Indice

Un ringraziamento particolare a Dominique Brunato

Dedicato al Prof. Mario Barbieri.

Introduzione

Pensando dove possa arrivare l'italiano parlato del terzo millennio, è lecito chiedersi se e in quale misura le tecnologie linguistico-computazionali possano essere di aiuto nel monitoraggio della lingua italiana (Montemagni, 2013), ma prima salta alla mente un'altra domanda: che cosa è un “monitoraggio” della lingua? Il monitoraggio di una lingua, o meglio, di un testo, comprende un insieme arbitrario di estrazioni statistiche riguardanti la sua struttura linguistica a vari livelli, compiute con tecnologie linguistico-computazionali (algoritmi sviluppati con i linguaggi di programmazione, trasposizione di un testo dalla versione cartacea alla versione digitale, ecc.).

L'identificazione della struttura linguistica di un testo avviene tipicamente in modo incrementale, attraverso analisi linguistiche a livelli di complessità crescente; partendo da una “segmentazione” di un testo potremmo considerare in un primo luogo di dividere il testo per frasi, considerando come elemento delimitatore di frase il punto, ma, nel monitoraggio della lingua, le unità di base del testo in formato digitale sono i *token*, una famiglia eterogenea che raggruppa, oltre alle parole ortografiche, anche numeri, sigle, segni di punteggiatura, e altri elementi del vastissimo inventario testuale (Lenci, 2005).

«L’intuizione di partenza riguardante il “potere diagnostico” delle tecnologie linguistico-computazionali in compiti di monitoraggio linguistico trova conferma in un recente filone di studi avviato a livello internazionale all’interno del quale le analisi linguistiche generate da strumenti di trattamento automatico del linguaggio sono usate, ad esempio, per misurare la leggibilità di testi (Montemagni, 2013, p. 145)», nonché per supportare la semplificazione semiautomatica degli stessi[1].

In questa direzione di ricerca si colloca questo elaborato, che si è proposto di analizzare come le tecnologie linguistico computazionali possano essere impiegate per favorire lo sviluppo di sistemi di semplificazione semiautomatica del testo.

Il punto di partenza di questa ricerca è stata l'annotazione[2] di un corpus, costituito in

1 Vedi ad esempio i lavori di (Saggion, 2014) per lo Spagnolo.

2 I testi annotati sono testi in cui viene codificata dell'informazione linguistica in associazione al testo. L'unità di annotazione è il tag, una parola chiave o un termine associato a un'informazione, che descrive l'oggetto rendendo possibile la classificazione e la ricerca di informazioni basata su parole chiave; i tags sono generalmente scelti in base a criteri informali e personalmente dagli autori/creatori dell'oggetto dell'indicizzazione.

modo tale da rappresentare una risorsa esemplificativa di un tipo di semplificazione che si può definire “intuitiva”. Infatti, caratteristica di questo corpus, è di essere costituito da due versioni allineate: una contenente dei testi nella loro forma “originale”, e l'altra gli stessi testi in una versione riadattata da alcune insegnanti per diverse categorie di persone (principalmente studenti stranieri con una competenza limitata di italiano, inseriti in ogni ordine e grado scolastico).

Dopo aver portato a termine l'opera di annotazione, sono state eseguite diverse analisi linguistico-computazionali finalizzate ad intercettare gli interventi di semplificazione degli insegnanti: attraverso tali analisi è stato evidenziato l'effetto di ogni regola, o combinazioni di regole, per la semplificazione del testo, e si è mostrato quali regole si sono rivelate più efficaci per lo studio e la comparazione dei due corpora.

Più nel dettaglio, la tesi si articola in tre capitoli. Il primo capitolo contiene due paragrafi, uno che descrive cosa si intende per semplificazione intuitiva, contrapponendola ad un approccio di semplificazione strutturata (che non è oggetto di questa tesi), e l'altro le classificazioni di conoscenza della lingua divisi in sei livelli.

Il secondo contiene la descrizione dei corpora trattati e specifiche osservazioni riguardanti le applicazioni della semplificazione intuitiva: come e in che modo le insegnanti hanno fornito una versione semplificata di alcuni testi per diverse categorie di persone.

Il terzo capitolo contiene l'analisi distribuzionale e qualitativa delle regole di semplificazione sui corpora, ed alcune osservazioni riguardanti i risultati delle analisi linguistico-computazionali su vari livelli anche attraverso la descrizione del software READ-IT (Dell'Orletta et al, 2011).

Infine in appendice vengono riportati i codici e gli script dei programmi creati per l'estrazione dell'informazione d'interesse.

1. SEMPLIFICAZIONE DEL TESTO E LIVELLI DI CONOSCENZA DELLA LINGUA

Prima di osservare quali interventi le diverse insegnanti, autrici della semplificazione, hanno effettivamente compiuto sul testo di partenza, per favorirne la comprensione, bisogna capire quali siano gli approcci di semplificazione possibili e come sono stati affrontati dagli studi linguistico-computazionali.
Inoltre, poiché si affronta un metodo di semplificazione rivolta principalmente a studenti di madrelingua non italiana, si farà spesso riferimento ad alcuni livelli di conoscenza della lingua e talvolta, alludendo proprio ad alcune sigle come “B1”, “B2”, “A1”, i lettori dell'elaborato possono avere poca chiarezza di fronte a certe sigle: nel paragrafo 1.2 si affronterà la questione più nel dettaglio.

1.1 Introduzione ai metodi di semplificazione del testo

La semplificazione manuale del testo può seguire due approcci: l'approccio intuitivo e l'approccio strutturale. L'approccio strutturale segue delle regole definite a priori da esperti e concepite per un destinatario ben definito, come ad esempio i bambini con difficoltà di comprensione del testo. Queste regole sono potenzialmente sfruttabili da sistemi linguistico-computazionali. È questo il metodo seguito dal progetto europeo *Terence* che è stato finalizzato alla pianificazione, allo sviluppo, e alla valutazione di un sistema adattivo di apprendimento per *poor comprehenders* sia per la lingua italiana che per la lingua inglese[3].
Ad esempio, in questo contesto, le costruzioni che sono state tipicamente semplificate nei testi per bambini sono quelle relative alle voci passive, alle proposizioni relative ed ipotetiche, dal momento che ricerche psicolinguistiche sulla comprensione attraverso la lettura hanno evidenziato che la comprensione di un testo è più legata alla coerenza e alla relazione fra gli elementi del testo che semplicemente alla somma delle caratteristiche linguistiche delle parole o delle frasi individuali nel testo. Inoltre è stato evidenziato che durante la lettura i bambini sono guidati a riconoscere e usare i cosiddetti “coesive links”, ovvero degli elementi che

3 http://terenceproject.eu/web/guest/home

fanno sì che un bambino (o anche una persona adulta), dopo aver letto un testo, riconosca in quel testo un dato che gli è particolarmente familiare o di sua appartenenza, e, di conseguenza, apprenda le relazioni semantiche nei testi. Il processo della semplificazione del testo tende a conservare quanto più possibile della struttura linguistica e testuale della storia autentica. Invero anche i bambini che si sforzano di leggere hanno bisogno di leggere testi con un vocabolario sufficientemente stimolante e una sintassi che migliori le loro abilità di lingua e di lettura. In linea con questo principio differentemente dagli altri sistemi esistenti il sistema di semplificazione offre ai lettori livelli graduali di difficoltà, accostandosi progressivamente alla difficoltà che i lettori incontrano nel testo autentico. Ma a tutti i livelli, l'attenzione è posta sulla struttura globale e sulla coerenza testo, così che anche la versione più semplice del testo conservi quanto più possibile la struttura narrativa e lo stile della storia originale.

L'oggetto di trattazione di questo elaborato è invece la semplificazione intuitiva, ovvero un tipo di semplificazione del testo che è normalmente raggiunta dalle insegnanti seguendo la loro conoscenza del contesto scolastico e delle abilità linguistiche dei propri studenti. Nel presentare i testi utilizzati per studiare questo tipo di approccio, verranno presentate più dettagliatamente le strategie di interventi sul testo che caratterizzano il lavoro di queste insegnanti.

1.2 Classificazione di conoscenza delle lingue

Il *Quadro comune europeo di riferimento per la conoscenza delle lingue* (QCER), in inglese *Common European Framework of Reference for Languages* (CEFR) è un sistema descrittivo impiegato per definire le abilità conseguite da chi studia una lingua straniera europea, nonché allo scopo di indicare il livello di un insegnamento linguistico. È stato messo a punto dal Consiglio come parte principale del progetto *Language Learning for European Citizenship* (apprendimento delle lingue per la cittadinanza europea) tra il 1989 e il 1996. Suo principale scopo è fornire un metodo per accertare e trasmettere le conoscenze applicato a tutte le lingue d'Europa; nel novembre 2001 una risoluzione del Consiglio d'Europa raccomandò di utilizzare il QCER per costruire sistemi di validazione dell'abilità linguistica.

Il quadro comune europeo di riferimento distingue tre ampie fasce di competenza ("Base", "Autonomia" e "Padronanza"), ripartite a loro volta in due livelli ciascuna per un totale di sei livelli complessivi, e descrive ciò che un individuo è in grado di fare in dettaglio a ciascun livello nei diversi ambiti di competenza: comprensione scritta (comprensione di elaborati scritti), comprensione orale (comprensione della lingua parlata), produzione scritta e produzione orale (abilità nella comunicazione scritta e orale).

A - Base

- **A1 - *Livello base***

  L'individuo comprende e usa espressioni di uso quotidiano e frasi basilari dirette a soddisfare bisogni di tipo concreto. Chi padroneggia questo livello sa presentare se stesso/a e gli altri ed è in grado di fare domande e rispondere su particolari dati personali (come dove abita, le persone che conosce e le cose che possiede). Interagisce in modo semplice, purché l'altra persona parli lentamente e chiaramente e sia disposta a collaborare.

- **A2 - *Livello elementare***

  L'individuo comunica in attività semplici e di abitudine che richiedono un semplice scambio di informazioni su argomenti familiari e comuni. Sa descrivere in termini semplici aspetti della sua vita, dell'ambiente circostante; sa esprimere bisogni immediati.

B - Autonomia

- **B1 - *Livello intermedio o "di soglia"***

  L'individuo comprende i punti chiave di argomenti familiari che riguardano la scuola, il tempo libero e argomenti circostanti la quotidianità, inoltre sa muoversi con disinvoltura in situazioni che possono verificarsi mentre viaggia nel paese di cui parla la lingua. È in grado di produrre un testo semplice relativo ad argomenti che siano familiari o di interesse personale. È in grado di esprimere esperienze ed avvenimenti, sogni, speranze e ambizioni e di spiegare

brevemente le ragioni delle sue opinioni e dei suoi progetti.

- **B2** - ***Livello intermedio superiore***

L'individuo comprende le idee principali di testi complessi su argomenti sia concreti che astratti e le discussioni tecniche sul suo campo di specializzazione. È in grado di interagire con una certa scioltezza e spontaneità che rendono possibile un'interazione naturale con i parlanti nativi senza sforzo per l'interlocutore. Sa produrre un testo chiaro e dettagliato su un'ampia gamma di argomenti e spiegare un punto di vista su un argomento fornendo i pro e i contro delle varie opzioni.

C - Padronanza

- **C1** - ***Livello avanzato o "di efficienza autonoma"***

L'individuo comprende un'ampia gamma di testi complessi e lunghi e ne sa riconoscere il significato implicito. Si esprime con scioltezza e naturalezza. Usa la lingua in modo flessibile ed efficace per scopi sociali, professionali ed accademici. Riesce a produrre testi chiari, ben costruiti, dettagliati su argomenti complessi, mostrando un sicuro controllo della struttura testuale, dei connettori e degli elementi di coesione.

- **C2** - ***Livello di padronanza della lingua in situazioni complesse***

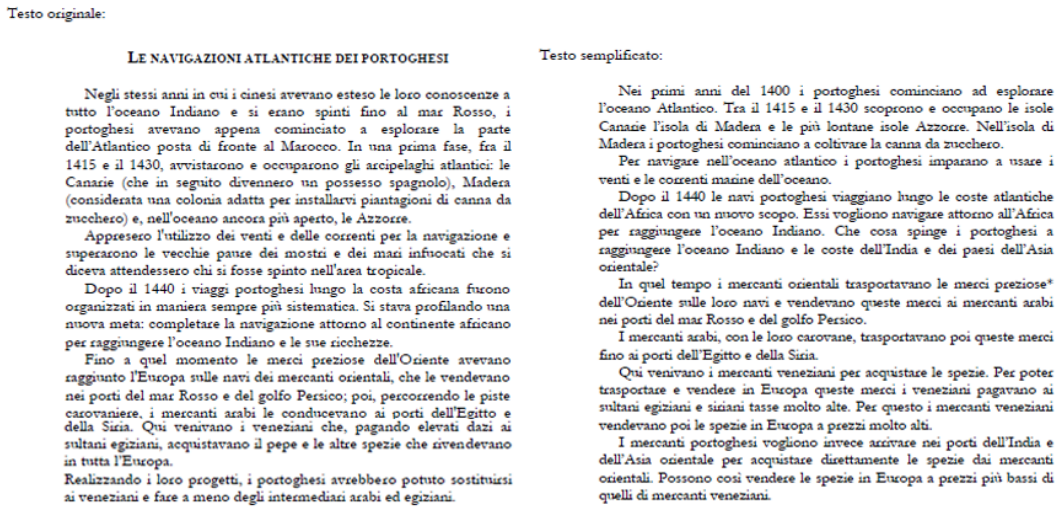
L'individuo comprende con facilità praticamente tutto ciò che sente e legge. Sa riassumere informazioni provenienti da diverse fonti sia parlate che scritte, ristrutturando gli argomenti in una presentazione coerente. Sa esprimersi spontaneamente, in modo molto scorrevole e preciso, individuando le più sottili sfumature di significato in situazioni complesse.

2. DESCRIZIONE DEI CORPORA E METODOLOGIE UTILIZZATE DAGLI INSEGNANTI PER LA SEMPLIFICAZIONE DEI TESTI

Come anticipato nel capitolo precedente, vediamo ora come diversi insegnanti hanno, intuitivamente, apportato una versione semplificata di diversi testi, poi confluiti nei corpora qui analizzati (*corpora* plurale di *corpus* che indica un insieme di testi), spaziando tra molteplici tipologie di argomento e di narrazione; è necessario osservare come le diverse insegnanti, seppur non con metodi linguistico-computazionali, ma guidati dalle loro competenze e dalle finalità educative del proprio intervento non solo hanno proposto versioni semplificate di diversi testi, ma, alla successiva lettura di essi da parte degli alunni, hanno affiancato alcuni esercizi e compiti tali che la comprensione sia favorita maggiormente soprattutto in considerazione del profilo linguistico degli alunni destinatari.

I testi trattati sono 24 per ciascun sotto-corpus parallelo allineato. Tipicamente un corpus parallelo allineato comprende testi nella loro lingua originale definita come L1, e nella loro traduzione in un'altra lingua (L2); nel caso qui esaminato invece la versione allineata rappresenta una versione semplificata ma sempre nella lingua di partenza. L'unità tipica di allineamento è la frase:

Figura 1. Un esempio di passaggio dal testo originale al semplificato



Testo originale:

LE NAVIGAZIONI ATLANTICHE DEI PORTOGHESI

Negli stessi anni in cui i cinesi avevano esteso le loro conoscenze a tutto l'oceano Indiano e si erano spinti fino al mar Rosso, i portoghesi avevano appena cominciato a esplorare la parte dell'Atlantico posta di fronte al Marocco. In una prima fase, fra il 1415 e il 1430, avvistarono e occuparono gli arcipelaghi atlantici: le Canarie (che in seguito divennero un possesso spagnolo), Madera (considerata una colonia adatta per installarvi piantagioni di canna da zucchero) e, nell'oceano ancora più aperto, le Azzorre.

Appresero l'utilizzo dei venti e delle correnti per la navigazione e superarono le vecchie paure dei mostri e dei mari infuocati che si diceva attendessero chi si fosse spinto nell'area tropicale.

