Introduzione

Creare mappe: indicizzazione e orientamento nel testo

Testualità, lettura e scrittura

Lettura e sapere venatorio

Intendo porre in epigrafe a questo lavoro di ricerca la definizione di *sapere venatorio* data da Carlo Ginzburg (1979, pp. 66 e 67). Egli definisce la nostra attitudine a leggere e a decifrare come frutto di un sapere venatorio, cioè della

> [...] capacità di risalire da dati sperimentali apparentemente trascurabili ad una realtà complessa e non sperimentabile direttamente

per questo motivo

> [...] il cacciatore sarebbe stato il primo a «raccontare una storia» perché era il solo in grado di leggere, nelle tracce mute (se non impercettibili) lasciate dalla preda, una serie coerente di eventi.

Il cacciatore che decifra tracce di animali compie un processo del tutto simile a quello del lettore che, attraverso la materialità di simboli geometrici, ricostruisce la rete di significati espressa dal testo: il percorso va da un oggetto materiale ad un oggetto immateriale, il significato, che nulla ha a che vedere con la propria rappresentazione fisica.

Scrittura: risultato di transcodifiche

La scrittura è un atto grafico, l'atto finale di un processo che, a partire dal senso, codifica quest'ultimo attraverso la sintassi e il lessico di un linguaggio naturale e ne dà una convenzionale rappresentazione nell'ordinamento lineare di una sequenza di *glifi*.

Allo stesso modo il processo di localizzazione dell'informazione nel testo è ancora un processo di ricerca simile alla caccia: da un insieme di orme e rami spezzati all'individuazione della preda, dalle tracce del significante al rinvenimento del significato, del quale il testo codifica graficamente la realizzazione linguistica.

Testo: luogo da mappare

Mi sia permesso di introdurre un'ulteriore paragone: il testo è un luogo in cui ci si può orientare attraverso strumenti fisici e gli indici e le tabelle stanno allo spazio testuale così come le mappe e le bussole stanno all'orientamento spaziale.

L'ente unificatore che permette di stabilire l'equivalenza semantica di ciascun codice è il soggetto-lettore, interprete in grado di superare attraverso la bontà delle sue inferenze l'ambiguità delle codifiche linguistiche e grafiche.

Il testo elettronico

Venendo a discutere della versione elettronica del testo ci accorgiamo come quanto detto fino ad ora si riproponga: il testo elettronico presenta una sua materialità nella codifica binaria che permette di renderlo memorizzabile nell'elaboratore e, allo stesso tempo, viene rappresentato graficamente sullo schermo del computer perché continui ad essere leggibile al lettore umano, al quale la codifica binaria rimane nascosta. Così, il testo elettronico è *medium* tra due forme di elaborazione: come testo grafico presentato a video, si presta all'interpretazione del lettore con caratteristiche simili a quelle del testo scritto; il testo come *data type*, formato di dati in codifica binaria, è l'oggetto del trattamento automatico. Pur tuttavia questa duplice dimensione non porta ad una equivalenza delle possibilità di elaborazione: infatti, il testo rappresentato a video continua ad essere oggetto di inferenza e di interpretazione per il lettore umano, né più né meno come il testo cartaceo, ma il testo come insieme di dati è un oggetto elaborabile da parte del *computer* soltanto nella dimensione del significante. O meglio, l'elaboratore elettronico opera matematicamente (e prima ancora fisicamente) su una codifica numerica del significante. L'operazione di marcatura del testo (*markup*), di cui molto si è discusso stamani, non è altro che il tentativo di esplicitare, attraverso l'aggiunta di segni, alcune delle caratteristiche del testo che, pur essendo evidenti

Testo elettronico come medium tra sistemi di elaborazione

al lettore umano, non possono essere oggetto di elaborazione elettronica. Si tratta in altre parole di realizzare, all'interno dello spazio testuale, una sorta di segnaletica che permetta la creazione di una mappa per l'elaborazione di elementi e di dimensioni del testo che vanno oltre la reticente sequenza simbolica.

Il testo elettronico, quindi, per poter essere in qualche modo *letto* dal calcolatore deve essere mappato o come nel caso del *markup* di tipo *strongly embedded* deve contenere segni che ne conducano il tracciamento di una mappa.

