende molto attendibili dei test a campione. Vi sono però altri modi per avere un'idea dell'effetto L2 sul parsing.

Nella tabella (2) e (3) possiamo vedere il numero di fallimenti (FAIL) del parser, ovvero il numero di token ai quali non è stata assegnata una categoria. Dal momento che la disambiguazione dell'analisi morfologica avviene a livello di chunking, il numero di FAIL corrisponde al numero di U_C, (unknown chunks) attribuiti a livello sintattico. Nella tabella (2) sono riassunti i risultati per il testo originale, ripulito solo dei segni di trascrizione chat, ma non dalle ripetizioni ed esitazioni, e senza alcuna normalizzazione delle forme non target. Si confronti l'incidenza di FAIL di questa tabella con quelle della tabella (3) nella quale si trovano i dati riguardanti il testo ripulito (CLEAN), come si può notare dal numero minore di token, sia non normalizzato (NON NORM) che normalizzato (NORM). L'incidenza di token non analizzati passa,

dal grado 0 di normalizzazione al grado massimo di normalizzazione, dall'11% al 3% all'1,4%.

Osservando la tabella (4) è possibile fare un confronto con le prestazioni di Italian NLP testato su trascrizioni di testi di italiano L1 parlato. Sono stati utilizzati i dati del LIP (De Mauro *et al.*, 1993), in particolare i testi di tipologia A (scambio comunicativo bidirezionale con presa di parola libera faccia a faccia: conversazioni in casa; conversazioni sul luogo di lavoro; conversazioni nell'ambito scolastico e universitario; conversazioni in luoghi ricreativi e sui mezzi di trasporto). Dai file sono stati eliminati soltanto i segni di trascrizione. In questo senso i dati sono quindi confrontabili con quelli in tabella (2) ma, come si vede, la percentuale di token non riconosciuti è solamente del 4,6%. Si tenga presente che anche in questo caso ci troviamo di fronte ad un testo parlato, che crea maggiori problemi ad un analizzatore morfosintattico come Italian NLP rispetto alla lingua scritta.

Tabella 2 - Numero e percentuale di fail nei testi di non normalizzati e non ripuliti

	NON NORM REDUNDANT TEXT		
TITOLO	num. TOKEN	FAIL	INCIDENZA
ANT	845	96	0,113609467
NAT	491	44	0,089613035
ALE	1144	97	0,084790210
CHR	772	155	0,200777202
CLA	486	23	0,047325103
COR	583	29	0,049742710
FRA	1247	181	0,145148356
FRI	272	31	0,113970588
GIS	391	62	0,158567775
HER	984	141	0,143292683
KAR	577	99	0,171577123
NEW	980	30	0,030612245
SEL	904	74	0,081858407
SIM	541	58	0,107208872
SUS	641	21	0,032761310
WIL	381	20	0,052493438
WOL	1043	196	0,187919463
totale	12282	1357	0,110486891

Tabella 3 - Numero e percentuale di fail nei testi ripuliti, normalizzati e non normailzzati

	NON NORM CLEAN TEXT			NORM CLEAN TEXT		
TITOLO	num. TOKEN	FAIL	INCIDENZA	num. TOKEN	FAIL	INCIDENZA
ANT	619	25	0,040387722	619	9	0,014539580
NAT	445	25	0,056179775	445	19	0,042696629
ALE	914	30	0,032822757	914	12	0,013129103
CHR	557	23	0,041292639	557	7	0,012567325
CLA	437	9	0,020594966	437	3	0,006864989
COR	526	14	0,026615970	526	11	0,020912548
FRA	910	22	0,024175824	910	15	0,016483516
FRI	219	10	0,045662100	219	3	0,013698630
GIS	274	13	0,047445255	274	7	0,025547445
HER	686	30	0,043731778	686	14	0,020408163
KAR	403	15	0,037220844	403	3	0,007444169
NEW	823	7	0,008505468	823	3	0,003645200
SEL	761	15	0,019710907	761	8	0,010512484
SIM	407	15	0,036855037	407	9	0,022113022
SUS	559	2	0,003577818	559	2	0,003577818
WIL	348	9	0,025862069	348	6	0,017241379
WOL	717	33	0,046025105	717	8	0,011157601
totale	9605	297	0,030921395	9605	139	0,014471629