Dopo il 1440 i viaggi portoghesi lungo la costa africana furono organizzati in maniera sempre più sistematica. Si stava profilando una nuova meta: completare la navigazione attorno al continente africano per raggiungere l'oceano Indiano e le sue ricchezze.

Fino a quel momento le merci preziose dell'Oriente avevano raggiunto l'Europa sulle navi dei mercanti orientali, che le vendevano nei porti del mar Rosso e del golfo Persico; poi, percorrendo le piste carovaniere, i mercanti arabi le conducevano ai porti dell'Egitto e della Siria. Qui venivano i veneziani che, pagando elevati dazi ai sultani egiziani, acquistavano il pepe e le altre spezie che rivendevano in tutta l'Europa.

Realizzando i loro progetti, i portoghesi avrebbero potuto sostituirsi ai veneziani e fare a meno degli intermediari arabi ed egiziani.

Testo semplificato:

Nei primi anni del 1400 i portoghesi cominciano ad esplorare l'oceano Atlantico. Tra il 1415 e il 1430 scoprono e occupano le isole Canarie l'isola di Madera e le più lontane isole Azzorre. Nell'isola di Madera i portoghesi cominciano a coltivare la canna da zucchero.

Per navigare nell'oceano atlantico i portoghesi imparano a usare i venti e le correnti marine dell'oceano.

Dopo il 1440 le navi portoghesi viaggiano lungo le coste atlantiche dell'Africa con un nuovo scopo. Essi vogliono navigare attorno all'Africa per raggiungere l'oceano Indiano. Che cosa spinge i portoghesi a raggiungere l'oceano Indiano e le coste dell'India e dei paesi dell'Asia orientale?

In quel tempo i mercanti orientali trasportavano le merci preziose* dell'Oriente sulle loro navi e vendevano queste merci ai mercanti arabi nei porti del mar Rosso e del golfo Persico.

I mercanti arabi, con le loro carovane, trasportavano poi queste merci fino ai porti dell'Egitto e della Siria.

Qui venivano i mercanti veneziani per acquistare le spezie. Per poter trasportare e vendere in Europa queste merci i veneziani pagavano ai sultani egiziani e siriani tasse molto alte. Per questo i mercanti veneziani vendevano poi le spezie in Europa a prezzi molto alti.

I mercanti portoghesi vogliono invece arrivare nei porti dell'India e dell'Asia orientale per acquistare direttamente le spezie dai mercanti orientali. Possono così vendere le spezie in Europa a prezzi più bassi di quelli di mercanti veneziani.

Adesso vediamo le diverse applicazioni ai testi scelti da parte delle insegnanti, inconsapevoli forse di aver prodotto corpora paralleli. Il riferimento alle fonti da cui i

testi derivano è riportato in bibliografia.

2.1 Anna Frank

Il primo testo preso in considerazione è stato quello di Anna Frank, formato da ben dieci paragrafi di frasi semplificate.
L'obiettivo dell'insegnante è stato quello di fornire un testo di narrativa semplificato per studenti appartenenti alla Scuola secondaria di primo grado, in modo da poter fornire un messaggio molto immediato riguardante il contesto storico (secondo conflitto mondiale e persecuzione degli Ebrei), affiancando inoltre la richiesta di alcuni prerequisiti come la visione de *La vita è bella* di Roberto Benigni e l'ascolto della colonna sonora di *Parla con lei* di Pedro Almodovar.
Per indirizzare meglio gli allievi alla comprensione e alla lettura dell'opera, l'insegnante ha inoltre introdotto alcuni dati biografici e sulla protagonista del diario e alcune informazioni riguardanti il contesto storico tramite un approccio empatico al testo:
Figura 2. “Chi è Anna Frank?”, breve illustrazione delle biografia di Anna Frank

casa di Anna Frank ad Amstermdan

Nell'estate del 1942 Anna si nasconde con la famiglia e alcuni amici in un alloggio segreto, nel centro della città di Amsterdan. Pochissime persone conoscono il rifugio segreto dei Frank.
Tra queste, ci sono **Elli** e **Miep**, due amiche di Anna. Elli e Miep hanno il compito di portare ai rifugiati cibo e informazioni su quello che succede nel mondo esterno.

il campo di concentramento di Bergen Belsen

Nell'estate del 1944 la polizia scopre il rifugio segreto, arresta e deporta (trasporta con la forza) Anna e i suoi familiari nel **campo di concentramento** di Bergen Belsen.

la tomba di Anna Frank

Anna muore nel campo di concentramento nel marzo del **1945.**

Dopo la guerra, alcune persone trovano il diario di Anna Frank nella soffitta dell'alloggio segreto

2.2 Testo di didattica riguardante la storia

Questo testo è stato semplificato allo scopo di fornire un testo semplificato e una buona inclusione sociale per gli studenti di origine straniera, favorendo una buona conoscenza dell'italiano parlato per lo studio di diverse discipline; gli studenti stranieri infatti, nonostante siano socialmente integrati nella scuola, non hanno molte nozioni cognitive e culturali, proprio per una scarsa conoscenza della lingua italiana. Inoltre la lingua per lo studio presenta caratteristiche specifiche che non possono essere ricavate dalla lingua della comunicazione, infatti gli studenti riscontrano diverse difficoltà nell'apprendimento di alcuni testi proprio per una complessità lessicale molto elevata.
Durante la comprensione di un testo vengono messe in atto diverse abilità mentali contemporaneamente non solo linguistiche, ma soprattutto cognitive, quali:

- selezionare;
- analizzare;
- generalizzare;
- classificare;
- dedurre;
- fare previsioni e ipotesi, che vengono poi ridefinite nel corso della lettura;
- collegare le informazioni che vengono presentate nel testo.

Talvolta I libri di testo, l'insegnante e la materia stessa involontariamente richiedono contenuti e saperi che suppongono l'allievo abbia. Se però questi elementi del sapere non sono presenti nella mentalità dell'allievo egli non può evocarli e di conseguenza non è in grado di comprendere il testo.
L'insegnante quindi si propone di offrire un testo ad alta comprensibilità, non un testo banale e riduttivo o “surrogato” estremamente ridotto di ciò che veniva spiegato nel testo originale, ma un testo capace di essere comprensibile e di semplice approccio; il testo semplificato quindi avrà un'alta comprensibilità per quanto riguarda il lessico, ovvero: viene utilizzato un vocabolario di base, vengono evitate le

espressioni idiomatiche e troviamo un uso molto ridotto delle nominalizzazioni. Viene semplificata la sintassi, costituita principalmente da frasi brevi, una struttura della frase secondo l'ordine SVO (Soggetto - Verbo - Oggetto), l'uso dei verbi nei modi finiti e nella forma attiva, l'uso esplicito dei soggetti, l'omissione delle forme impersonali e delle subordinazioni superiori al primo grado.
I testi ad alta comprensibilità non sostituiscono il libro di testo, ma lo affiancano, favorendo molto l'attenzione dell'allievo e insegnandogli tecniche di studio che non facciano leva sulla memoria, ma sulla comprensione delle informazioni e dei concetti.
Nonostante la resa ottimale del testo semplificato, gli alunni però dovranno sempre tenere il testo originale accanto a quello semplificato per evitare che si fossilizzino su un livello linguistico basso.

2.3 L'isola di Arturo

La professoressa ha fornito una versione riscritta e semplificata di un estratto del romanzo *L'isola di Arturo* di Elsa Morante, rivolgendo il suo operato ad apprendenti di italiano L2 livello B1, età 16/17 anni, ed ha stilato una lista di concetti molto comuni nell'universo delle semplificazioni testuali:

1. **Lessico** → per la sostituzione delle parole complicate sono state adottate due risorse di riferimento: VdB e LIP. La sostituzione è stata realizzata mediante l'uso di sinonimi più vicini alla lingua comune e di parafrasi esplicative.

2. **Morfosintassi:**
   - verbale: passato prossimo > presente storico

3. **Sintassi:**
   - ordine marcato > ordine non marcato (SVO)
   - preferenza paratassi
   - in caso di periodo ricco di subordinate:

- esplicitazione proposizioni implicite
- splitting in + frasi

2.4 Il mito di Pangu

Viene proposto un testo rivolto a studenti di quarta primaria e prima secondaria di primo grado con prerequisito di conoscenza della lingua italiana a livello A2 / B1.
A differenza delle metodologie trattate fino ad adesso, le professoresse hanno prefissato molti obiettivi che andavano oltre la proposta del testo semplificato, ma introducevano anche altre attività per verificare la comprensione completa del testo da parte degli alunni.
Qui sotto sono riportati tutti gli obiettivi delle insegnanti:

- Sviluppare il lessico (con attenzione anche all'uso figurativo dello stesso)
- Sviluppare la conoscenza di alcune strutture della lingua italiana
- Sviluppare la capacità di ascolto
- Sviluppare la capacità di confronto interculturale
- Conoscere le origini del mito
- Conoscere la struttura del mito
- Manipolare testi semplici

Qui sotto vengono riportate tutte le metodologie utilizzate dalle insegnanti per l'adempimento degli obiettivi riportati sopra:

- Punto di partenza sono testi di miti di diverse culture, eventualmente modificati e semplificati per renderli più accessibili.
- Attività di lettura ed esercizi per la comprensione globale e analitica del testo (domande chiuse e aperte)
- Analisi del lessico: riconoscimento, classificazione analisi, schede lessicali (i nomi degli dei, i nomi delle piante e dei frutti, i nomi dei personaggi fantastici).

- Analisi degli aspetti morfosintattici per far emergere le struttura delle lingua italiana (schemi grammaticali, esercizi di completamento e di produzione orale e scritta)
- Esercitazione guidate per avviare la produzione di un testo "favola" orale e scritto, ad esempio esercizi di riscrittura (cambia il finale, trasforma in dialogo o in testo senza dialogo)
- Esercizi per fare emergere e comprendere lo scopo del mito.
- Esercizi per la rilevazione degli elementi comuni dei miti.

Gli insegnanti hanno stimato inoltre una durata totale del lavoro approssimandola ad 8/10 ore per il percorso base di comprensione delle verifiche, ed altre 3/4 ore di consolidamento.
Il lavoro è stato principalmente composto dal racconto del mito e dalla successiva lettura, in modo che gli alunni prima comprendano l'argomento e successivamente, quando ascoltano il racconto letto dall'insegnanti, abbiano meno probabilità di distrazione, fenomeno che avviene spesso nel caso inverso (prima lettura e successiva spiegazione e conseguente interrogazione di uno o più alunni, i quali nella maggior parte dei casi non sanno cosa è stato appena letto, tranne nel caso in cui un alunno spicchi, a livello di comprensione, più dei suoi compagni).
La lettura è stata affiancata dalla visione del cartone animato targato Disney *Hercules* e dalla visione di alcune illustrazioni tratte da libri.
Seguono esercizi per la comprensione globale e analitica sul lessico, sulle strutture morfosintattiche; sono stati previsti inoltre degli approfondimenti di difficoltà maggiore per un livello di conoscenza della lingua italiana pari a B1.

2.5 L'avventura di due sposi

Questa raccolta di brani è nata con lo scopo di proporre un testo antologico ad una classe di alunni multietnici del terzo anno della scuola secondaria di primo grado e nel primo anno della scuola secondaria di secondo grado, aventi differenti livelli linguistici.
L'obiettivo primario da parte delle autrici del libro è quello di avvicinare gli studenti

stranieri a brani complessi. Non è da tralasciare il fatto che il testo in questione sia una novella tratta da una raccolta chiamata *Gli amori difficili* di Italo Calvino composta tra il 1949 e il 1967, indi per cui il linguaggio potrebbe risultare molto complicato ad una prima lettura da parte di uno studente straniero. Lo scopo è stato quello di insegnare a leggere, a capire e a produrre nella forma scritta, fornendo un metodo di lavoro efficace e specifico per l'apprendimento di un italiano come seconda lingua preponendo i seguenti obiettivi:

- Saper leggere, comprendere e analizzare un brano;
- Saper comunicare con maggior ricchezza lessicale, utilizzando diversi linguaggi (verbale, iconico, scritto);
- Saper scrivere, imparando tecniche riassuntive, esprimendo riflessioni personali;
- Saper riflettere sulla lingua: dalla riflessione linguistica – grammaticale all'applicazione e all'uso consapevole delle regole;

2.6 Io sono così

Questo è un testo tratto da un'antologia attualmente in uso per le scuole medie poiché rappresenta un punto di raccordo tra la 5^ elementare e la 1^ media; infatti il testo viene proposto in entrambe la classi. Come prerequisito l'allievo deve possedere un discreto livello linguistico che potrebbe coincidere con il livello A2 del paragrafo 1.2.

I professori dopo aver individuato gli elementi linguistici che aumentavano la complessità del testo hanno proposto le semplificazioni da apportare; prima di presentarlo agli alunni, il testo è stato riletto da altri insegnanti per verificare eventuali errori che possano impedire un corretto apprendimento del testo.

Per verificare l'efficacia della semplificazione apportata è stata presa in esame una signora rumena residente in Italia da qualche anno, con una conoscenza basilare della lingua italiana, rientrante nel livello minimo richiesto (A1, B1). Nella comprensione del testo la signora si è avvalsa della competenza linguistica della figlia che ha svolto un ruolo molto importante in questo caso: ha testato con la madre le semplificazioni necessarie alla comprensione del testo, ed ha poi suggerito quelle che si sono rivelate più proficue.

Gli obiettivi finali delle insegnanti sono stati:

- la comprensione del testo;
- la riproduzione guidata orale e scritta di un testo analogo;
- l'arricchimento lessicale nel contesto di una descrizione fisica e psicologica;
- la produzione libera orale e scritta di un testo analogo;
- La comprensione e la risoluzione di alcuni esercizi dediti alla comprensione del testo.

2.7 Il Pinocchio di Collodi

Uno dei testi composto da un elevato numero di frasi è stato sicuramente il terzo capitolo del Pinocchio di Collodi (ben 63 frasi originali e 51 frasi semplificate); l'obbiettivo da parte delle insegnanti di un istituto tecnico industriale statale è stato proporre una versione semplificata per studenti stranieri inseriti nel biennio della scuola secondaria di II grado, di diverse provenienze geografiche. Le competenze richieste riguardano la conoscenza della lingua per un livello A1 e B2. Per ogni livello linguistico richiesto, cambia la portata e la difficoltà delle richieste proposte dalle insegnanti.

Come scopo finale, gli studenti devono essere in grado di saper comprendere la sequenza narrativa nel testo presentato e comprende il significato di modi di dire ed espressioni idiomatiche largamente diffuse, e il testo ne è molto ricco, visto che sono presenti molti discorsi diretti e stereotipi della comunicazione parlata.

Prima di cimentare gli alunni nella lettura dell'opera collodiana, è stato fornito un incipit di brainstorming (una tecnica di creatività di gruppo per far emergere idee volte alla risoluzione di un problema).

Il capitolo preso in questione è il 3°. Dopo una attività di comprensione del testo, si chiede di individuare il significato di alcuni termini ed espressioni idiomatiche conservate dal testo originale. Si propongono poi attività lessicali e sulla morfosintassi. Si chiede una produzione orale a partire da un’illustrazione.

Infine, per il livello B1, si propone la lettura del testo originale, discusso e commentato insieme all’insegnante, e la produzione di un breve testo scritto.

2.8 Il gatto e il topo e la volpe e il corvo

Due dei testi più piccoli analizzati sono stati *Il gatto e i topi* e la V*olpe e il corpo*, entrambe due celebri favole di Esopo (il primo testo conteneva sette frasi tratte dalla favola originale e sei frasi semplificate dal docente, il secondo cinque frasi del testo originale e cinque frasi semplificate).