Mi sia ora permessa una breve citazione letteraria, da Borges, *Storia Universale dell'infamia*:

> ... In quell'Impero, l'Arte della Cartografia giunse a una tal Perfezione che la Mappa di una sola Provincia occupava tutta una Città, e la Mappa dell'Impero tutta una Provincia. Col tempo, queste Mappe smisurate non bastarono più. I Collegi dei Cartografi fecero una Mappa dell'Impero che aveva l'Immensità dell'Impero e coincideva perfettamente con esso. Ma le Generazioni Seguenti, meno portate allo Studio della Cartografia, pensarono che questa Mappa enorme era inutile e non senza Empietà la abbandonarono alle Inclemenze del Sole e degli Inverni.

Necessità di una scelta di scala

L'opera di chi si appresta ad editare il testo per renderlo elaborabile dalla macchina è accostabile a quella paradossale dei cartografi imperiali [ma speriamo che non sia simile anche negli esiti finali]: alla base della rappresentazione digitale del testo deve esserci una scelta, per così dire, di "scala" che ne selezioni le caratteristiche che si desidera rendere elaborabili, dato che la marcatura del testo, come operazione ermeneutica di esplicitazione di strutture di significato, può facilmente produrre una mappa più grande dell'impero.

La scala scelta

La proposta di un sistema di mappatura semantica dei testi latini, che qui si formula, opera nell'ambito di una *scala* che va dalla ricognizione della dimensione lessicale del testo alla sua categorizzazione semantica, nel tentativo di fornirne una sistemazione computazionale a partire dal riconoscimen-

to dell'insufficienza degli strumenti di orientamento basati sull'ordinamento alfabetico.

La consapevolezza dell'utilità degli indici alfabetici per il reperimento delle informazioni nel testo è stata bene illustrata in Papia, che, nella prefazione del suo *Elementarium doctrinae rudimentum*, si prefiggeva di fissare *regulas certas*, perché il lettore potesse risalire velocemente ai contenuti, e sceglieva di disporre i lemmi in rigorosa successione alfabetica. Allo stesso tempo, però, Papia si rendeva conto della difficoltà che una tale disposizione comportava a causa della varietà delle grafie: egli ricorda come

> Hyaena a quibusdam per i, ab aliis per y vel per aspirationem cum diphthongo in penultima scribitur

La difformità colpiva anche la pronuncia dei vocaboli e lo stesso Papia segnalava che

> quam verbenam quidam, alii berbenam vocant herbam

Folia librorum querere

Un altro esempio di reperimento dell'informazione del testo attraverso i suoi aspetti materiali è la mnemotecnica che Ugo di San Vittore proponeva ai suoi allievi nel *De tribus maximis circumstantiis gestorum*:

> Multum ergo valet ad memoriam confirmandam ut, cum libros legimus, non solum numerum et ordinem versuum vel sententiarum, sed etiam ipsum colorem et formam simul et situm positionemque litterarum per imaginationem memoriae imprimere studeamus, ubi illud et ubi illud scriptum vidimus, qua parte, quo loco (suppremo, medio, vel imo) constitutum aspeximus, quo colore tractum litterae vel faciem membranae ornatem intuiti sumus.

Egli consigliava ai propri allievi di mandare a mente le caratteristiche grafiche della pagina, segnando nella memoria visiva la posizione delle *sententiae* nel testo. Si trattava di una forma di orientamento senza indice che si serviva della memoria posizionale e degli aspetti materiali del testo come elemento di collegamento tra il senso e la sua realizzazione in una posizione della pagina.

Precognizione dei risultati

Il reperimento dell'informazione da *corpora* testuali elettronici presenta elementi comuni al primo e al secondo metodo: è infatti quasi sempre un

orientamento attraverso un indice (poco importa se questo sia un indice statico pre-generato o oppure venga creato al volo dalla scansione lineare dei documenti) e la consultazione di una base dati testuale prevede la formulazione di ipotesi su come i concetti siano stati lessicalizzati nel linguaggio (o nei linguaggi) dei documenti archiviati e allo stesso tempo deve ricostruirne la forma di rappresentazione grafica; in altre parole l'accesso al testo ideale è possibile soltanto formulando ipotesi sul testo materiale: riprendendo la metafora venatoria potremmo affermare che è come se il cacciatore per risalire alla preda dovesse ipotizzarne la forma delle tracce [e certamente, della caccia, questo tipo di ricerca condivide tutti gli aspetti aleatori]. È chiaro il contrasto con l'esempio medievale: laddove gli allievi di Ugo di San Vittore ritrovavano la posizione dei segni nella pagina, avendone memorizzato gli aspetti materiali dopo una accurata lettura, il ricercatore, che si serve dello strumento digitale, formula una ipotesi su un aspetto materiale del testo (la sua sequenza di glifi) per ritrovare un significato che non è stato, per così dire, visitato in precedenza.