Tabella 4 - Corpus LIP, FAIL nell'etichettatura dei testi di tipologia A
(scambio comunicativo bidirezionale con presa di parola libera faccia a faccia)

FAIL= numero di categorie non assegnate

num. TOKEN	FAIL	INCIDENZA
108174	4986	0,046092407

Anche nella tabella (5), come nella (3) viene infine effettuato un confronto tra l'output dell'etichettatura tra testo non normalizzato e testo normalizzato. In entrambi i casi partiamo dal testo CLEAN, un input "ripulito", con lo stesso numero di token. Nella terza colonna è indicato il numero di token normalizzati; nella terza colonna viene indicato il numero di token che sono stati analizzati diversamente dal parser in seguito alla normalizzazione. Come si può vedere, il numero di normalizzazioni non è altissimo. Esso tuttavia contribuisce ad ottenere una migliore performance del parser, non solo sulla parola normalizzata, ma anche sulle parole circostanti, che vengono più facilmente disambiguate e riconosciute, come si nota dal fatto che il numero di differenze riscontrate tra testo normalizzato e non è sempre superiore al numero di normalizzazioni effettuate. La normalizzazione è dunque un intervento puntuale e non troppo invasivo, che però rende meglio interpretabile anche il testo non modificato.

TITOLO	CLEAN TEXT			
	num. TOKEN	NORMALIZZAZIONI	DIFFERENZA	INCIDENZA
ANT	619	32	44	0,0710823910
NAT	445	11	16	0,0359550562
ALE	914	28	34	0,0371991247
CHR	557	24	25	0,0448833034
CLA	437	9	11	0,0251716247
COR	526	13	16	0,0304182510
FRA	910	17	18	0,0197802198
FRI	219	14	14	0,0639269406
GIS	274	9	10	0,0364963504
HER	686	25	30	0,0437317784
KAR	403	24	27	0,0669975186
NEW	823	3	4	0,0048602673
SEL	761	7	9	0,0118265440
SIM	407	8	10	0,0245700246
SUS	559	0	0	0,0000000000
WIL	348	8	10	0,0287356322
WOL	717	51	54	0,0753138075
totale	9605	283	332	0,0345653306

Tabella 5 - Differenza tra file puliti, normalizzati e non normalizzati

7. Griglia di annotazione

L'annotazione finale emerge dalla fusione tra l'output dell'analizzatore morfosintattico e il risultato dell'etichettatura manuale. Nel progetto si è scelto di utilizzare l'annotazione in formato xml, realizzata sempre in modo automatico, a partire dall'output del parser e dalla tabella di annotazione manuale. In pratica ogni token viene etichettato con le informazioni dei vari livelli di analisi automatica (lemma, POS, morfologia e chunk) e manuale (normalizzazioni, inserimenti, informazioni conversazionali e flag verbali).

Lo schema di annotazione finale contiene i seguenti tag e attributi, in ordine gerarchico:

<file> con attributo *name*, nel quale sono registrate, in forma di codice alfanumerico, tutte le informazioni riguardanti il testo (codice dell'informante, numero di mesi trascorsi in Italia al momento della registrazione, livello di competenza linguistica in italiano, corpus di appartenenza, L1 dell'informante, argomento del testo).

<frase> che delimita i confini di frase del testo; per i testi orali le frasi sono delimitate dai punti preservati dalla trascrizione, che in genere coincidono con la fine di un turno o con particolari contesti intonativi.

<chunk> che contiene le informazioni sul tipo di unità minima morfosintattica nella quale il token è inserito

<morf> che contiene l'analisi morfo-sintattica; i suoi attributi sono *lemma, Head* (categoria lessicale del token), *gend*, *numb*, *mood*, *tense*, *case*, *grade* (analisi morfologica) e l'eventuale *flag verbale*

<token> con attributo *target* che riporta l'indicazione della forma target e di quella originaria per gli elementi normalizzati

Qui di seguito possiamo vedere un frammento di testo annotato:

(10) questa storia si tratta di un -, un ragazzo -' che possessa due -, un -, due -, due animali -, un -, due animali -, un un un cane e un -, un -, una rana.

```
<frase val= "questa storia si tratta di un ragazzo che possiede due animali un cane e una rana .">
<chunk val="N_C">
        <morf lemma="QUESTO" Head="det" gend="f" numb="s">questa</morf>
        <morf lemma="STORIA" Head="nn" gend="f" numb="s">storia</morf>
</chunk>
.....
        <morf lemma="POSSEDERE" Head="v_fin" mood="ind" tense="pres"
        pers="3" numb="s" flag="tr"> <token target="possiede">possessa</token>
        </morf>
.....
<conv rip_e="yes" fs="yes">due</conv>
<conv rip_e="yes" fs="yes">animali</conv>
.....
</frase>
```

8. Etichettatura posizionale

Uno dei problemi più spinosi da risolvere a livello di etichettatura è quello di mantenere l'**informazione posizionale** all'interno della sintassi xml. La posizione relativa di ogni token all'interno di un testo porta con sé un tipo di informazione relazionale e orizzontale, mentre gli specifici sistemi di query xml sono sviluppati soprattutto per recuperare informazione gerarchica. Al momento si è scelto di implementare l'informazione posizionale in maniera esplicita nell'annotazione xml attraverso indicizzazione. Il problema di annotare informazione posizionale è tanto più importante in un corpus L2, dal momento che l'informazione in merito alle relazioni tra le parole non può essere facilmente registrata a livello sintattico. Le relazioni sintattiche non possono essere date per scontate, e riconosciute in maniera non ambigua in una frase L2. In effetti è stato sostenuto che, soprattutto nelle Varietà Basiche, l'organizzazione degli enunciati è guidata da principi semantici e pragmatici, piuttosto che dalla reggenza sintattica (Klein & Perdue, 1997).

Qui di seguito viene mostrato un frammento di (4) al quale è stata aggiunta

l'annotazione posizionale sul nodo <morf>. Come si può vedere, ciascun singolo nodo contiene le informazioni riguardanti il proprio intorno, registrate all'interno degli attributi @p1-@p6. Dal momento che l'intorno di ogni token ha come confine invalicabile quello della frase, i token di inizio periodo hanno le posizioni di sinistra vuote e i token di fine periodo hanno le posizioni di destra vuote. Attualmente l'indicizzazione e, conseguentemente, la ricerca posizionale, sono state implementate solo per il nodo <morf> e non per quello <chunk>.

(11) <frase val="questa storia si tratta di un ragazzo che possiede due animali un cane e una rana .">
<chunk val="N_C">
<morf p5=" p3=" p1=" p2="nn" p4="pron_p" p6="nn" pos="0" lemma="QUESTO" Head="det" gend="f" numb="s">questa</morf>
<morf p5= "p3=" p1= "det" p2="pron_p" p4="nn" p6="prep" pos="1" lemma= "STORIA" Head="nn" gend="f" numb="s">storia</morf>
</chunk>
...
<morf p5="conj" p3="det" p1="conj_c" p2=" p4= "nn" p6="art_i" pos="11" lemma= "POSSEDERE" Head="v_fin" mood="ind" tense="pres" pers="3" numb="s" flag="tr">
<token original="possiede">possessa</token>
</morf>
<morf p5="det" p3="conj_c" p1="v_fin" p2="nn" p4="art_i" p6="nn" pos="12" lemma="DUE">due</morf>
<morf p5="conj_c" p3="v_fin" p1=" p2="art_i" p4="nn" p6="conj_c" pos="13" lemma= "ANIMALE" Head="nn" gend="m" numb="p">animali</morf>
</chunk>
<chunk val="NA_C">
<morf p5="v_fin" p3="" p1="nn" p2="nn" p4="conj_c" p6="art_i" pos="14" lemma="UN" Head="art_i" gend="m" numb="s">un</morf>
<morf p5="" p3="nn" p1="art_i" p2="conj_c" p4="art_i" p6="nn" pos="15" lemma= "ANE" Head="nn" gend="m" numb="s">cane</morf>
</chunk>
<chunk val="COORD_C">
<morf p5="nn" p3="art_i" p1="nn" p2="art_i" p4="nn" p6="" pos= "16" lemma= "E" Head="conj_c">e</morf>
</chunk>
<chunk val="N_C">
<morf p5="art_i" p3="nn" p1="conj_c" p2="nn" p4="" p6="" pos="17" lemma= "UNA" Head="art_i" gend="f" numb="s">una</morf>
<morf p5="nn" p3="conj_c" p1="art_i" p2="" p4="" p6="" pos="18" lemma= "RANA" Head="nn" gend="f" numb="s">rana</morf>
</chunk>
<chunk val="PUNC_C">
<morf p5="conj_c" p3="art_i" p1="nn" p2="" p4="" p6="" pos= "19" lemma=".">.</morf>
</chunk>
</frase>