L'insegnante ha stilato una lista composta da tre campi costituenti gli obiettivi prefissati, gli inventari delle famiglie di parole necessari per la comprensione del testo, e gli esercizi finali per la comprensione del testo:

Figura 4. Obiettivi, ambiti lessicali, e contenuti linguistici richiesti e trattati

OBIETTIVI	AMBITI LESSICALI	CONTENUTI LINGUISTICI
Comprendere messaggi ascoltati utilizzando supporti extralinguistici (mimo, immagini...).		
Comprendere ed eseguire consegne orali e scritte in ambito scolastico.		Verbo essere indicativo presente: *c'è/ci sono*.
Nominare animali, oggetti, ... e collegarli ai rispettivi sostantivi.	Animali.	Formula domanda-risposta: che cos'è? che cosa sono? quanti sono? quanti... ci sono? di che colore è? di che colore sono?
Chiedere informazioni.	Oggetti scolastici e d'uso quotidiano.	
Interagire con i compagni e con il docente.	Colori.	Singolare e plurale.
Leggere e comprendere in modo autonomo un testo semplificato.	Numeri.	Concordanza nome/aggettivo.
Scrivere parole correttamente.		Pronomi personali soggetto.
Completare testi (cloze).		

l'opera del docente consiste nel proporre alcune immagini e consecutivamente chiedere alla classe che cosa rappresentino. Gli alunni non italofoni completano una scheda di esercizi riguardanti il lessico, mentre il resto della classe, magari in gruppetti di 2-3, fa delle ipotesi o anticipazioni sul contenuto della lettura che ci si appresta a fare.
Segue un breve confronto orale.
Successivamente il docente legge prima la favola nella versione originale agli alunni, leggendo con maggiore intensità le parole evidenziate in grassetto.
In seguito gli alunni rileggono in modo silenzioso; mentre uno dei ragazzi rilegge la favola, un compagno la mima.
Per entrambi i testi è stato portato a termine lo stesso tipo di lavoro.

2.9 Testo di storia del liceo

la professoressa ha scelto un paragrafo tratto da un manuale di Storia del primo anno dei Licei e l'ha riscritto cercando di mantenere la complessità di contenuto, dato che parte di esso non deve essere tralasciato, eliminando le forme linguistiche più ostiche e rendendo espliciti i nessi logici delle strutture sintattiche; ha apportato modifiche non solo al testo in se, ma anche ai titoli dei paragrafi, trasformando, nel primo titolo, il genitivo in una vera e propria subordinata inserendo un nuovo verbo. Questi sono i cambiamenti all'interno del nuovo testo semplificato:

- Sono stati ridotti il numero degli incisi;
- sono state separate nel modo più netto possibile le frasi principali dalle subordinate e si è cercato di ricorrere il meno possibile a gradi di subordinazione superiori al primo;
- a livello di morfologia sono state mantenute forme verbali ritenute solitamente complesse quali la diatesi passiva, il passato remoto e il trapassato prossimo, il modo congiuntivo, poiché gli apprendenti di un liceo, affrontando lo studio del latino, devono certamente avere familiarità con tali strutture.

La professoressa ha deciso però di conservare, e di evidenziare in grassetto, i termini specifici della disciplina, che rientrano a far parte del bagaglio lessicale degli

studenti: tali parole dovranno essere spiegate dagli studenti stessi o dall'insegnante, anche ricorrendo al dizionario, e le definizioni che verranno fuori dalla discussione dovranno essere raccolte in un glossario.

Assieme al testo semplificato e al glossario, è stata fornita una mappa concettuale del testo in questo modo sarà possibile per gli allievi esercitarsi a riferire l'argomento tenendo ben presenti le relazioni logiche tra le varie parti e concentrandosi su un utilizzo consapevole (non esclusivamente mnemonico) del lessico specifico precedentemente appreso.

A differenza degli altri testi ed oltre alle mappe concettuali fornite dalla professoressa, non sono stati proposti esercizi riguardanti la comprensione dei dati testuali.

3. METODI LINGUISTICO-COMPUTAZIONALI PER L'ANALISI E IL RICONOSCIMENTO DELLE REGOLE DI SEMPLIFICAZIONE UTILIZZATE

Gran parte del lavoro svolto per la stesura di questo elaborato di laurea triennale, è consistito nell'annotazione delle regole di semplificazione, mettendo a confronto i corpora allineati (originale – semplificato).
In questo capitolo conclusivo verranno di seguito:

1. spiegate il tipo di regole usate (p. 3.1)
2. riportati i risultati della distribuzione quantitativa di queste regole (p. 3.2)
3. analizzati i risultati della semplificazione dal punto di visto qualitativo. In questo contesto verrà introdotto nell'analisi uno strumento per la valutazione automatica della leggibilità, READ-IT (p. 3.3)

3.1 Regole utilizzate per l'annotazione del corpus allineato

In questo paragrafo sono elencate tutte le regole utilizzate per l'annotazione del corpus allineato, le quali specificano quale regola dedita alla semplificazione testuale è stata utilizzata; queste regole sono state redatte dal laboratorio di ricerca Italia Natural Language Processing Lab[4] dell'Istituto di Linguistica Computazianole "Antonio Zampolli" all'interno del Centro Nazionale delle ricerche di Pisa, ed equivalgono precisamente a tags[5] nella annotazione XML[6].
Questi tags non sono stati inseriti seguendo un'ordine di frequenza di utilizzo (dal più utilizzato al meno utilizzato), bensì secondo le tipologie di applicazione.
Sono stati evidenziati tags che trasformano una parola o una porzione di testo, regole che inseriscono un elemento mancante alla frase come una parole o un'altra frase, o un pezzo di essa; e infine l'ultima applicazione dei tags è consistita nella rimozione e nella cancellazione di una parola o di una parte intera della frase o del testo.
A se stanti troviamo prima tre regole operanti su una pozione di frasi o su un insieme

4 www.italianlp.it
5 Da ora in avanti considereremo equivalenti regola e tag.
6 L'XML (eXtensible Markup Language) è un linguaggio di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

di frasi:

- **<split>**: da inserire per segnalare che una parte della frase originale (es. proposizione coordinata) è stata resa come frase autonoma nella versione semplificata.

  Esempio

  Frase originale: “Il signor Van Daan ed io litighiamo sempre, invece egli va molto d’accordo con Margot”

  Frasi semplificate: “Il signor Van Daan ed io litighiamo sempre. Però il signor Van Daan va molto d’accordo con Margot.”

  Annotazione: Il signor Van Daan ed io litighiamo sempre <SPLIT>, invece egli va molto d’accordo con Margot </SPLIT>

- **<merge>**: da inserire per segnalare la frase (o le frasi) autonoma(e) nella versione originale che sono state unite in una singola frase nella versione semplificata.

  Esempio

  Frasi originali: “Gli ebrei debbono consegnare le biciclette. Gli ebrei non possono salire in tram, gli ebrei non possono più andare in auto.”

  Frase semplificata: “Gli ebrei non possono più andare in bicicletta, non possono salire in tram e non possono andare in auto.”

  Annotazione: <MERGE> Gli ebrei debbono consegnare le biciclette. Gli ebrei non possono salire in tram, gli ebrei non possono più andare in auto.</MERGE>

- **<spostamento>**: da aggiungere per segnalare uno spostamento di parti della frase (es. una frase subordinata che nell'originale precede la principale mentre nel semplificato segue la principale)

  Frase originale: “Dopo che è uscito il sole, sono andata al mare”

  Frase semplificata: “Sono andata al mare, dopo che è uscito il sole”

  Annotazione: <spostamento>Dopo che è uscito il sole,</spostamento>sono andata al mare.

  es.2 (spostamenti pre e post verbali)

Frase originale: “è arrivato il ragazzo”
Frase semplificata: “il ragazzo è arrivato”
<spostamento>è arrivato</spostamento>il ragazzo.

1) Trasformazione

- **<sost_lex>**: da aggiungere per segnalare una sostituzione lessicale (es. uso di un sinonimo) dall'originale al semplificato.
  possibili attributi*:*
  - “forma”: indicare il sostituto (può essere una o più parole)
  Esempio
  Frase originale: “da ieri il tempo è splendido fuori, ed io sono molto animata.”
  Frase semplificata: “da ieri il tempo è bellissimo, ed io sono molto felice.”
  Annotazione: da ieri il tempo è <SOST_LEX forma = “bellissimo”>splendido fuori</SOST_LEX>, ed io sono molto <SOST_LEX forma = “felice”>animata<SOST_LEX>

- **<anafora>:** da aggiungere per segnalare i casi in cui un pronome è stato sostituito da un sintagma nominale lessicale.
  Esempio
  Frase originale: “Poi le mise al sole per farle asciugare e tornò su in Cielo per andare a caccia.”
  Frase semplificata: “Poi mette le figure umane al sole per asciugare la creta e torna in Cielo per andare a caccia.”
  Annotazione: Poi <ANAFORA sost=“le figure umane”>le <ANAFORA> mise al sole per farle asciugare e tornò su in Cielo per andare a caccia

- **<tratti_verbo>**: da aggiungere per segnalare i casi in cui il verbo è stato mantenuto ma sono cambiati alcuni dei suoi tratti (tempo, modo, persona, es. dal passato remoto al presente).
  Anche in questo caso, indica i tratti come attributi del tag.

Esempio

Frase originale: “I bei tempi finirono nel maggio 1940”

Frase semplificati: “I bei tempi finiscono nel maggio 1940”

Annotazione: I bei tempi<TRATTI_VERBO modo= “indicativo” tempo = “presente” persona = “3.p”>finirono</TRATTI_VERBO> nel maggio 1940

- **<att_passivo>**: cambiamento della diatesi verbale (da attivo a passivo). Tag da marcare sul verbo.

  Esempio

  Frase originale: “Il povero fotografo ambulante, cui si deve quest'unica sua immagine, l'ha ritratta ai primi mesi di gravidanza.”

  Frase semplificata: “La fotografia è stata fatta durante i primi mesi della sua gravidanza da un fotografo ambulante.”

  Annotazione: Il povero fotografo ambulante, cui si deve quest'unica sua immagine, l'<ATT_PASSIVO>ha ritratta</ATT_PASSIVO> ai primi mesi di gravidanza.

- **<pass_attivo>**: cambiamento della diatesi verbale (da passivo ad attivo). Tag da marcare sul verbo.

- **<nominalizzazione_piu>:** da inserire nel caso in cui un verbo, nella versione semplificata diventa un sostantivo

  Attributo: “forma”

  Frase originale: “Per chi ha bisogno di nascondersi”

  Frase semplificata: “come nascondiglio”

  Annotazione: <sost_lex forma="come">per chi</sost_lex> <verbo_meno>ha bisogno</verbo_meno><nominalizzazione_piu forma ="nascondiglio">di nascondersi</nominalizzazione_piu>

- **<nominalizzazione_meno>:** da inserire per segnalare lo “scioglimento” di una nominalizzazione o di una perifrasi nominale, trasformata nella corrispondente struttura verbale.

Attributo: “forma”
Frase originale: “viene a noia”
Frase semplificata: “annoiano”
Annotazione: <nominalizzazione_meno forma="annoiano">viene a noia</nominalizzazione_meno>

2) Inserimento

- **<sogg_espl>**: da aggiungere per segnalare i casi in cui nella frase originale c'è un soggetto sottointeso che è stato esplicitato nella frase semplificata.
  Esempio
  Frase originale: “Così ha un’aria più allegra.”
  Frase semplificata: “Così la stanza ha un'aria più allegra.”
  Annotazione: “Così </SOGG_ESPL sog=“la stanza”> ha un'aria più allegra.”

- **<verbo_piu>**: da aggiungere per segnalare i casi in cui nella frase originale manca un verbo che è stato inserito nella frase semplificata.
  Può avere gli attributi “tempo”, “modo”, “persona”.
  Esempio
  Frase originale: “Perciò questo diario.”
  Frase semplificata: “Perciò desidero avere questo diario.”
  Annotazione: Perciò <VERBO_PIU modo = “indicativo” tempo = “presente” persona = “1.s.”></> questo diario.

- **<insert>**: da inserire per segnalare altri tipi di inserimento (parole che non sono soggetto o verbo) oppure sequenze di più parole.
  Attributo: “forma”
  Frase originale: “Le leggi antisemitiche si susseguivano l'una all'altra.”
  Frase semplificata: “A causa delle leggi antisemitiche, gli ebrei...”
  Annotazione: </insert forma=“A causa delle”>Le leggi antisemitiche si susseguivano l'una all'altra.

3) Rimozione

- **<verbo_meno>**: da aggiungere per segnalare i casi in cui un verbo nella frase originale è stato eliminato nella frase semplificata.

- **<sogg_sott>**: da aggiungere per segnalare i casi in cui nella frase originale c'è un soggetto esplicito che è stato sottinteso nella frase semplificata.

- **<delete>**: da aggiungere per segnalare una frase originale (o una parte di frase) completamente rimossa nella versione semplificata.
  Esempio
  Frase originale: “Sebbene sia umido, credo che ad Amsterdam non abbiamo mai costruito niente di più comodo per chi ha bisogno di nascondersi.”
  Frase semplificato: “E' umido ma è comodo come nascondiglio.”
  Annotazione: Sebbene sia umido<DELETE>, credo che ad Amsterdam non abbiamo mai costruito niente di più </DELETE>comodo per chi ha bisogno di nascondersi.

È stato previsto l'uso del tag <*manca_regola*> quando nessuna delle regole precedenti poteva essere applicata ai testi per intercettare il tipo di riscrittura o semplificazione. Per esempio:
Frase originale: “A volte mi domando: Che non ci sia nessuno capace di comprendere che, ebrea o non ebrea, io sono soltanto una ragazzotta con un gran bisogno di divertirmi e stare allegra?.”
Frase Semplificata: “Perché molte persone non capiscono che anche una ragazza ebrea ha bisogno di divertimento e di felicità?”

3.2 Una analisi distribuzionale dei testi marcati

Il lavoro di annotazione delle frasi è stato lungo e laborioso. A lavoro ultimato è stato molto probabile che alcuni tags non fossero nella corretta annotazione XML. La prima azione da svolgere è stata la validazione del documento grazie all'editor *Xml*

Copy Editor. Da ricordare che per portare a termine delle corrette analisi linguistico-computazionali, i corpora paralleli, previa annotazione, dovevano essere prima spezzati per separare i testi in due documenti differenti, contenenti uno le frasi originali, l'altro le frasi semplificate.

Utilizzando il linguaggio di programmazione Python, sono stati rintracciati subito i tags.

Qui sotto viene riportato un estratto di output di un programma che ci ha consentito di recuperare le seguenti statistiche: 1) Frequenza di applicazione delle regole, 2) Numero di frasi alle quali sono state applicate le varie regole, 3) Frequenza di combinazione di regole, senza frequenza delle singole parole nelle frasi. Sono stati selezionati solo i primi tags ordinati per frequenza.

L'output completo è riportato in appendice A.

```
1) Frequenza di applicazione delle REGOLE:
sost_lex 495.0

delete 251.0

insert 188.0

tratti_verbo 164.0

spostamento 108.0
```

(…)

```
2) numero di frasi alle quali sono state applicate le varie REGOLE:
sost_lex 206

delete 148

tratti_verbo 110

insert 108
```

(…)

```
3) Frequenza di combinazione di regole (senza frequenza) nelle
frasi:
sost_lex    25.0
sost_lex insert delete  11.0
spostamento sost_lex insert delete  8.0
tratti_verbo sost_lex   8.0
sost_lex delete   8.0
```

(…)

In questo output di un programma Python si può osservare quali siano le regole di semplificazione che hanno una maggiore frequenza di utilizzo, il numero di frasi alle quali sono state applicate le varie regole, e le varie combinazioni delle regole all'interno di una frase[7].

Osservando le frequenze assolute[8] dei tags notiamo subito quali sono le regole maggiormente applicate dalle insegnanti per la semplificazione di un testo.

Analizzando per esempio l'utilizzo del tag <*sost_lex*>, che ricorre ben 495, possiamo vedere come le insegnanti abbiano "abusato" di molteplici sostituzioni lessicali: fra queste, la trasformazione di alcuni aggettivi, come per esempio "bei" che diventa "felici", o i bigrammi[9] all'interno della frase "Un tempo la terra era vuota e senza abitanti" che diventano "*Una volta* la terra era vuota e *solitaria*".