Un modello di Information retrieval per i testi latini

IR su testi latini: un modello

Parlando di testi latini, la realizzazione di un modello specifico per l'*IR* semantico si scontra con tre ordini di problemi:

- l'aspetto grafico-ortografico delle parole contenute nel testo;
- la flessione;
- l'imprevedibilità delle realizzazioni lessicali della dimensione semantica.

La proposta che qui si formula si basa sull'impiego di uno strumento di indicizzazione che permetta la compressione delle forme *allografe* o *alloglife*, la loro riconduzione ad un lemma e il collegamento dei lemmi ad una struttura, un *thesaurus semantico*, che permetta di organizzare i significati a partire dalla loro realizzazione lessicale, permettendo di simulare la competenza semantica nello strumento di ricerca.

I primi due momenti di indicizzazione operano una progressiva *reductio ad unum* degli elementi lessicali, che va dal catalogo delle forme ad un lemmario, organizzato in modo da permettere l'individuazione della posizione dei lemmi nel testo. Il terzo momento collega i lemmi individuati ad una struttura più ampia (quindi si può considerare come un processo di espansione) che costituisce ed esplicita una rete di significazione.

Ricerche lessicali

Compressione delle allografie

Fuzzy problem

Il processo di compressione delle grafie alternative si confronta con un problema di tipo *fuzzy*, dato che decidere se una stringa di un indice *è allografa* di un'altra significa sindacare sulla loro similarità, valutazione che esula da un approccio di tipo *booleano*. Non necessita di dimostrazione l'affermazione che similarità e differenza tra stringhe si snodano in un continuo, tanto che ogni stringa di un testo è simile ad un'altra, purché esse abbiano almeno un carattere in comune. Inoltre, si dovrebbe parlare non soltanto di allografia ma anche di *alloglifia*, dato che l'ambiguità tipografica permette la rappresentazione degli stessi grafemi con glifi diversi [ae ; æ]. Pertanto, nell'implementazione del sistema, si è scelto di servirsi di una tabella di accelerazione elencante le allografie più consuete e di utilizzare una combinazione algoritmica per l'individuazione degli *alloglifi* inconsueti.

Lemmatizzazione

E pluribus unum

Il secondo momento di compressione dell'indice, vede la realizzazione di un meta-indice che raggruppi le forme per lemmi apponendo ad esse un codice di descrizione. Il tipo di implementazione realizzata nella presente ricerca si serve di un approccio semi-automatico e di una base di conoscenza lessicale[1] di circa quarantamila lemmi.

La costruzione dell'indice dei lemmi avviene risolvendo le situazioni di

[1] Realizzata a partire da fonti di pubblico dominio

omografia esolemmatica (tra forme di due o più lemmi)[2] in maniera assistita attraverso un algoritmo che su base statistica individua i candidati più probabili per la disambiguazione. Questo tipo di organizzazione permette ricerche lessicali molto più efficienti rispetto a quelle possibili sulle collezioni elettroniche che operano solo sulle forme flesse.

Ricerche semantiche

Semantica e NLP

La possibilità di operare ricerche attraverso l'astrazione dei contenuti rappresenta il limite di frontiera degli attuali studi sulla elaborazione dei linguaggi naturali (NLP): di particolare interesse può risultare l'applicazione di queste tecniche alle lingue concluse, come il greco antico e il latino. Qui si propone di applicare alla struttura della frase latina una indicizzazione semantica, rifacendosi al modello di Roussey *et al.* (1999). La base per l'implementazione del modello è la realizzazione di un *thesaurus* semantico che permetta di unire più lemmi all'interno di una definizione di dominio e che permetta di chiarire le relazioni tra i lemmi, definendo due livelli di conoscenza:

1. un livello concettuale che dia un modello del campo di studio formato dai concetti e dalle relazioni che intercorrono tra di essi;

2. un campo terminologico che rappresenti l'insieme delle manifestazioni linguistiche di un concetto nel testo.