9. XQuery

Il vantaggio dell'utilizzo di xml rispetto ad altri sistemi di annotazione è che il testo può essere interrogato utilizzando uno strumento specifico, chiamato **xquery** (query per xml). La sintassi di xquery permette di effettuare ricerche selettive solo su specifici livelli della griglia di annotazione xml. Utilizzando quello che viene chiamato **xpath**, è possibile infatti individuare uno specifico nodo xml percorrendo in maniera discendente tutti i nodi superiori. Nel nostro caso per fare una ricerca nel tag <morf> bisogna passare attraverso /file/frase/chunk. Dal momento tuttavia che esiste un solo nodo morf nella nostra griglia, è possibile abbreviare l'xpath con //morf. E' poi cercare la stringa immessa nell'elemento taggato stesso con Text(), o in uno dei suoi attributi, che sono identificati con @attributo. Sono possibili anche ricerche con imposizione di due condizioni necessarie o alternative.

Qui di seguito possiamo vedere alcuni esempi di xquery applicabili al nostro testo:

(12) ricerca per lemma

```
//morf[@lemma=cane]
```

(13) ricerca per source

```
//morf[(text()=cane) or (token/text()=cane)]
```

(14) ricerca target (trova tutte le istanze di "cane", sia quelle originariamente "cane", es. "il cane", sia quelle tendenzialmente "cane", es. "il cano")

```
//morf[(text()=cane) or (token/text()=cane) or (token/target()=cane) ]
```

(15) ricerca posizionale (aggettivo preceduto da avverbio)

```
//morf[(@Head='adj') and (@p1='adv') ]
```

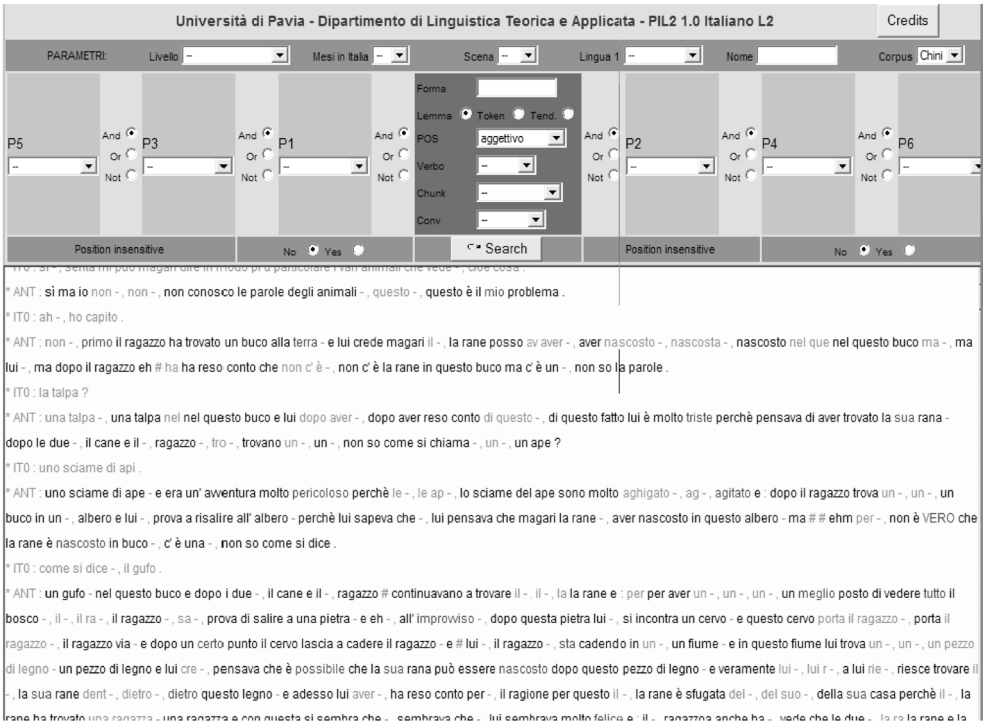
(16) ricerca per chunk (ricerca di chunk nominali)

```
//chunk[@val='N_C']
```

10. Interfaccia di ricerca PIL2

Tra i vangtaggi del sistema xquery vi è quello di essere facilmente implementabile all'interno di una pagina html. Questo ha reso possibile realizzare un prototipo di interfaccia di ricerca per PIL2.