Un altro tag molto ricorrente è il tag <*tratti_verbo*>. Andando a sbirciare tra i valori degli gli attributi di questo tag possiamo vedere che, nella maggior parte dei casi, molti verbi hanno cambiato il loro tempo in presente; per esempio la frase di uno dei testi di Anna Frank "alle tre qualcuno *suonò* alla porta." diventa "alle tre una persona *suona* alla porta."

Abbiamo visto come l'output di questo programma sia relativo a tutto il corpus dei testi originali senza considerare come le regole siano diversamente distribuite per ogni testo allineato. Per verificare le regole applicate singolarmente ad ogni testo è stato utilizzato un programma simile generante un nuovo output, di cui si riporta un estratto. Anche questo è visibile in app. A.

```
corpus # 1 : sost_lex 13.0 delete 4.0 insert 3.0 verbo_piu 2.0 merge
2.0 sogg_sott 2.0 verbo_meno 0.0 nominalizzazione_piu 0.0 anafora
0.0 nominalizzazione_meno 0.0 spostamento 0.0 sogg_espl 0.0
tratti_verbo 0.0 split 0.0 pass_attivo 0.0

corpus # 2 : sost_lex 14.0 insert 3.0 delete 3.0 spostamento 2.0
verbo_piu 2.0 tratti_verbo 1.0 verbo_meno 0.0 nominalizzazione_piu
0.0 anafora 0.0 nominalizzazione_meno 0.0 sogg_espl 0.0 merge 0.0
sogg_sott 0.0 split 0.0 pass_attivo 0.0
```

7 Il delimitatore di ogni frase è il punto.

8 Data una parola, contare quante volte ricorre all'interno del testo.

9 Sequenze di due parole consecutive

```
corpus # 3 : sost_lex 14.0 insert 12.0 tratti_verbo 8.0 delete 5.0
spostamento 4.0 verbo_piu 2.0 sogg_espl 2.0 merge 2.0 split 2.0
verbo_meno 1.0 anafora 1.0 nominalizzazione_piu 0.0
nominalizzazione_meno 0.0 sogg_sott 0.0 pass_attivo 0.0
```

(…)

```
corpus # 15 : tratti_verbo 20.0 sost_lex 16.0 spostamento 5.0 insert
4.0 anafora 4.0 delete 4.0 verbo_meno 0.0 nominalizzazione_piu 0.0
nominalizzazione_meno 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0
sogg_sott 0.0 split 0.0 pass_attivo 0.0
```

(…)

```
corpus # 23 : tratti_verbo 9.0 sost_lex 8.0 insert 5.0 verbo_meno
2.0 spostamento 2.0 verbo_piu 2.0 delete 2.0 nominalizzazione_piu
0.0 anafora 0.0 nominalizzazione_meno 0.0 sogg_espl 0.0 merge 0.0
sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 24 : sost_lex 63.0 delete 47.0 insert 28.0 spostamento 13.0
verbo_piu 5.0 merge 5.0 tratti_verbo 5.0 verbo_meno 4.0
nominalizzazione_piu 3.0 anafora 2.0 sogg_espl 2.0 split 1.0
pass_attivo 1.0 nominalizzazione_meno 0.0 sogg_sott 0.0
```

Grazie a questo output possiamo vedere quali e quante regole vengono applicate ad ogni sotto-corpus. Alcuni gruppi contengono un alto numero di frasi e, di conseguenza, alcune frequenze assolute possono avere un numero più alto rispetto alla media di applicazione delle regole per ogni blocco di frasi.
L'ordine decrescente di frequenza mostra che le regole più applicate ad ogni sotto-corpus siano *sost_lex* e *tratti_verbo*, analizzate al capito precedente.

3.3 Analisi qualitativa delle regole di semplificazione

Fino adesso abbiamo visto come sono distribuite le regole e quali combinazioni di regole possiamo trovare all'interno del corpus annotato; ora valutiamo l'effetto delle

regole rispetto di semplificazione rispetto alla leggibilità del testo.

A tale scopo è stato utilizzato un tool chiamato READ-IT[10] (Dell'Orletta et al, 2011), un'applicazione web capace di valutare la leggibilità di un testo e di estrarne il profilo linguistico. Con questo tool possiamo verificare quanto un testo sia leggibile, e verificare se le semplificazioni apportate dalle insegnanti siano effettivamente riuscite. L'output del programma si articola in due sezioni distinte dedicate a:

- La valutazione della leggibilità del documento effettuata da diversi modelli di analisi basati su diversi tipi di informazione, che potremmo vedere come diversi indici di leggibilità;
- la ricostruzione del profilo linguistico del testo, condotta in relazione a un sottoinsieme dei parametri utilizzati dal programma per la valutazione della sua leggibilità, articolati secondo il livello di descrizione linguistica di appartenenza. Questa seconda sezione è tesa a fornire elementi di analisi utili a comprendere i risultati riportati nella prima sezione: si tratta di informazioni utili per il linguista e il linguista computazionale che permettono di monitorare il funzionamento del sistema ed eventualmente correggerlo.

Il tool sfrutta una catena di analisi linguistica in grado di analizzare il testo fino all'analisi sintattica ed utilizza le caratteristiche linguistiche estratte da quest'analisi automatica per assegnare quattro livelli di leggibilità. La valutazione globale della leggibilità del testo viene condotta sulla base di diverse configurazioni di caratteristiche del testo che producono quattro modelli di leggibilità:

- Dylan BASE: in questo modello le caratteristiche considerate sono quelle tipicamente usate nelle misure tradizionali della leggibilità di un testo, ovvero la lunghezza della frase, calcolata come numero medio di parole per frase, e la lunghezza delle parole, calcolata come numero medio di caratteri per parola. Questo modello può essere visto come un’approssimazione delle misure tradizionali di leggibilità, in particolare dell’indice Gulpease (Piemontese, Lucisano, 1988), un indice specificamente concepito per la lingua italiana, che considera due variabili linguistiche: la lunghezza della

10 www.ilc.cnr.it/dylanlab/apps/texttools/?tt_user=guest

parola e la lunghezza della frase rispetto al numero delle lettere. Formula:

$$89 + \frac{300 * (numero\ delle\ frasi) - 10 * (numero\ delle\ lettere)}{numero\ delle\ parole}$$

i risultati sono compresi tra 0 e 100, dove il valore "100" indica la leggibilità più alta e "0" la leggibilità più bassa. In generale risulta che testi con un indice inferiore a 80 sono difficili da leggere per chi ha la licenza elementare, con un indice inferiore a 60 sono difficili da leggere per chi ha la licenza media, con un indice inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

- Dylan LESSICALE: questo modello si focalizza sulle caratteristiche lessicali del testo, costituite dalla composizione del vocabolario così come dalla sua ricchezza lessicale.
- Dylan SINTATTICO: questo modello si basa su un'informazione di tipo grammaticale, ovvero sulla combinazione di tratti morfo-sintattici e sintattici desunti dai corrispondenti livelli di analisi linguistica.
- Dylan GLOBALE: si tratta di un modello basato sulla combinazione di tratti di varia natura, che spaziano dalle caratteristiche generali del testo del modello Dynal BASE a quelle lessicali e sintattiche degli altri due modelli.

Per ciascun modello, la percentuale esprime il livello di difficoltà, ovvero si riferisce alla probabilità di appartenenza del testo in esame alla classe dei testi di difficile leggibilità: la barra a fianco esprime visivamente questo valore, dove il rosso rappresenta la probabilità di appartenenza alla classe dei testi difficili e il verde a quelli di facile lettura.
Nell'output del tool viene riportata un'altra sigla: VdB. VdB indica il *Vocabolario di base*, una risorsa lessicale della lingua italiana, creata dal linguista Tullio de Mauro, che comprende circa 7.000 parole, quelle che hanno la maggiore frequenza statistica di utilizzo nella nostra lingua. Sono quelle che più usiamo, che più ci sono familiari.
Il vocabolario di base si divide in:

1. Vocabolario fondamentale, composto da 1.991 parole. Sono le più usate in assoluto

nella nostra lingua (esempi: amore, lavoro, pane).

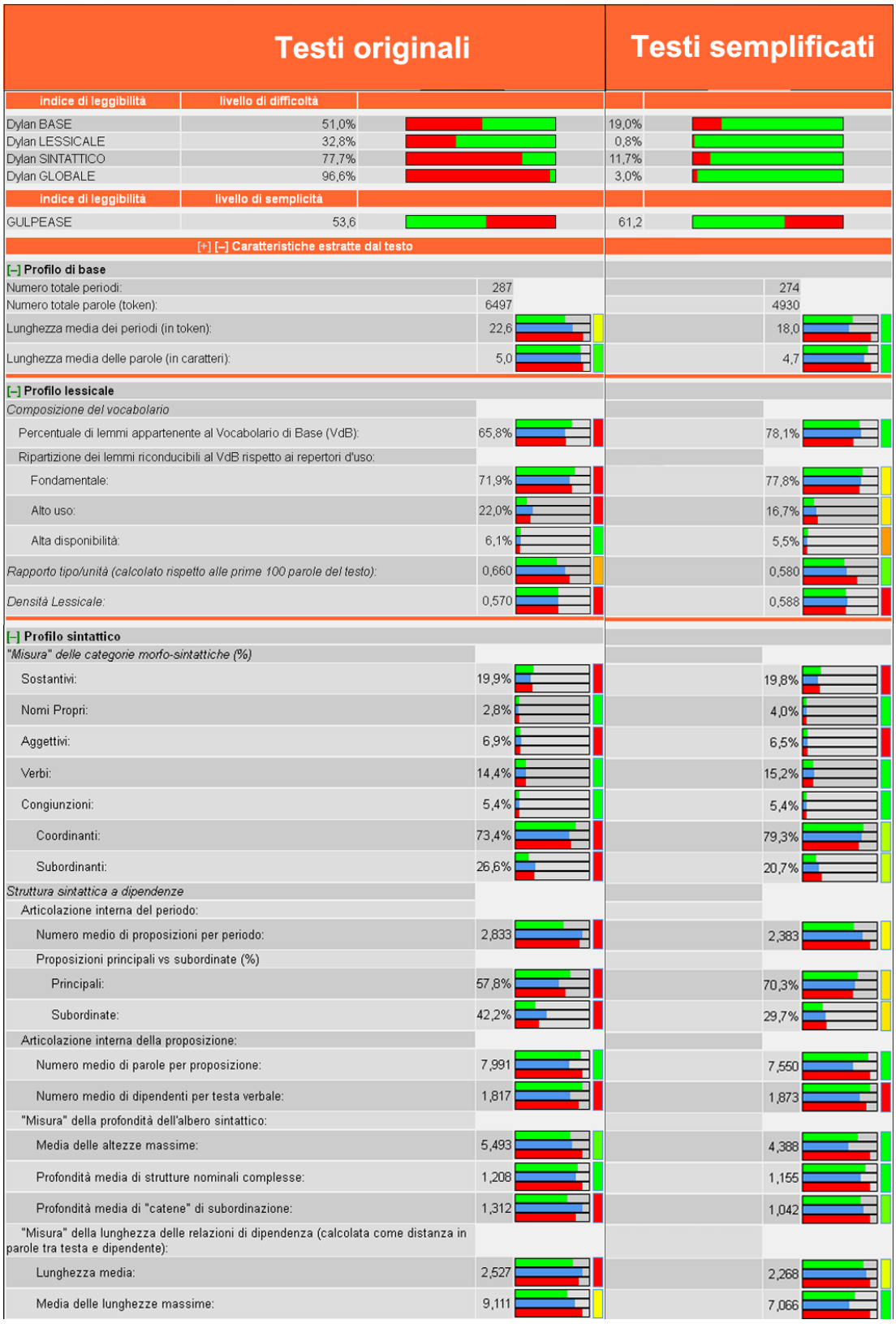
2. Vocabolario di alto uso, composto da 2.750 parole. Sono molto usate, ma meno di quelle del Vocabolario fondamentale (esempi: palo, seta, toro).

3. Vocabolario di alta disponibilità, composto da 2.337 parole. Sono poco usate nella lingua scritta, ma molto in quella parlata (esempi: mensa, lacca, tuta).

Infine, nel modello sintattico, vengono considerate proprietà che caratterizzano l'albero sintattico di una frase come ad esempio la media delle altezze massime, la profondità media di strutture nominali complesse (cioè il numero di modificatori che dipendono da un nome testa della dipendenza) e la profondità media di catene di subordinazione.

Tornando all'analisi della semplificazione, prima operazione che è stata svolta è il confronto tra tutti i testi originali e i semplificati, per verificare se effettivamente gli indici in READ-IT intercettano gli interventi di semplificazione.

Figura 5. output di READ-IT riguardante il confronto tra tutti i testi originale e semplificati



	Testi originali	Testi semplificati
indice di leggibilità	**livello di difficoltà**	
Dylan BASE	51,0%	19,0%
Dylan LESSICALE	32,8%	0,8%
Dylan SINTATTICO	77,7%	11,7%
Dylan GLOBALE	96,6%	3,0%
indice di leggibilità	**livello di semplicità**	
GULPEASE	53,6	61,2
[+] [–] Caratteristiche estratte dal testo		
[–] Profilo di base		
Numero totale periodi:	287	274
Numero totale parole (token):	6497	4930
Lunghezza media dei periodi (in token):	22,6	18,0
Lunghezza media delle parole (in caratteri):	5,0	4,7
[–] Profilo lessicale		
Composizione del vocabolario		
Percentuale di lemmi appartenente al Vocabolario di Base (VdB):	65,8%	78,1%
Ripartizione dei lemmi riconducibili al VdB rispetto ai repertori d'uso:		
Fondamentale:	71,9%	77,8%
Alto uso:	22,0%	16,7%
Alta disponibilità:	6,1%	5,5%
Rapporto tipo/unità (calcolato rispetto alle prime 100 parole del testo):	0,660	0,580
Densità Lessicale:	0,570	0,588
[–] Profilo sintattico		
"Misura" delle categorie morfo-sintattiche (%)		
Sostantivi:	19,9%	19,8%
Nomi Propri:	2,8%	4,0%
Aggettivi:	6,9%	6,5%
Verbi:	14,4%	15,2%
Congiunzioni:	5,4%	5,4%
Coordinanti:	73,4%	79,3%
Subordinanti:	26,6%	20,7%
Struttura sintattica a dipendenze		
Articolazione interna del periodo:		
Numero medio di proposizioni per periodo:	2,833	2,383
Proposizioni principali vs subordinate (%)		
Principali:	57,8%	70,3%
Subordinate:	42,2%	29,7%
Articolazione interna della proposizione:		
Numero medio di parole per proposizione:	7,991	7,550
Numero medio di dipendenti per testa verbale:	1,817	1,873
"Misura" della profondità dell'albero sintattico:		
Media delle altezze massime:	5,493	4,388
Profondità media di strutture nominali complesse:	1,208	1,155
Profondità media di "catene" di subordinazione:	1,312	1,042
"Misura" della lunghezza delle relazioni di dipendenza (calcolata come distanza in parole tra testa e dipendente):		
Lunghezza media:	2,527	2,268
Media delle lunghezze massime:	9,111	7,066

Ciò che si osserva è fondamentalmente la diminuzione dei valori, in molteplici casi, nella versione semplificata.

Tabella 1. Percentuali Dylan, nell'ordine testi originali e semplificati

Dylan BASE	51,0%	19,0%
Dylan LESSICALE	32,8%	0,8%
Dylan SINTATTICO	77,7%	11,7%
Dylan GLOBALE	96,6%	3,0%
GULPEASE	53,6	61,2

L'indice più significativo di questa tabella è il Dylan lessicale, il quale riesce ad approssimare quasi a 0 la sua percentuale, motivo per cui il livello di difficoltà di lettura decrementa maggiormente. Non da trascurare anche il Dylan Globale che mostra per i testi semplificati una percentuale di ben 3,0% per i testi semplificati, contro 96,6% dei testi originali.

Si può dedurre inoltre a chi siano rivolti tutti i testi semplificati, grazie all'indice di Gulpease: l'indice di 53,6 implica che i testi originali siano rivolti a ragazzi compresi tra le scuole medie e le scuole superiori, mentre 61,2 implica che il target delle insegnanti sia complessivamente la scuola media.