Thesaurus semantico

WordNet

Tra i tipi di modellizzazione della conoscenza lessicale è parsa particolarmente interessante la rappresentazione dei rapporti semantici nei dizionari definita dagli studi di Miller *et al.* (1990) e Fellbaum (1998): a partire dal riconoscimento della natura del tutto accidentale dello *spelling* delle parole, nel modello di *WordNet* le parole sono organizzate per blocchi di significato, denominati *synset*, che raccolgono tutti i lemmi che lessicalizzano lo stesso

[2]Per una classificazione tassonomica dell'omografia latina si rimanda a: Passarotti e Ruffolo (2004)

concetto; i *synset* sono collegati tra loro per mezzo di relazioni che includono, assieme alla sinonimia, anche l'iponimia, la meronimia e l'antinomia. Attraverso la struttura di relazione si formalizza una gerarchia tra le parole, separando il livello lessicale da quello semantico.

L'applicazione di tale modello ad un *thesaurus* semantico per la lingua latina rappresenta uno degli oggetti su cui maggiormente si è concentrata l'attività di ricerca.

Mapping

Risultando impraticabile e antieconomico per un singolo studioso costruire da zero l'insieme di relazioni di una rete di questo tipo, il punto di inizio del progetto di costruzione si è basato sull'ipotesi che la rete dei significati, definita per la versione inglese, potesse essere in gran parte portata verso altri linguaggi. Una ipotesi plausibile, se ci si limita alle principali lingue indoeuropee che presentano una vasta sovrapposizione culturale, ma che deve essere ancora verificata per una lingua conclusa come il latino.

Matrice lessicale multilingue

Multilingual lexical matrix

La matrice lessicale della *Wordnet* inglese è bidimensionale (si estende nelle due dimensioni dei lemmi e dei significati); sulla scorta di quanto effettuato in progetti analoghi[3] per le lingue moderne, aggiungendo una terza dimensione alla matrice (la dimensione delle lingue) diventa possibile considerare la lingua latina. Per realizzare la matrice multilingue, si rende necessario mappare i lemmi latini sui significati corrispondenti, andando a costruire l'insieme dei *synset* per il latino. Il risultato è stato quello di una completa ridefinizione delle relazioni lessicali, mentre per la creazione della rete di relazioni semantiche sono state impiegate, per quanto possibile, quelle già definite per l'inglese. La dimensione dei significati, pertanto, è stata considerata costante rispetto alle possibili lessicalizzazioni a livello linguistico. In un primo tempo, attraverso l'utilizzo di algoritmi di *matching* e di un dizionario bilingue, è stata verificata la corrispondenza tra i lemmi della lingua latina, che doveva essere aggiunte alla rete. Questa prima fase di automazione ha prodotto un insieme di possibili connessioni tra una parola latina e i significati nella rete

[3] Alcuni tra i più significativi: Artale *et al.* (1997); Chen *et al.* (2002); Lee *et al.* (2004)

WordNet. A questa fase è seguito l'intervento umano per validare le scelte proposte.

Non solo IR

Il campo di applicazione di una rete così costituita va dal disambiguamento dei contesti, all'IR semantico.

Il collegamento tra lemmi presenti nel testo e rete semantica, costituisce una mappa della catena di significati realizzati, che prescinde della loro lessicalizzazione: l'insieme dei riferimenti ai *synset* contenuti nella struttura di una frase diventa, così, il descrittore del contenuto di quella frase e l'elemento su cui può operare il processo di ricerca.

Indice

Capitolo 1

L'Information retrieval sui corpora testuali

Dall'inizio della civiltà umana, il sapere è stato trasmesso attraverso il linguaggio e conservato nella comunicazione scritta. Dalle incisioni rupestri ai rotoli di pergamena, dai torchi a stampa alle biblioteche elettroniche, il testo rimane la principale forma di condivisione della conoscenza. Quella del testo è una centralità culturale che solo minimamente è stata intaccata dalla recente affermazione dei sistemi multimediali, prova ne sia che Internet, il mezzo di comunicazione oggi ritenuto specchio virtuale del mondo reale, è ancora una volta un sistema di comunicazione fortemente basato su veicoli linguistici e testuali.