Qui accanto, Figura 1, possiamo vedere una schermata dell'applicazione: in alto la maschera di ricerca, e in basso l'output di una query, con i risultati evidenziati in rosso.



Vediamo qui di seguito in maniera più specifica le funzioni della maschera di ricerca.

Selezione parametri. La sezione blu in alto contiene una serie di parametri che è possibile settare per ridurre la ricerca su un numero limitato di file.

Livello	grado di competenza linguistica dell'informante
Mesi	tempo di permanenza in Italia al momento della registrazione
Scena	argomento della conversazione o del testo
Lingua 1	lingua materna dell'informante (attualmente sono disponibili 10 scene dal film Pane e Tulipani per il Corpus ISA e Frog story per il Corpus Chini)
Nome	campo in cui inserire il codice di un singolo informante per recuperare tutti i suoi documenti

Ricerca per stringhe. Il campo *Forma* può contenere una sequenza di caratteri che si vogliono recuperare nel testo. E' possibile inserire stringhe corrispondenti a forme complete, o utilizzare il carattere jolly "*" per ricercare stringhe non complete. Il

carattere jolly può precedere la stringa ("*ato", che permette di cercare forme che terminano con la stringa "ato") o seguire la stringa ("mang*", che permette di cercare forme che iniziamo con la stringa "mang"). La ricerca per stringhe può essere effettuata utilizzando tre opzioni diverse:

Lemma	viene confrontata la stringa immessa con la forma lemmatizzata di ciascun token del corpus selezionato
Source	la stringa immessa viene confrontata con la forma esatta
Target	la stringa immessa viene confrontata sia con la forma esatta che con la forma tendenziale

Ricerca per PoS. L'opzione *PoS* permette di effettuare la ricerca per Parti del discorso (Parts of Speech). Nel menu a tendina è possibile selezionare a quale classe di parole debba appartenere la forma cercata.

Le classi sono le seguenti:

aggettivo	preposizione	pronome	nome
avverbio	interiezione	(non personale)	nome proprio
congiunzione coordinativa	numerale cardinale	pronome personale	verbo non finito
congiunzione subordinativa	numerale ordinale	articolo determinativo	verbo finito
determinatore		articolo indeterminativo	

Si noti che non esistono classi sovraordinate, quindi volendo recuperare tutti i numerali, si dovranno fare due ricerche, una per gli ordinali e una per i cardinali, e lo stesso dicasi per congiunzioni, articoli, verbi, nomi e pronomi.

Flag verbali. Nello standard di etichettatura adottato, i verbi sono trattati in maniera particolare, e ad essi viene aggiunto manualmente un flag che riporta informazioni aggiuntive riguardanti la valenza.

intr_av	verbo intransitivo con avere
intr_es	verbo intransitivo con essere
tr	verbo transitivo
imp	impersonale
pass	passivo
due_aux	doppio ausiliare
exist	esistenziale
mod	modale
lvc	light verb construction
pron	pronominale

Chunk. Secondo quanto visto al paragrafo 5, i chunk sono unità sintattiche minime non ricorsive. I valori possibili per questo campo sono:

N_C	gruppo nominale semplice
NA_C	gruppo nominale con aggettivo
P_C	gruppo preposizionale semplice
PNA_C	gruppo preposizionale con aggettivo
BE_C	copula + aggettivo, copula + part pass inaccusativo
CHE_C	tutti gli usi di "che"
COORD_C	tutti gli usi di "e", "o", "ma"
FV_C	gruppi verbali di modo finito
I_C	gruppi verbali di modo infinito
G_C	gruppi verbali di modo gerundio
ADV_C	avverbi isolati o a destra di una testa
PUNC_C	punteggiatura
ADJ_C	aggettivi qualificativi
SUBORD_C	congiunzioni subordinative
U_C	chunk non riconosciuto

La ricerca per chunk si combina con la ricerca per Forma, per PoS o per Flag verbale su una sola posizione, ad esempio per cercare tutte le forme "rana" inserite in un P_C (chunk preposizionale). Non può invece essere combinata con la ricerca per PoS su più posizioni, dal momento che, attualmente l'annotazione posizionale è implementata, come si è visto al paragrafo 8, solamente a livello del nodo <morf>.