Si può osservare nel profilo di base come il numero dei periodi si abbassi rispetto ai testi originali (274 contro 287), oppure come il numero di token diminuisca di ben 1567 tokens nella versione semplificata. Ad influire ulteriormente sono anche la media delle dimensioni dei periodi e dei tokens i quali diminuiscono sempre nel semplificato (periodi: 18,0 contro 22,6; tokens: 4,7 contro 5,0).

Nel profilo lessicale si possono osservare alcuni elementi relativi al vocabolario di base e la ripartizione di diversi lemmi[11]. Complessivamente si può osservare come la percentuale dei lemmi del vocabolario di base sia superiore nei testi semplificati (78,1% contro 65,8 nei testi originali); i lemmi ad uso fondamentale sono il 77,8% nella versione semplificata e 71,9% nella versione originale, questo significa che la

11 In linguistica, e in particolare in morfologia, il lemma costituisce la forma canonica di una parola. Il rapporto fra lemmi e parole è particolarmente importante nelle lingue dotate di un ricco paradigma flessivo delle parole. Tipicamente il lemma è la parola di ricerca del dizionario.

ricorrenza delle parole “fondamentali” all'interno dei testi in formato originale è minore rispetto alla versione semplificata.

Il rapporto tipo/unita (*Type/Token Ratio* abbreviata come TTR) è un indice di ricchezza lessicale, calcolato come il numero di parole tipo, o vocabolario[12], diviso il numero di tutte le parole del testo. Il quoziente è sempre compreso tra 0 e 1; se si approssima a 0 significa che il testo non è molto vario lessicalmente, mentre se si approssima a 1 significa che è molto vario; se il quoziente equivale a uno (caso rarissimo ma non impossibile), significa che il testo non presenta parole ripetute.

Nei testi analizzati si può riscontrare una TTR per i testi originali di 0,660, e di 0,580 per i testi semplificati; questo indica che le insegnanti, autrici delle semplificazioni, hanno volutamente ripetuto alcune parole, all'interno dei testi semplificati, proprio per semplificare la lettura di essi. Non a caso troviamo a destra un valore diminuito di ben 0,08, che per questi range di punteggi è un valore molto significativo.

La densità lessicale è un indice che caratterizza variazioni di registro linguistico e viene calcolata come il rapporto tra il numero totale di occorrenze nel testo di sostantivi, verbi, avverbi, aggettivi, e il numero totale di parole nel testo, ad esclusione dei segni di punteggiatura (Dell'Orletta, 2012-2013). In questo caso si riscontrano dei valori al limite dell'equivalenza: 0,570 per i testi originali, e 0,588 per i testi semplificati; questo indica una leggera variazione del registro linguistico di 0,018 nei testi semplificati. Plausibile dato che la variazione delle opzioni di semplificazione variano in base alla persona che adotta le semplificazioni.

Nel profilo sintattico, e in un primo luogo rispetto alle categorie morfosintattiche, possiamo osservare gli aumenti e le diminuzioni dei valori relativi agli elementi del testo più importanti tra cui i nomi propri che nei corpora semplificati hanno una percentuale di 4,0% e negli originali di 2,8%; implicazione del fatto che molti nomi propri vengono ripetuti nella versione semplificata per renderli più salienti al lettore: un indice di semplicità da non trascurare. Un'analisi simile si può fare nei confronti delle congiunzioni coordinanti che hanno uno scarto del 5,9% (79,3% per i testi semplificati e 73,4 per quanto concerne i testi originali).

Nelle articolazioni dei periodi possiamo trovare un'ascesa della percentuale di

12 Il vocabolario di un testo, è l'insieme di tutte le parole contate una sola volta, ovvero l'insieme delle parole “tipo”; da differenziare con la definizione di parole “unità” che sono tutte le parole del testo.

utilizzo delle proposizioni principali nei corpora semplificati di 12,5 (70,3 contro 57,8 negli originali), ed un abbassamento delle proposizioni subordinate sempre di 12,5 nei corpora semplificati. Le misure coincidono perfettamente perché quelle che prima erano proposizioni subordinate, dopo diventano proposizioni principali. All'interno dell'articolazione delle proposizione si può trovare un abbassamento di 0,441 del numero medio di parole per proposizione (7,991 nei testi originali e 7,550 nei testi semplificati).

La media delle profondità degli alberi sintattici[13] nei corpora originali si abbassa di ben 1,105 (5,493 nei corpora originali e 4,388 nei corpora semplificati); la leggibilità di una frase con alberi più corti è molto più semplice rispetto ad una frase lunga e dotata di molte articolazioni. Infatti la misura media delle catene di subordinazione, nei testi semplificati, cala di ben 0,27 (1,312 per i testi originali e 1,042 per i testi semplificati).

Si può osservare anche un abbassamento della media delle lunghezze massime delle relazioni di dipendenza, calcolata come distanza in parole tra la testa (verbo della proposizione principale) e l'ultima parola della dipendente; l'abbassamento è di 2,045 (9,111 per i corpora originai e 7,066 per i corpora semplificati).

3.4 Osservazioni per regole raggruppate

Sono state osservate le più importanti regole su tutti i sotto-corpora messi a confronti, originali e semplificati. Si può fare un altro tipo di confronto selezionando nei sotto-corpora originali le frasi contenenti determinati tipi di tags[14].

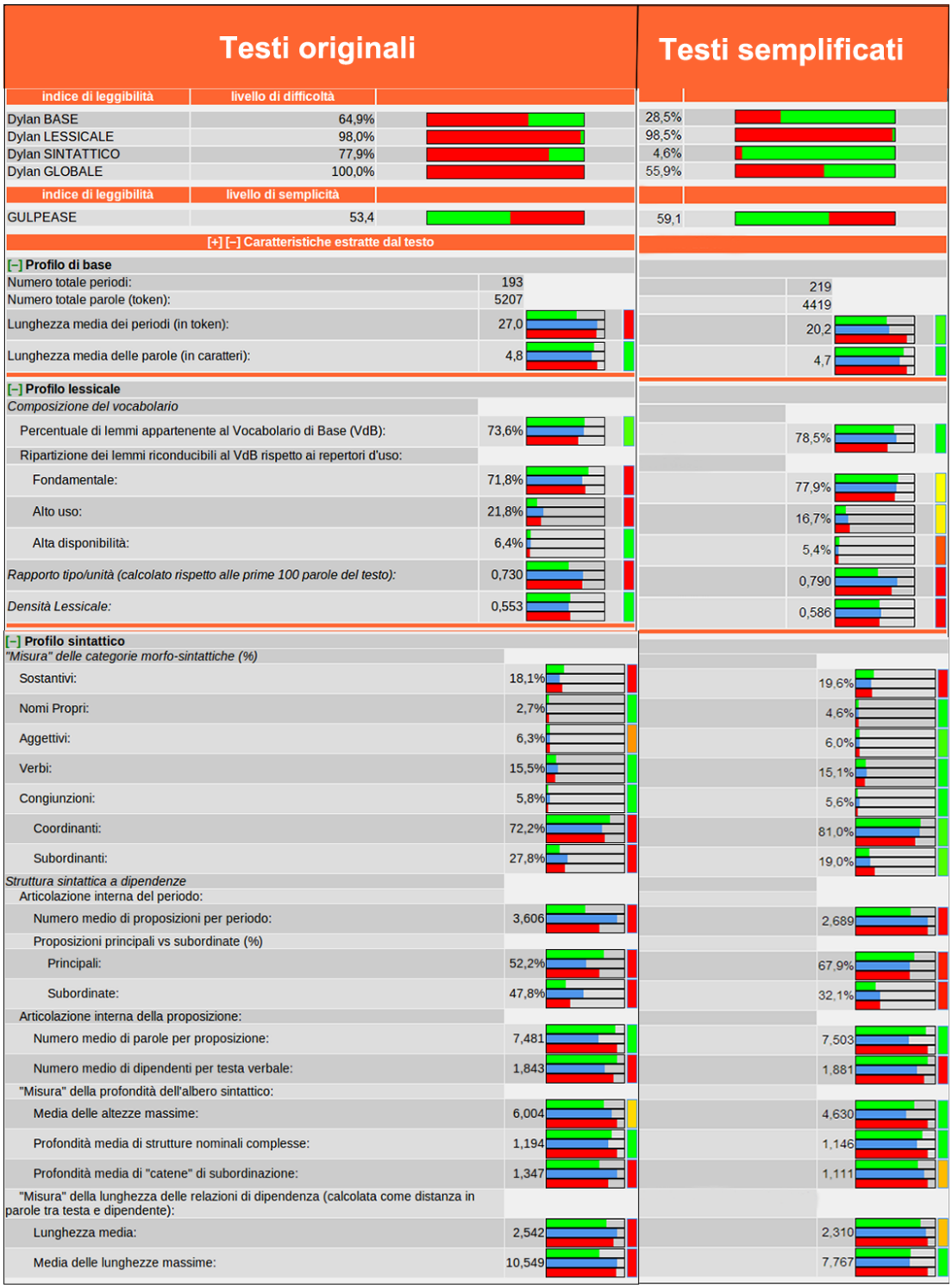
Tra le molteplici estrazioni statistiche eseguite, l'analisi si è concentrata su due combinazioni di regole. Una combinazione prende in considerazione quelle che riducono la lunghezza media della frase, o perché la dividono (*split*) o perché ne eliminano qualche elemento (*delete*, *verbo_meno*); per verificare se anche gli altri tags hanno un effetto sulla leggibilità, sono state selezionate le frasi che non contengono le regole descritte prima ma tutte le altre (quindi, *verbo_piu*, *sogg_espl*,

13 l'albero sintattico in un'annotazione sintattica a dipendenze rappresenta il numero di archi che intercorrono tra una foglia (rappresentata da parole del testo senza dipendenti) e la radice (root) dell'albero.

14 Per estrarre frasi contenenti regole in un file XML è stato utilizzato un programmato un codice Python, reperibile in appendice.

insert, *merge*, *spostamento*, *sogg_sott*, *sost_lex*, *anafora*, *nominalizzazione_piu*, *nominalizzazione_meno*, *pass_attivo*, *att_passivo*), in modo che si possa osservare le variazioni del testo.

Figura 6. risultato del confronto in READ-IT delle frasi originali, e delle frasi semplificate alle quali sono state applicate le regole *split*, *delete*, e *verbo_meno*.



Testi originali		Testi semplificati
indice di leggibilità	**livello di difficoltà**	
Dylan BASE	64,9%	28,5%
Dylan LESSICALE	98,0%	98,5%
Dylan SINTATTICO	77,9%	4,6%
Dylan GLOBALE	100,0%	55,9%
indice di leggibilità	**livello di semplicità**	
GULPEASE	53,4	59,1
[+] [–] Caratteristiche estratte dal testo		
[–] Profilo di base		
Numero totale periodi:	193	219
Numero totale parole (token):	5207	4419
Lunghezza media dei periodi (in token):	27,0	20,2
Lunghezza media delle parole (in caratteri):	4,8	4,7
[–] Profilo lessicale		
Composizione del vocabolario		
Percentuale di lemmi appartenente al Vocabolario di Base (VdB):	73,6%	78,5%
Ripartizione dei lemmi riconducibili al VdB rispetto ai repertori d'uso:		
Fondamentale:	71,8%	77,9%
Alto uso:	21,8%	16,7%
Alta disponibilità:	6,4%	5,4%
Rapporto tipo/unità (calcolato rispetto alle prime 100 parole del testo):	0,730	0,790
Densità Lessicale:	0,553	0,586
[–] Profilo sintattico		
"Misura" delle categorie morfo-sintattiche (%)		
Sostantivi:	18,1%	19,6%
Nomi Propri:	2,7%	4,6%
Aggettivi:	6,3%	6,0%
Verbi:	15,5%	15,1%
Congiunzioni:	5,8%	5,6%
Coordinanti:	72,2%	81,0%
Subordinanti:	27,8%	19,0%
Struttura sintattica a dipendenze		
Articolazione interna del periodo:		
Numero medio di proposizioni per periodo:	3,606	2,689
Proposizioni principali vs subordinate (%)		
Principali:	52,2%	67,9%
Subordinate:	47,8%	32,1%
Articolazione interna della proposizione:		
Numero medio di parole per proposizione:	7,481	7,503
Numero medio di dipendenti per testa verbale:	1,843	1,881
"Misura" della profondità dell'albero sintattico:		
Media delle altezze massime:	6,004	4,630
Profondità media di strutture nominali complesse:	1,194	1,146
Profondità media di "catene" di subordinazione:	1,347	1,111
"Misura" della lunghezza delle relazioni di dipendenza (calcolata come distanza in parole tra testa e dipendente):		
Lunghezza media:	2,542	2,310
Media delle lunghezze massime:	10,549	7,767

L'effetto riguardante la variazione delle caratteristiche di leggibilità in questo caso riguarda principalmente i periodi che aumentano di 26 nei testi semplificati a causa

dell'effetto del tag *split* (219 periodi nella versione semplificata e 193 nella versione originale dei testi), il quale ha spezzato diverse frasi e di conseguenza aumentato il numero di esse. Le conseguenze delle divisioni della frase si riscontrano anche nell'abbassamento della lunghezza media dei periodi in termini di tokens nei testi semplificati, per un valore di 20,2 contro 27,0 negli originali (6,8 di scarto). Ovviamente questo ha un forte effetto sulla semplificazione delle frasi. Le frasi originali hanno una complessità del 100% mentre la loro versione semplificata 55,9%, considerando l'indice READ-IT globale, mentre considerando GULPEASE si passa da 53,4 a 59,1 (originale e semplificato).
Oltre alla diminuzione delle lunghezze medie delle frasi, all'interno di questo "rimpicciolimento" si può notare la differenza di utilizzo delle diverse congiunzioni.

Tabella 2. differenze di percentuale tra le diverse congiunzioni, nell'ordine testi originali e semplificati

Coordinanti	72,2%	81,0%
Subordinanti	27,8%	19,0%

La coordinazione all'interno di alcune frasi piccole aumenta di ben 8,8%, mentre la subordinazione cala dello stesso valore: 8,8%. ancora una volta l'effetto del tag *split* è stato riscontrato fra questi calcoli matematici.
Sempre nel profilo sintattico del tool possiamo riscontrare un abbassamento del numero medio di proposizioni per periodo di 0,917, quasi un periodo meno per le frasi semplificate (2,689 contro 3,606).