La progressiva digitalizzazione di collezioni di testi di interesse per gli studiosi di discipline storiche, letterarie, linguistiche e filologiche ha portato a introdurre nelle normali pratiche di ricerca scientifica l'utilizzo di banche dati testuali e motori di ricerca. Prima di intraprendere l'analisi di un sistema di reperimento dell'informazione specificamente immaginato per le collezioni di testi mediolatini è necessario porre alcune basi sostanziali per individuare i problemi di fondo che hanno stimolato la presente ricerca. In primo luogo è necessario definire gli obiettivi di ricerca a cui le collezioni di testi elettronici rispondono negli studi umanistici. Successivamente si illustrerà che cosa si intende per *Information Retrieval*[1] per poterne, poi, mostrare le specificità

[1]Da qui in avanti IR

e la singolarità nell'ambito delle collezioni di testi mediolatini: ambito nel quale, come vedremo, le pratiche di ricerca testuale, trovano differenziano per obiettivi e metodi dall'uso generale del reperimento dell'informazione.

1.1 Le collezioni di testi elettronici negli studi umanistici

Se si guarda all'utilizzo delle collezioni di testi nella pratica degli studi umanistici si può ravvisare in un percorso che conduce alla nascita di un nuovo tipo di lessicologia e, parimenti, di lessicografia.

I caratteri essenziali di questa nuova lessicografia, intesa come *lessicografia al computer*, hanno cominciato a delinearsi a partire dalle prime esperienze di automatizzazione delle ricerche sui dizionari. In questo contesto non possono essere taciute le esperienze fondanti di Padre Roberto Busa a partire dagli anni '40 del secolo scorso e del centro per l'Automazione dell'Analisi Letteraria di Gallarate e l'apporto fondamentale di Antonio Zampolli alla fondazione di una nuova metodologia di ricerca[2]. Il plauso ai portati delle nuove tecnologie verrà sancito anche da Giovanni Nencioni che in un suo intervento del 1987 descrive meglio di ogni altro le caratteristiche della nuova lessicografia.

In quel saggio si poneva in evidenza l'urgente richiesta di dinamicità, che veniva rivolta allo studio e alla classificazione operata dal lessicografo, e l'aspettativa di

> una lessicografia non solo semasiologica, ma anche onomasiologica, che cioè s'interessi, oltre che della storia semantica delle parole, del loro rapporto con le cose, cioè della denominazione degli oggetti.

Secondo Nencioni, il limite intrinseco al "dizionario alfabetico bloccato nelle sue pagine stampate e nella sua struttura" è superabile grazie a uno "strumento nuovo che ha reso possibile una nuova lessicografia",

[2]Si ricordino i fondamentali interventi di Zampolli e Duro (1968) e Zampolli (1968)

> la *banca dei dati*, cioè la costituzione di una memoria elettronica aperta ed interrogabile. Questa memoria può essere di fatto vasta o ristretta, totale o parziale, anche circoscritta a singoli generi o autori; e tuttavia non ha, di diritto, limiti quantitativi e può accrescersi e modificarsi progressivamente. Viene così eliminata la selezione imposta dalle proporzioni fisiche del dizionario tradizionale, e anche quella censoria in essa implicita; e superato è infine l'ordine alfabetico, reso inutile da un programma di reperimento e contrario alla manovrabilità e dinamicità del dizionario[3].

La prospettiva di Nencioni è quella di una rifondazione del metodo tanto che egli afferma che

> una banca di questi dati [...] modificherebbero i criteri e i limiti [...] e la stessa concezione dell'analisi lessicografica, la quale sempre più apparirebbe non una bloccata e quindi incerta registrazione e archiviazione ma il più potente strumento di conoscenza della lingua. Strumento che, come tutti gli strumenti, ovviamente è e deve essere costruito in ragione di una teoresi che sta dietro l'impugnatura o l'oculare o la tastiera, e di perseguimenti che stanno oltre la lama o l'obiettivo o l'elaboratore[4].