Query su più posizioni. La ricerca per PoS è possibile anche su più di una posizione, selezionando la categoria cercata nella parte rossa (Posizione centrale, o P0) e costruendo l'intorno sintattico desiderato utilizzando i campi in grigio, numerati da 1 a 6. Le posizioni con numero dispari si trovano a sinistra del nucleo centrale, e quelle con numero pari si trovano a destra.

La ricerca per PoS su più posizioni essere combinata semplice o "ibrida":

- *semplice*: selezionare una PoS in posizione P0 e uno o più PoS nelle posizioni successive (es. P1= aggettivo + P0=nome cercherà un aggettivo seguito da un nome, P3=aggettivo + P0=nome, cercherà un aggettivo seguito una parola qualsiasi seguita da un nome)
- *ibrida*: permette definire in maniera più articolata la posizione centrale, utilizzando anche gli altri criteri di selezione presenti del campo rosso.

11. Sviluppi futuri

Come si è detto, PIL2 è attualmente un prototipo, e sono in progetto diversi interventi di miglioramento.

Dal punto di vista del **parsing**, sono in corso di sperimentazione alcune modifiche alle regole di chunking stesse, che permettano al software di essere più tollerante alla variabilità dell'Interlingua. In particolare si stanno valutando i risultati di altri parser, tra i quali una versione modificata di Italian NLP, nella quale le regole di chunking sono state rilassate in modo da consentire il riconoscimento di sintagmi (e la conseguente corretta etichettatura per PoS) anche in assenza di accordo corretto.

Dal punto di vista **dell'etichettatura**, rimane da implementare la ricerca posizionale per chunk e per forma. In modo che tutte le possibilità di ricerca attualmente presenti nella maschera di ricerca in posizione P0 (riquadro rosso nell'interfaccia di ricerca) siano accessibili anche su tutte le altre posizioni (sezioni in grigio).

Per quanto riguarda **l'interfaccia**, attualmente essa non implementa la ricerca di tutte le categorie disponibili nell'etichettatura. Rimangono da implementare le ricerche sulle informazioni morfologiche e flessionali, come modi e tempi verbali, numero, genere, caso, grado. Attualmente inoltre l'interfaccia è utilizzabile off line lanciando un'applicazione html. In una prossima fase del progetto è prevista la messa on line di una versione beta dell'interfaccia di ricerca, con la possibilità di poter effettuare ricerche sui diversi corpora pavesi.

BIBLIOGRAFIA

DE MAURO, T., MANCINI F., VEDOVELLI, M., & VOGHERA, M., *Lessico di frequenza dell'italiano parlato,* ETASLIBRI, Milano, 1993.

FEDERICI, S., MONTEMAGNI, S. & PIRRELLI, V., Shallow parsing and text chunking: a view on underspecification in syntax, in *Proceedings of the Eight European Summer School In Logic, Language and Information*, Prague, Czech Republic, 1996.

KLEIN, W. & PERDUE, C., The basic variety, or: Couldn't language be much simpler, in *Second Language Research* 13, 1997, 301-47.

MACWHINNEY, B., *The CHILDES Project. Volume I: Tools for Analyzing Talk: Transcription Format and Programs*, Lawrence Erlbaum, Mahwah (NJ), 2000.

RASTELLI S., ISA - Un corpus di italiano scritto di americani: problemi di annotazione, primi campionamenti e osservazioni sulla didattica ad anglofoni, in *ITALS* III, 8, 2005.

RASTELLI, S., Going beyond errors: position and tendency tags in a learner corpus, in A. Sansò (ed.), *Language Resources and Linguistic Theory*, Franco Angeli, Milano, 96-109, 2007.

SADEGH ASTANEH
sadegh.astaneh@unimi.it

FRANCESCA FRONTINI
francescafrontini@unipv.it