Tabella 3.confronto delle percentuali di frasi subordinate e principali, nell'ordine testi originali e semplificati

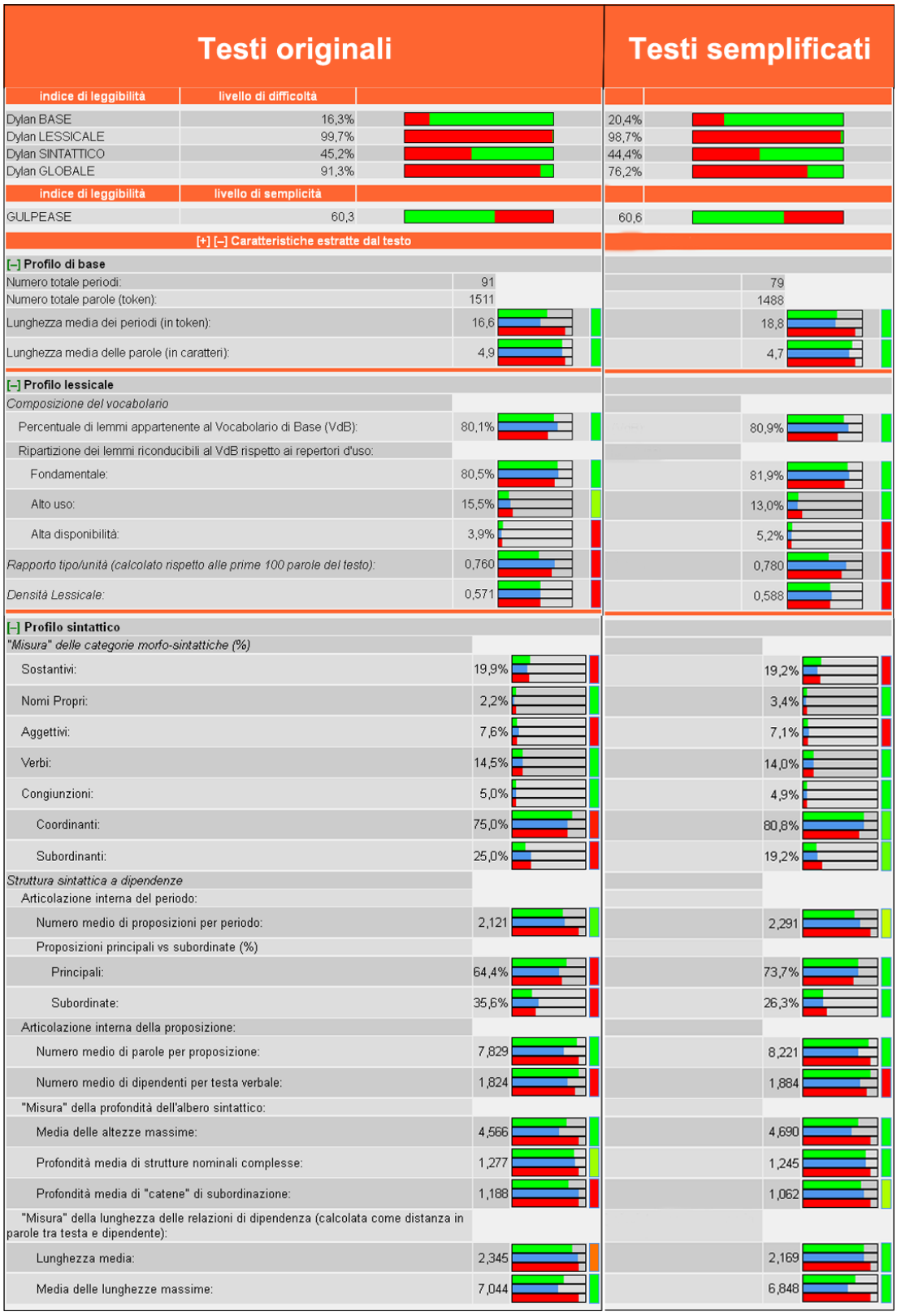
Principali	52,2%	67,9%
Subordinate	47,8%	32,1%

Lo scarto di entrambe le coppie di valori mostrate è equivalente: 15,7%; questo implica che le proposizioni principali dei corpora semplificati aumentano influendo anche sulla diminuzione delle proposizioni subordinati (sempre nel semplificato).

Un altro interessante indice di osservazione è la misura media della lunghezza delle relazioni di dipendenza (distanza in parole tra testa e dipendente) la quale conta uno scarto di 2,782 (10,549 nei testi originali e 7,767 nei testi semplificati).

Spostando l'attenzione su tutte le altre regole, escluse ovviamente *split*, *delete* e *verbo_meno*, si può fare un'osservazione opposta alla precedente.

Figura 7. risultato del confronto in READ-IT delle frasi originali, e delle frasi semplificate alle quali non sono state applicate le regole *split*, *delete*, e *verbo_meno*.



Testi originali		Testi semplificati	
indice di leggibilità	**livello di difficoltà**		
Dylan BASE	16,3%	20,4%	
Dylan LESSICALE	99,7%	98,7%	
Dylan SINTATTICO	45,2%	44,4%	
Dylan GLOBALE	91,3%	76,2%	
indice di leggibilità	**livello di semplicità**		
GULPEASE	60,3	60,6	
[+] [–] Caratteristiche estratte dal testo			
[–] Profilo di base			
Numero totale periodi:	91	79	
Numero totale parole (token):	1511	1488	
Lunghezza media dei periodi (in token):	16,6	18,8	
Lunghezza media delle parole (in caratteri):	4,9	4,7	
[–] Profilo lessicale			
Composizione del vocabolario			
Percentuale di lemmi appartenente al Vocabolario di Base (VdB):	80,1%	80,9%	
Ripartizione dei lemmi riconducibili al VdB rispetto ai repertori d'uso:			
Fondamentale:	80,5%	81,9%	
Alto uso:	15,5%	13,0%	
Alta disponibilità:	3,9%	5,2%	
Rapporto tipo/unità (calcolato rispetto alle prime 100 parole del testo):	0,760	0,780	
Densità Lessicale:	0,571	0,588	
[–] Profilo sintattico			
"Misura" delle categorie morfo-sintattiche (%)			
Sostantivi:	19,9%	19,2%	
Nomi Propri:	2,2%	3,4%	
Aggettivi:	7,6%	7,1%	
Verbi:	14,5%	14,0%	
Congiunzioni:	5,0%	4,9%	
Coordinanti:	75,0%	80,8%	
Subordinanti:	25,0%	19,2%	
Struttura sintattica a dipendenze			
Articolazione interna del periodo:			
Numero medio di proposizioni per periodo:	2,121	2,291	
Proposizioni principali vs subordinate (%)			
Principali:	64,4%	73,7%	
Subordinate:	35,6%	26,3%	
Articolazione interna della proposizione:			
Numero medio di parole per proposizione:	7,829	8,221	
Numero medio di dipendenti per testa verbale:	1,824	1,884	
"Misura" della profondità dell'albero sintattico:			
Media delle altezze massime:	4,566	4,690	
Profondità media di strutture nominali complesse:	1,277	1,245	
Profondità media di "catene" di subordinazione:	1,188	1,062	
"Misura" della lunghezza delle relazioni di dipendenza (calcolata come distanza in parole tra testa e dipendente):			
Lunghezza media:	2,345	2,169	
Media delle lunghezze massime:	7,044	6,848	

In questo caso il numero delle frasi passa da 91 nei testi originali a 79 nei testi semplificati. Questo indica un utilizzo del tag *merge* che ha fatto sì che il numero dei periodi nella versione semplificata sia stato molto ridotto di ben 12 periodi, dato che concatena una o più frasi, es:

Frasi originali: “Le leggi antisemitiche si susseguivano l’una all’altra. Gli ebrei debbono portare la stella giudaica.”

Frase semplificata: “A causa delle leggi antisemitiche gli ebrei devono portare la stella giudaica.”

Inoltre questa riduzione delle frasi ha effetto nella lunghezza media delle stesse che passa da 16,6 a 18,8. Nonostante ad un aumento medio della lunghezza delle frasi di solito corrisponda un aumento della complessità delle stesse, in questo caso l'effetto delle regole di semplificazione applicate ci permette di passare da una complessità di 91,3 a 76,2, quindi sebbene la lunghezza media delle frasi sia aumentata, il testo risulta semplificato. Questa semplificazione non è intercettata dall'indice GULPEASE che sfrutta per il calcolo della complessità del testo, solo caratteristiche superficiali della frase. Mente invece READ-IT è in grado di intercettare parametri linguistici più complessi come ad esempio il rapporto di uguaglianza tra gli scarti delle percentuali delle proposizioni principali e delle proposizioni subordinate.

Tabella 4. confronto tra le percentuali delle frasi subordinate e principali

Proposizioni	**Testi Originali**	**Testi Semplificati**
Principali	64,4%	73,7%
Proposizioni Subordinate	35,6%	26,3%

L'uguaglianza degli scarti del 9,3% dimostra che, che quelle proposizioni che nella versione originale erano in precedenza subordinate diventano principali nella versione semplificata.

Facendo un passo indietro per osservare alcuni tokens del testo si può osservare che esiste un'altra uguaglianza di scarti, ovvero la differenza fra le congiunzioni coordinanti e subordinanti.

Tabella 5. confronto e differenza delle percentuali delle congiunzioni coordinanti e subordinanti nei rispettivi testi

Congiunzioni	**Testi Originali**	**Testi Semplificati**	**Scarto**
Coordinanti	75,0%	80,8%	5,8%
Subordinanti	25,0%	19,2%	5,8%

Quelle stesse congiunzioni subordinanti che davano vita a delle proposizioni subordinanti si trasformano precisamente in congiunzioni coordinanti, e lo scarto del 5,8% ne è la riprova.

Un altro interessante parametro è la lunghezza media dei link sintattici che diminuisce nonostante le frasi aumentino mediamente di lunghezza. Questo può essere conseguenza di regole come ad esempio lo spostamento, che può essere usato per ridurre lo spazio tra due parole sintatticamente in relazione.

3.5 Conclusione

In questo lavoro è stato affrontato un tema molto attuale e ancora poco esplorato in linguistica computazionale che è quello della semplificazione automatica del testo. È stata sottolineata l'importanza di creare una risorsa adeguata al tipo di compito, che più nel dettaglio, ha affrontato l'aspetto della semplificazione “intuitiva”. Per questo è stato costituito un corpus che abbiamo qui definito come corpus “parallelo monolingue”. Va sottolineata la difficoltà di reperire testi totalmente allineati (ovvero a livello di singole frasi), dal momento che la produzione di un testo semplificato si inserisce in un contesto di attività più ampio, che include vari interventi previsti dagli insegnanti per facilitare la comprensione in favore di specifici destinatari (studenti con un livello di conoscenza dell'italiano limitato).

Una volta costituito un corpus composto da un numero di testi significativo, la fase successiva è stata l'annotazione, il cui obiettivo è stato quello di intercettare i tipi di semplificazione, sia lessicale che sintattica, attraverso delle regole appositamente predisposte. Successivamente per valutare il peso di ciascuna di queste regole sono stati sviluppati due programmi che hanno consentito di identificare le regole maggiormente produttive: come abbiamo visto, si tratta principalmente di quelle che

intercettano cambiamenti a livello del lessico, eliminano parti di frase o dividono la frase. Per valutare l'effetto delle regole di semplificazione anche da un punto di vista più qualitativo, è stato introdotto nell'analisi il software READ-IT. Questo tool misura la leggibilità del testo sulla base di complesse configurazioni di caratteristiche linguistiche estratte in maniera automatica. Come abbiamo visto dalla prima estrazione, che compara i due corpora nella totalità, READ-IT ha attribuito un punteggio di leggibilità superiore ai testi semplificati. Tuttavia per valutare in maniera più granulare l'effetto delle singole regole di semplificazione sulla leggibilità sono state effettuate due diverse estrazioni. La prima ha considerato solo le regole che riducono la lunghezza media della frase perché la riduzione della lunghezza è chiaramente associata ad una minore difficoltà: nel paragrafo 3.4 si è visto che in effetti l'influenza di queste regole è molto evidente sulla leggibilità, ed è catturata non solo da un modello più di base, ma anche da altri che considerano fattori linguistici più “sofisticati”. La seconda estrazione ha voluto verificare se anche le altre regole avessero un effetto sul miglioramento della leggibilità. In questo caso le previsioni erano meno scontate. Quello che è emerso è che anche regole non implicate nella riduzione del testo hanno comunque un impatto significativo che non viene colto da indici di leggibilità tradizionali (vedi risultati di GULPEASE fig.7) ma che invece READ-IT è riuscito ad intercettare. Questo testimonia l'importanza di un monitoraggio linguistico del testo a livello di complessità crescente.

4. BIBLIOGRAFIA

4.1 Bibliografia primaria

Lenci, Alessandro, Simonetta Montemagni, Vito Pirelli. *Testo e computer – elementi di linguistica computazionale.* Roma, Carocci, 2005.

Brad Dayley. *Python – Codice e comandi essenziali.* Piacenza, Pearson, 2007.

Steven Bird, Ewan Klein, Edward Loper. *Natural Language processing with Python.* A cura di Livio Mondini, Sebastopol, O'Reilly, 2009.

Simonetta Montemagni. *Tecnologie linguistico-computazionali e monitoraggio della lingua italiana"*. In Studi Italiani di Linguistica Teorica e Applicata (SILTA) Anno XLII, Numero 1, pp. 145-172, 2013.

Dell'Orletta, Felice, Simonetta Montemagni, Giulia Venturi. *READ-IT: assessing readability of Italian texts with a view to text simplification*. In: SLPAT '11 – SLPAT '11 Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies (Edimburgo, UK, 30 Luglio 2011). Proceedings, pp. 73 – 83. Association for Computational Linguistics Stroudsburg, PA, USA, 2011.

Tullio De Mauro. *Il dizionario della lingua italiana.* Torino, Paravia, 2000.

Lucisano, Pietro, Maria Emanuela Piemontese. "*GULPEASE: una formula per la predizione della difficoltà dei testi in lingua italiana*", «*Scuola e città*», 3, 31, marzo 1988, La Nuova Italia.

Stefan Bott, Horacio Saggion: *Text simplification resources for Spanish***.** Language Resources and Evaluation 48(1): 93-120 (2014)

4.2 Bibliografia secondaria

4.2.1 Monografie

Maria Ferrari, Elisa Maggi, Franca Marchesi. 2008. *Antologia ITALIANO L2 - Testi d'autore facilitati e semplificati per classi plurilingue*. Bergamo, Sestante.

Tiziano Franzi, Simonetta Damele. *A ciascuno il suo.* A cura di Gabriella Candia, Torino, Loescher, 2010.

4.2.2 Siti web

Wikipedia, voce *Quadro comune europeo di riferimento per la conoscenza delle lingue*
http://it.wikipedia.org/wiki/Quadro_comune_europeo_di_riferimento_per_la_conoscenza_delle_lingue (visitato il 16 Aprile 2014)

Progetto Terence:
http://terenceproject.eu/web/guest/home (visitato il 29 Maggio 2014)

Capire per studiare 3:
http://www.google.it/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0CCAQFjAA&url=http%3A%2F%2Fnuke.istitutocomprensivoroncalli.it%2FLinkClick.aspx%3Ffileticket%3DgYWntxgeATE%253D%26tabid%3D507%26mid%3D1649&ei=p4uyU43_BYrf4QS3u4DYCA&usg=AFQjCNGegjrdr_mkn-f8f_pF6gV-3WgCRQ&sig2=m6aeVfhw0cEhrJuDHETbcQ&bvm=bv.69837884,bs.1,d.ZGU (visitato il 9 maggio 2014)

Percorsi di apprendimento per gli stranieri nella scuola italiana:
http://www.researchgate.net/publication/40783189_Percorsi_di_apprendimento_per_gli_stranieri_nella_scuola_italiana (visitato il 13 Aprile 2014)

Approccio alla lingua italiana per allievi stranieri:
http://www.retetrevisointegrazionealunnistranieri.it/download/laboratoriorete.pdf (visitato il 16 Aprile 2014)

Il mito:
http://www.scuolavicospinea.it/docenti/RISM/public/gruppo%2013.pdf (visitato il 15 Aprile 2014)

Io sono così:
http://www.scuolavicospinea.it/docenti/RISM/public/gruppo%208.pdf (visitato il 17 Aprile 2014)

Il pinocchio di Collodi:
http://www.scuolavicospinea.it/docenti/RISM/public/gruppo%208.pdf (visitato il 28 Aprile 2014)

5. APPENDICI

Principalmente la selezione delle frasi e la selezione di eventuali porzioni di testo, prima dell'effettivo processo di analisi del testo, è avvenuta tramite il linguaggio di programmazione Python[15]; questo linguaggio processa il testo e ne estrae alcune inferenze statistiche, come la distribuzione dei tags, ad esempio; ma nel caso della preparazione ed estrazione di testo dal documento marcato in XML deve essere seguita la procedura seguente:

1. Creazione del programma Python estraente le frasi;
2. Esecuzione del programma tramite riga di comando con indirizzamento in *file.xml;*
3. Modifica del *file.xml* con aggiunta dell'intestazione e di un nodo radice xml
4. Validazione della forma tramite *XML copy Editor;*
5. Creazione di un documento *xsl*[16] per la visualizzazione a schermo del testo puro del *file.xml*, privo di tags;

5.1 Appendice A

Programma estraente regole 1

Uno dei programmi a livello distribuzionale delle regole di semplificazione è il programma *estraiRegole.py.* Come abbiamo visto nel cap.3 viene fornito un output riguardante le frequenze delle regole in ordine decrescente, il numero di frasi alle quali sono state applicate le regole e la frequenza di combinazione in ordine decrescente:

```
# -*- coding: utf-8 -*-
import  sys
import codecs

if len(sys.argv)<2:
    print "python programma_estraiRegole.py nomefile.out"
```

15 Python è un linguaggio di programmazione sviluppato all'inizio degli anni '90. È particolarmente adatto allo sviluppo di sistemi per il trattamento di dati testuali, e molte librerie e molte funzioni di base sono già presenti nelle librerie standard del linguaggio

16 Acronimo di eXtensible Stylesheet Language, è il linguaggio di descrizione dei fogli di stile per i documenti in formato XML.

```
    exit()

def Ordina(dict):
    return sorted(dict.items(), key=lambda x: x[0], reverse=True)  #ordina per chiave

def Ordina1(dict):
    return sorted(dict.items(), key=lambda x: x[1], reverse=True)   #ordina per
valore

def main(file1):
    fileRegole=codecs.open(file1, "r", "utf8")
    Frasi_sost_lex={}
    Frasi_delete={}
    Frasi_split={}
    Frasi_merge={}
    Frasi_sogg_espl={}
    Frasi_sogg_sott={}
    Frasi_anafora={}
    Frasi_verbo_piu={}
    Frasi_verbo_meno={}
    Frasi_pass_attivo={}
    Frasi_att_passivo={}
    Frasi_spostamento={}
    Frasi_insert={}
    Frasi_tratti_verbo={}
    Frasi_nominalizzazione_piu={}
    Frasi_nominalizzazione_meno={}
    FRASI={}
    REGOLE={}
    REGOLE_TOT={'sost_lex':0.0,'delete':0.0,
'split':0.0,'merge':0.0,'sogg_espl':0.0,'sogg_sott':0.0,'anafora':0.0,'verbo_piu':0.0
,'verbo_meno':0.0,'tratti_verbo':0.0,'pass_attivo':0.0,'att_passivo':0.0,'spostamento
':0.0,'insert':0.0, 'nominalizzazione_piu':0.0,'nominalizzazione_meno':0.0}
    numeroFrase=0.0
    for l in fileRegole:
        REGOLE={}
        if not ("doc" in l[:5]):
            numeroFrase+=1
        if l =="":
            break
        lS=l.strip().split(" ")
        for tok in lS:
            if "sost_lex" in tok and not("/sost_lex" in tok):
                REGOLE["sost_lex"]=1
                REGOLE_TOT["sost_lex"]+=1
                Frasi_sost_lex[numeroFrase]=1
            if "delete" in tok and not("/delete" in tok):
                REGOLE_TOT["delete"]+=1
                REGOLE["delete"]=1
```

```
                Frasi_delete[numeroFrase]=1
            if "split" in tok and not("/split" in tok):
                REGOLE_TOT["split"]+=1
                REGOLE["split"]=1
                Frasi_split[numeroFrase]=1
            if "merge" in tok and not("/merge" in tok):
                REGOLE_TOT["merge"]+=1
                REGOLE["merge"]=1
                Frasi_merge[numeroFrase]=1
            if "sogg_espl" in tok:
                REGOLE_TOT["sogg_espl"]+=1
                REGOLE["sogg_espl"]=1
                Frasi_sogg_espl[numeroFrase]=1
            if "sogg_sott" in tok and not("/sogg_sott" in tok):
                REGOLE_TOT["sogg_sott"]+=1
                REGOLE["sogg_sott"]=1
                Frasi_sogg_sott[numeroFrase]=1
            if "anafora" in tok and not("/anafora" in tok):
                REGOLE_TOT["anafora"]+=1
                REGOLE["anafora"]=1
                Frasi_anafora[numeroFrase]=1
            if "verbo_piu" in tok and not("/verbo_piu" in tok):
                REGOLE_TOT["verbo_piu"]+=1
                REGOLE["verbo_piu"]=1
                Frasi_verbo_piu[numeroFrase]=1
            if "verbo_meno" in tok and not("/verbo_meno" in tok):
                REGOLE_TOT["verbo_meno"]+=1
                REGOLE["verbo_meno"]=1
                Frasi_verbo_meno[numeroFrase]=1
            if "pass_attivo" in tok and not("/pass_attivo" in tok) or "pass_att" in
tok and not("/pass_att" in tok):
                REGOLE_TOT["pass_attivo"]+=1
                REGOLE["pass_attivo"]=1
                Frasi_pass_attivo[numeroFrase]=1
            if "att_passivo" in tok and not("/att_passivo" in tok):
                REGOLE_TOT["att_passivo"]+=1
                REGOLE["att_passivo"]=1
                Frasi_att_passivo[numeroFrase]=1
            if "spostamento" in tok and not("/spostamento" in tok):
                REGOLE_TOT["spostamento"]+=1
                REGOLE["spostamento"]=1
                Frasi_spostamento[numeroFrase]=1
            if "insert" in tok:
                REGOLE_TOT["insert"]+=1
                REGOLE["insert"]=1
                Frasi_insert[numeroFrase]=1
            if "tratti_verbo" in tok and not("/tratti_verbo" in tok):
                REGOLE_TOT["tratti_verbo"]+=1
                REGOLE["tratti_verbo"]=1
```

```
                Frasi_tratti_verbo[numeroFrase]=1
            if "nominalizzazione_piu" in tok and not("/nominalizzazione_piu" in tok):
                REGOLE_TOT["nominalizzazione_piu"]+=1
                REGOLE["nominalizzazione_piu"]=1
                Frasi_nominalizzazione_piu[numeroFrase]=1
            if "nominalizzazione_meno" in tok and not("/nominalizzazione_meno" in
tok):
                REGOLE_TOT["nominalizzazione_meno"]+=1
                REGOLE["nominalizzazione_meno"]=1
                Frasi_nominalizzazione_meno[numeroFrase]=1
        REGOLE_ORDINATE=Ordina(REGOLE)
        stringa=""
        if REGOLE:     #se hai applicato almeno una regola
            for regola in REGOLE_ORDINATE:
                stringa+=regola[0]+" "
            stringa=stringa.strip()     #contiene la combinazione di tutte le regole
applicate ad una frase
            if not stringa in FRASI:
                FRASI[stringa]=0
            FRASI[stringa]+=1.0

    print "RISULTATI:\n"
    print "1) Frequenza di applicazione delle REGOLE:"
    for el in Ordina1(REGOLE_TOT):
        print el[0], el[1],"\n"


pr={'tratti_verbo':len(Frasi_tratti_verbo),'sost_lex':len(Frasi_sost_lex),'delete':le
n(Frasi_delete),'split':len(Frasi_split),'merge':len(Frasi_merge),'sogg_espl':len(Fra
si_sogg_espl),'sogg_sott':len(Frasi_sogg_sott),'anafora':len(Frasi_anafora),'verbo_pi
u':len(Frasi_verbo_piu),'verbo_meno':len(Frasi_verbo_meno),'pass_attivo':len(Frasi_pa
ss_attivo),'att_passivo':len(Frasi_att_passivo),'spostamento':len(Frasi_spostamento),
'insert':len(Frasi_insert),'nominalizzazione_meno':len(Frasi_nominalizzazione_meno),'
nominalizzazione_piu':len(Frasi_nominalizzazione_piu)}
    print "2) numero di frasi alle quali sono state applicate le varie REGOLE:"
    for el in Ordina1(pr):
        print el[0],el[1],"\n"

    FRASI_ord=Ordina1(FRASI)
    print "3) Frequenza di combinazione di regole (senza frequenza) nelle frasi:"
    for elem in  FRASI_ord:
        stringa=elem[0]+"\t"+str(elem[1])
        print stringa,"\n"

main(sys.argv[1])
```

Output:

RISULTATI:

1) Frequenza di applicazione delle REGOLE:
sost_lex 495.0

delete 251.0

insert 188.0

tratti_verbo 164.0

spostamento 108.0

anafora 57.0

verbo_piu 41.0

sogg_espl 33.0

split 30.0

merge 23.0

nominalizzazione_meno 15.0

verbo_meno 13.0

pass_attivo 9.0

nominalizzazione_piu 7.0

sogg_sott 2.0

att_passivo 1.0

2) numero di frasi alle quali sono state applicate le varie REGOLE:
sost_lex 206

delete 148

tratti_verbo 110

insert 108

spostamento 86

anafora 47

verbo_piu 32

sogg_espl 32

split 29

merge 23

nominalizzazione_meno 13

verbo_meno 12

pass_attivo 9

nominalizzazione_piu 7

sogg_sott 1

att_passivo 1

3) Frequenza di combinazione di regole (senza frequenza) nelle frasi:
sost_lex 25.0

sost_lex insert delete 11.0

spostamento sost_lex insert delete 8.0

tratti_verbo sost_lex 8.0

sost_lex delete 8.0

spostamento sost_lex delete 6.0

sost_lex insert 6.0

tratti_verbo sost_lex insert delete 5.0

tratti_verbo sost_lex delete 5.0

tratti_verbo delete 5.0

tratti_verbo sost_lex insert 5.0

tratti_verbo 5.0

tratti_verbo spostamento sost_lex 4.0

delete 4.0

merge 3.0

tratti_verbo spostamento anafora 3.0

spostamento sost_lex 3.0

verbo_piu sost_lex insert delete 3.0

tratti_verbo spostamento sost_lex sogg_espl delete anafora 2.0

sost_lex merge insert delete 2.0

spostamento sost_lex delete anafora 2.0

verbo_piu tratti_verbo spostamento split sost_lex insert delete anafora 2.0

tratti_verbo spostamento sogg_espl delete anafora 2.0

tratti_verbo sost_lex merge insert delete anafora 2.0

verbo_piu sost_lex insert 2.0

tratti_verbo spostamento sost_lex insert delete anafora 2.0

tratti_verbo sost_lex delete anafora 2.0

tratti_verbo spostamento split sost_lex delete anafora 2.0

tratti_verbo sost_lex insert delete anafora 2.0

split sost_lex insert delete 2.0

sost_lex sogg_espl 2.0

tratti_verbo split sost_lex sogg_espl insert delete 2.0

verbo_piu spostamento sost_lex delete 2.0

verbo_meno spostamento sost_lex delete 2.0

verbo_meno tratti_verbo sost_lex 2.0

spostamento split sost_lex sogg_espl delete 1.0

verbo_piu tratti_verbo merge insert delete 1.0

verbo_meno sost_lex insert 1.0

verbo_piu spostamento sost_lex nominalizzazione_meno insert delete 1.0

tratti_verbo spostamento split sost_lex sogg_espl insert delete 1.0

tratti_verbo insert delete 1.0

verbo_piu sost_lex sogg_espl nominalizzazione_meno insert delete 1.0

verbo_piu tratti_verbo sost_lex delete 1.0

sost_lex nominalizzazione_piu insert delete 1.0

verbo_meno tratti_verbo sost_lex merge delete 1.0

tratti_verbo sost_lex sogg_espl anafora 1.0

tratti_verbo spostamento sost_lex insert delete 1.0

split sost_lex anafora 1.0

spostamento sost_lex pass_attivo insert 1.0

tratti_verbo spostamento sost_lex sogg_espl pass_attivo nominalizzazione_meno insert delete 1.0

verbo_meno tratti_verbo 1.0

tratti_verbo sogg_espl delete 1.0

split sost_lex sogg_espl insert delete 1.0

sost_lex sogg_sott 1.0

insert 1.0

tratti_verbo spostamento sost_lex delete 1.0

verbo_piu insert delete 1.0

tratti_verbo split sost_lex sogg_espl delete 1.0

tratti_verbo spostamento sost_lex insert 1.0

tratti_verbo sost_lex sogg_espl 1.0

sost_lex merge 1.0

tratti_verbo sost_lex nominalizzazione_piu delete 1.0

verbo_piu verbo_meno tratti_verbo spostamento sost_lex insert delete 1.0

tratti_verbo split sost_lex delete anafora 1.0

spostamento sost_lex pass_attivo nominalizzazione_meno merge insert delete anafora 1.0

tratti_verbo spostamento insert delete anafora 1.0

tratti_verbo spostamento insert anafora 1.0

tratti_verbo spostamento sost_lex nominalizzazione_meno 1.0

spostamento split sost_lex delete anafora 1.0

verbo_piu spostamento split sost_lex insert delete anafora 1.0

verbo_piu tratti_verbo sost_lex insert 1.0

spostamento sost_lex nominalizzazione_meno delete anafora 1.0

spostamento sost_lex sogg_espl nominalizzazione_meno insert delete 1.0

verbo_piu spostamento split sost_lex merge insert 1.0

verbo_piu verbo_meno tratti_verbo spostamento sost_lex nominalizzazione_piu delete 1.0

spostamento split sost_lex insert delete anafora 1.0

spostamento split sost_lex insert delete 1.0

tratti_verbo split sost_lex pass_attivo insert delete anafora 1.0

verbo_meno tratti_verbo sost_lex insert 1.0

tratti_verbo split delete 1.0

spostamento sost_lex pass_attivo nominalizzazione_meno insert delete 1.0

tratti_verbo sogg_espl insert delete 1.0

tratti_verbo spostamento sost_lex insert anafora 1.0

spostamento sost_lex sogg_espl insert delete 1.0

tratti_verbo spostamento split sost_lex sogg_espl nominalizzazione_meno insert delete anafora 1.0

verbo_piu sost_lex sogg_espl nominalizzazione_meno delete 1.0

verbo_piu spostamento split sost_lex 1.0

insert delete 1.0

verbo_meno tratti_verbo sost_lex insert delete 1.0

verbo_meno spostamento split sost_lex insert delete 1.0

tratti_verbo sost_lex sogg_espl delete anafora 1.0

tratti_verbo spostamento sost_lex delete anafora 1.0

sost_lex merge anafora 1.0

verbo_piu tratti_verbo spostamento split sost_lex sogg_espl insert delete 1.0

tratti_verbo spostamento sost_lex merge insert delete 1.0

tratti_verbo split sost_lex delete 1.0

verbo_piu spostamento sost_lex insert 1.0

sost_lex nominalizzazione_meno insert delete 1.0

tratti_verbo spostamento sost_lex sogg_espl insert delete 1.0

spostamento insert delete att_passivo 1.0

sost_lex sogg_espl merge delete 1.0

tratti_verbo sost_lex anafora 1.0

spostamento split sost_lex 1.0

merge insert delete 1.0

verbo_piu sost_lex sogg_espl anafora 1.0

sost_lex anafora 1.0

tratti_verbo sost_lex nominalizzazione_piu nominalizzazione_meno insert anafora 1.0

verbo_piu nominalizzazione_piu merge delete 1.0

verbo_piu tratti_verbo spostamento sost_lex insert 1.0

spostamento sost_lex merge insert 1.0

spostamento sost_lex pass_attivo insert delete 1.0

```
spostamento sost_lex sogg_espl pass_attivo merge delete anafora    1.0

verbo_piu tratti_verbo split sost_lex insert delete  1.0

verbo_piu      1.0

sost_lex nominalizzazione_piu delete  1.0

tratti_verbo sost_lex insert anafora  1.0

tratti_verbo spostamento sost_lex sogg_espl merge insert    1.0

tratti_verbo spostamento sost_lex sogg_espl pass_attivo insert delete      1.0

tratti_verbo spostamento delete      1.0

verbo_piu spostamento sost_lex sogg_espl delete anafora     1.0

verbo_piu tratti_verbo spostamento sost_lex delete anafora  1.0

tratti_verbo sost_lex pass_attivo insert delete     1.0

tratti_verbo sost_lex merge insert delete    1.0

tratti_verbo sost_lex merge delete   1.0

spostamento insert    1.0

tratti_verbo spostamento sost_lex anafora    1.0

verbo_piu split sost_lex nominalizzazione_meno delete       1.0

nominalizzazione_piu delete   1.0

tratti_verbo sogg_espl merge delete  1.0

verbo_piu tratti_verbo spostamento sost_lex insert delete   1.0

sost_lex insert delete anafora       1.0
```

Programma estraente regole 2

Successivamente come abbiamo visto nella sez. 3.2, è stato portato a temine un altro programma, avente delle caratteristiche simili. Il programma estrae le frequenze assolute dei tags per ogni testo:

```
# -*- coding: utf-8 -*-
import sys
import codecs

if len(sys.argv)<2:
 print "python programma.py file_input.txt"
 exit()

def OrdinaVal(dict):
 return sorted(dict.items(), key=lambda x: x[1], reverse=True)

def statistiche_per_testo(fileRegole):
 regole={'sost_lex':0.0,'delete':0.0,
'split':0.0,'merge':0.0,'sogg_espl':0.0,'sogg_sott':0.0,'anafora':0.0,'verbo_piu':0.0
,'verbo_meno':0.0,'tratti_verbo':0.0,'pass_attivo':0.0,'att_passivo':0.0,'spostamento
':0.0,'insert':0.0, 'nominalizzazione_piu':0.0,'nominalizzazione_meno':0.0}
 i=0
 lista=[]
 for l in fileRegole:
 if l =="":
 break
 lS=l.strip().split(" ")
 for tok in lS:
 if "sost_lex" in tok and not("/sost_lex" in tok):
regole["sost_lex"]=regole["sost_lex"]+1
 if "delete" in tok and not("/delete" in tok):regole["delete"]=regole["delete"]+1
 if "split" in tok and not("/split" in tok):regole["split"]=regole["split"]+1
 if "merge" in tok and not("/merge" in tok):regole["merge"]=regole["merge"]+1
 if "sogg_espl" in tok:regole["sogg_espl"]=regole["sogg_espl"]+1
 if "sogg_sott" in tok and not("/sogg_sott" in
tok):regole["sogg_sott"]=regole["sogg_sott"]+1
 if "anafora" in tok and not("/anafora" in tok):regole["anafora"]=regole["anafora"]+1
 if "verbo_piu" in tok and not("/verbo_piu" in
tok):regole["verbo_piu"]=regole["verbo_piu"]+1
 if "verbo_meno" in tok and not("/verbo_meno" in
tok):regole["verbo_meno"]=regole["verbo_meno"]+1
 if "tratti_verbo" in tok and not("/tratti_verbo" in
tok):regole["tratti_verbo"]=regole["tratti_verbo"]+1
                      if "pass_attivo" in tok and not("/pass_attivo" in
tok):regole["pass_attivo"]=regole["pass_attivo"]+1
                      if "spostamento" in tok and not("/spostamento" in
tok):regole["spostamento"]=regole["spostamento"]+1
                      if "insert" in tok:regole["insert"]=regole["insert"]+1
                      if "nominalizzazione_piu" in tok and
not("/nominalizzazione_piu" in
tok):regole["nominalizzazione_piu"]=regole["nominalizzazione_piu"]+1
                      if "nominalizzazione_meno" in tok and
```

```
not("/nominalizzazione_meno" in
tok):regole["nominalizzazione_meno"]=regole["nominalizzazione_meno"]+1
                        if "att_passivo" in tok and not("/att_passivo" in tok) or
"att_pass" in tok and not("/att_pass" in
tok):regole["pass_attivo"]=regole["pass_attivo"]+1
                        if "/doc" in tok:
                                lista.append(regole)
                                regole={'sost_lex':0.0,'delete':0.0,'split':0.0,'merge'
:0.0,'sogg_espl':0.0,'sogg_sott':0.0,'anafora':0.0,'verbo_piu':0.0,'verbo_meno':0.0,'
tratti_verbo':0.0,'pass_attivo':0.0,'att_passivo':0.0,'spostamento':0.0,'insert':0.0,
'nominalizzazione_piu':0.0,'nominalizzazione_meno':0.0}
        return lista

def main(file1):
        corpus=codecs.open(file1, "r", "utf8")
        stat=statistiche_per_testo(corpus)
        i=1
        for el in stat:
                dz=OrdinaVal(el)
                print "\ncorpus #",i,":",dz[0][0],dz[0][1],dz[1][0],dz[1][1],dz[2]
[0],dz[2][1],dz[3][0],dz[3][1],dz[4][0],dz[4][1],dz[5][0],dz[5][1],dz[6][0],dz[6]
[1],dz[7][0],dz[7][1],dz[8][0],dz[8][1],dz[9][0],dz[9][1],dz[10][0],dz[10][1],dz[11]
[0],dz[11][1],dz[12][0],dz[12][1],dz[13][0],dz[13][1],dz[14][0],dz[14][1]
                i=i+1

main(sys.argv[1])
```

Output:

```
corpus # 1 : sost_lex 13.0 delete 4.0 insert 3.0 verbo_piu 2.0 merge 2.0 sogg_sott
2.0 verbo_meno 0.0 nominalizzazione_piu 0.0 anafora 0.0 nominalizzazione_meno 0.0
spostamento 0.0 sogg_espl 0.0 tratti_verbo 0.0 split 0.0 pass_attivo 0.0

corpus # 2 : sost_lex 14.0 insert 3.0 delete 3.0 spostamento 2.0 verbo_piu 2.0
tratti_verbo 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 anafora 0.0
nominalizzazione_meno 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo
0.0

corpus # 3 : sost_lex 14.0 insert 12.0 tratti_verbo 8.0 delete 5.0 spostamento 4.0
verbo_piu 2.0 sogg_espl 2.0 merge 2.0 split 2.0 verbo_meno 1.0 anafora 1.0
nominalizzazione_piu 0.0 nominalizzazione_meno 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 4 : sost_lex 24.0 insert 14.0 spostamento 8.0 verbo_piu 6.0 tratti_verbo 6.0
delete 6.0 sogg_espl 3.0 split 2.0 verbo_meno 1.0 nominalizzazione_piu 1.0 anafora
1.0 nominalizzazione_meno 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 5 : sost_lex 14.0 insert 3.0 tratti_verbo 2.0 delete 2.0
```

nominalizzazione_piu 1.0 anafora 1.0 split 1.0 verbo_meno 0.0 nominalizzazione_meno 0.0 spostamento 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 6 : sost_lex 13.0 insert 4.0 tratti_verbo 4.0 delete 3.0 anafora 2.0 verbo_piu 2.0 spostamento 1.0 sogg_espl 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 nominalizzazione_meno 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 7 : sost_lex 6.0 tratti_verbo 3.0 insert 2.0 delete 2.0 nominalizzazione_piu 1.0 anafora 1.0 nominalizzazione_meno 1.0 verbo_meno 0.0 spostamento 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 8 : sost_lex 16.0 insert 8.0 delete 7.0 tratti_verbo 2.0 verbo_meno 1.0 nominalizzazione_piu 1.0 anafora 1.0 split 1.0 nominalizzazione_meno 0.0 spostamento 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 9 : sost_lex 22.0 insert 10.0 delete 8.0 tratti_verbo 5.0 anafora 2.0 nominalizzazione_meno 2.0 spostamento 2.0 verbo_piu 2.0 merge 2.0 verbo_meno 0.0 nominalizzazione_piu 0.0 sogg_espl 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 10 : sost_lex 13.0 delete 11.0 insert 10.0 spostamento 4.0 verbo_piu 3.0 tratti_verbo 3.0 anafora 1.0 nominalizzazione_meno 1.0 merge 1.0 split 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 sogg_espl 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 11 : sost_lex 11.0 insert 6.0 spostamento 5.0 delete 5.0 verbo_piu 2.0 tratti_verbo 1.0 split 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 anafora 0.0 nominalizzazione_meno 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 12 : sost_lex 6.0 delete 5.0 tratti_verbo 4.0 spostamento 3.0 nominalizzazione_meno 2.0 insert 1.0 split 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 anafora 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 13 : sost_lex 20.0 delete 7.0 insert 6.0 spostamento 4.0 merge 2.0 tratti_verbo 2.0 verbo_meno 1.0 nominalizzazione_meno 1.0 sogg_espl 1.0 split 1.0 pass_attivo 1.0 nominalizzazione_piu 0.0 anafora 0.0 verbo_piu 0.0 sogg_sott 0.0

corpus # 14 : sost_lex 34.0 delete 22.0 insert 9.0 tratti_verbo 9.0 spostamento 7.0 anafora 6.0 verbo_piu 4.0 nominalizzazione_meno 2.0 sogg_espl 2.0 split 2.0 pass_attivo 2.0 merge 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 sogg_sott 0.0

corpus # 15 : tratti_verbo 20.0 sost_lex 16.0 spostamento 5.0 insert 4.0 anafora 4.0 delete 4.0 verbo_meno 0.0 nominalizzazione_piu 0.0 nominalizzazione_meno 0.0 verbo_piu 0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 16 : tratti_verbo 24.0 sost_lex 6.0 anafora 4.0 spostamento 4.0 delete 2.0 insert 1.0 verbo_meno 1.0 verbo_piu 1.0 sogg_espl 1.0 nominalizzazione_piu 0.0 nominalizzazione_meno 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 17 : sost_lex 13.0 spostamento 4.0 insert 3.0 delete 3.0 split 2.0 anafora

```
1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 nominalizzazione_meno 0.0 verbo_piu 0.0
sogg_espl 0.0 merge 0.0 tratti_verbo 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 18 : sost_lex 19.0 delete 12.0 insert 6.0 spostamento 5.0
nominalizzazione_meno 2.0 verbo_piu 2.0 split 2.0 anafora 1.0 sogg_espl 1.0 merge 1.0
pass_attivo 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0 tratti_verbo 0.0 sogg_sott
0.0

corpus # 19 : sost_lex 20.0 insert 18.0 delete 15.0 spostamento 8.0 sogg_espl 5.0
tratti_verbo 4.0 split 4.0 nominalizzazione_meno 3.0 anafora 2.0 verbo_piu 1.0
verbo_meno 0.0 nominalizzazione_piu 0.0 merge 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 20 : sost_lex 42.0 delete 10.0 spostamento 6.0 insert 4.0 split 3.0 anafora
2.0 verbo_piu 2.0 tratti_verbo 2.0 merge 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0
nominalizzazione_meno 0.0 sogg_espl 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 21 : sost_lex 78.0 delete 63.0 tratti_verbo 41.0 insert 25.0 anafora 23.0
spostamento 20.0 sogg_espl 14.0 merge 6.0 split 6.0 verbo_piu 3.0 verbo_meno 2.0
nominalizzazione_meno 1.0 nominalizzazione_piu 0.0 sogg_sott 0.0 pass_attivo 0.0

corpus # 22 : sost_lex 10.0 tratti_verbo 9.0 insert 3.0 delete 3.0 anafora 2.0
spostamento 1.0 sogg_espl 1.0 verbo_meno 0.0 nominalizzazione_piu 0.0
nominalizzazione_meno 0.0 verbo_piu 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo
0.0

corpus # 23 : tratti_verbo 9.0 sost_lex 8.0 insert 5.0 verbo_meno 2.0 spostamento 2.0
verbo_piu 2.0 delete 2.0 nominalizzazione_piu 0.0 anafora 0.0 nominalizzazione_meno
0.0 sogg_espl 0.0 merge 0.0 sogg_sott 0.0 split 0.0 pass_attivo 0.0

corpus # 24 : sost_lex 63.0 delete 47.0 insert 28.0 spostamento 13.0 verbo_piu 5.0
merge 5.0 tratti_verbo 5.0 verbo_meno 4.0 nominalizzazione_piu 3.0 anafora 2.0
sogg_espl 2.0 split 1.0 pass_attivo 1.0 nominalizzazione_meno 0.0 sogg_sott 0.0
```

5.2 Appendice B

Programma estraente frasi contenenti determinate regole 1

Codice di un programma estraente le frasi contenenti al loro interno i tags *delete*, *verbo_meno*, e *split*:

```
# -*- coding: utf-8 -*-
import sys
import codecs
from xml.dom import minidom
```

```
if len(sys.argv)<1:
 print "python parserXml.py nomefile di output"
 exit()


def main(file1):
        #analizza il documento xml in una struttura dom
        xmldoc=minidom.parse(file1)
        #recupera i nodi dalla radice alla struttura
        cNodes=xmldoc.childNodes
        #cerchiamo un nodo tramite il nome, in questo caso mi interessano i tag frase
        frasiList = cNodes[2].getElementsByTagName("frase")
        splitList=[]
        deleteList=[]
        verbo_menoList=[]
        for node in frasiList:
                if(node.getElementsByTagName("split")):
                        for el in node.getElementsByTagName("split"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        splitList.append(var.parentNode.t
oxml())
                                else:
                                        splitList.append(el.parentNode.toxml())
                if (node.getElementsByTagName("delete")):
                        for el in node.getElementsByTagName("delete"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        deleteList.append(var.parentNode.
toxml())
                                else:
                                        deleteList.append(el.parentNode.toxml())
                if (node.getElementsByTagName("verbo_meno")):
                        for el in node.getElementsByTagName("verbo_meno"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
```

```
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        verbo_menoList.append(var.parentN
ode.toxml())
                                else:
                                        verbo_menoList.append(el.parentNode.toxml())
        listaRegole= verbo_menoList + deleteList + splitList
        newList=set(listaRegole)
        for el in newList:
                print el.encode("utf-8"),"\n"

main(sys.argv[1])
```

Programma estraente frasi contenenti determinate regole 2

Successivamente è stato creato un programma “antagonista” di quello appena enunciato, avente la caratteristica di creare una lista di frasi contenenti tutte le regole, escluse *split*, *delete*, *verbo_meno.* Nel programma sono evidenziate in giallo le differenze rispetto al precedente:

```
# -*- coding: utf-8 -*-
import sys
import codecs
from xml.dom import minidom

if len(sys.argv)<1:
 print "python programma.py nomefile di output"
 exit()

def main(file1):
 #analizza il documento xml in una struttura dom
 xmldoc=minidom.parse(file1)
 #recupera i nodi dalla radice alla struttura
 cNodes=xmldoc.childNodes
 #cerchiamo un nodo tramite il nome, in questo caso mi interessano i tag frase
 frasiList = cNodes[2].getElementsByTagName("frase")
 splitList=[]
 deleteList=[]
 verbo_menoList=[]
 tot=[]
        for node in frasiList:
                splitList=[]
```

```
                deleteList=[]
                verbo_menoList=[]
                if(node.getElementsByTagName("split")):
                        for el in node.getElementsByTagName("split"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        splitList.append(var.parentNode.t
oxml())
                                else:
                                        splitList.append(el.parentNode.toxml())
                if (node.getElementsByTagName("delete")):
                        for el in node.getElementsByTagName("delete"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        deleteList.append(var.parentNode.
toxml())
                                else:
                                        deleteList.append(el.parentNode.toxml())
                if (node.getElementsByTagName("verbo_meno")):
                        for el in node.getElementsByTagName("verbo_meno"):
                                if("frase" not in str(el.parentNode)):
                                        boolean=False
                                        var=el.parentNode
                                        while(boolean==False):
                                                if("frase" not in str(var.parentNode)):
                                                        var=var.parentNode
                                                else:
                                                        boolean=True
                                                        verbo_menoList.append(var.parentN
ode.toxml())
                                else:
                                        verbo_menoList.append(el.parentNode.toxml())
                if not(verbo_menoList) and not(deleteList) and not(splitList):
                        tot.append(node.toxml())
        newList=set(tot)
        for el in newList:
                print el.encode("utf-8"),"\n"
```

```
main(sys.argv[1])
```

5.3 Appendice C

Foglio di stile per i documenti XML

Dopo che l'output è stato indirizzato in un file xml tramite riga di comando (*python nomeProgramma.py nomeFileInput.xml > fileOutput.xml*), vengono separati i tags, con le espressioni regolari, da eventuali spazi non utilizzati tra tags e contenuti. Successivamente il file viene dotato di un nodo radice e di un link per un file XSL tale che il contenuto del documento XML possa essere visualizzato a schermo sul browser, privo di tags:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <xsl:apply-templates />
  </xsl:template>
</xsl:stylesheet>
```

Successivamente il contenuto del documento viene copiato in un file plain text, pronto per essere analizzato.