In questa metafora, con la quale si chiudeva l'articolo di Giovanni Nencioni, ci piace immaginare il bisturi dell'anatomopatologo, il microscopio del naturalista e il nostro elaboratore elettronico. In questa definizione emerge chiaramente come l'uso strumentale dell'elaboratore non sia disgiunto dalla teoresi che ne precede l'impiego e che di per se stessa è oggetto di ricerca, pur con un fine diverso da quello della pura scienza informatica. Una visione, quella di Nencioni, molto vicina alla definizione di una informatica umanistica come disciplina che riflette sulle forme di rappresentazione della conoscenza piu' adeguate alle singole discipline umanistiche, sui modelli di dati che se ne ricavano, e sull'implementazione dei formalismi che si possono applicare a tali modelli di dati. Da un lato la semiotica della rappresentazio-

[3] Nencioni (1987, p.149)

[4] Ibidem

ne digitale, dall'altro i procedimenti di elaborazione di tali forme specifiche di rappresentazione della conoscenza.

Chiaro è anche l'obiettivo della creazione delle banche dati testuali: la possibilità di fornire materiali che permettano lo studio del lessico e che amplino la conoscenza della lingua. Nencioni ripone nello strumento elettronico una fiduciosa prospettiva di superamento e di sostituzione del dizionario tradizionale con una banca dati di testi: entusiasmo che ci sentiamo di sottoscrivere solo in parte. L'utilizzo di banche dati, infatti, ha senz'altro rivoluzionato il metodo lessicografico in quelle operazioni concernenti le fasi preparatorie della realizzazione di dizionari, ma non può sostituire il dizionario come opera dove la conoscenza lessicale è organizzata e strutturata in modo coerente. A tal proposito occorre citare un recente intervento di Busa (2004):

> lexica historica unius linguae, etiam non mortuae, utpote documenta quae vere tale nomen merentur, numquam totaliter atque integre obsolescunt

L'operazione di riflessione metalinguistica, di normalizzazione e di organizzazione strutturale che compone la pratica della costruzione di un dizionario, indubbiamente può essere migliorata attraverso l'utilizzazione di spogli elettronici e certamente l'uso di collezioni di testi gestibili elettronicamente ha portato a modificare le pratiche di quella che da Quemada (1990) viene definita "*lexicographie prédictionairique*". La versatilità degli strumenti elettronici costituiti da una base dati testuale associata ad un sistema di ricerca ha certamente contribuito in un primo tempo alla semplificazione del lavoro di costruzione degli indici di forme e delle concordanze a stampa, così come la diffusione del *personal computer* ha successivamente reso inutile il passaggio al cartaceo consentendo direttamente lo sfruttamento delle banche dati digitali da parte degli studiosi.

Nuovi problemi, però, si affacciano nelle pratiche della lessicografia al *computer* e nella costruzione delle raccolte digitali. Prima di tutto, diventa fondamentale la fase di trascrizione dei dati dei *corpora*, che già era stata sottolineata a più riprese nel quadro dell'analisi conversazionale[5]; Elinor Ochs

[5]cfr. Sacks *et al.* (1974)

(1979) parla giustamente di «transcription as a theory» in quanto il fatto di trascrivere comporta scelte, priorità, obiettivi. Si tratta della prima fase dell'analisi di un testo, in cui si prende contatto con i dati e si cominciano a formulare ipotesi. Poiché l'adeguatezza di una trascrizione non è mai assoluta, ma dipende dalla natura dei dati e dall'indagine cui si intende sottoporli, occorre prima di tutto valutare quali criteri adottare. Secondo la proposta avanzata da Orletti e Testa (1991), ci si deve innanzitutto chiedere quale grado di specializzazione dare alla trascrizione, nella consapevolezza che potranno essere sottoposti ad elaborazione elettronica solo i fenomeni che in qualche modo vengano marcati e segnalati per renderne possibile l'identificazione da parte del calcolatore; in tale processo occorre procedere con estrema coerenza interna: i simboli devono essere usati con il medesimo valore in quanto il rapporto tra segno e fenomeno che esso rappresenta risulti biunivoco. Si tratta certamente di una vera fase di preedizione dei testi, se si guarda al risultato finale della loro edizione digitale, e di riedizione rispetto al materiale di partenza[6].

La costituzione di un *corpus* digitale pone sempre il problema di soddisfare le esigenze e gli interrogativi degli studiosi che ne saranno gli utenti privilegiati: decisioni che si collocano a monte dell'organizzazione dei dati, connessa con la tipologia del materiale raccolto e con le finalità dell'analisi linguistica. Il primo scopo di ogni strumento lessicografico si indentifica con l'indagine dei fenomeni lessicali ed entro ques'orizzonte appaiono fondamentali almeno quattro problemi:

- la categorizzazione dei materiali
- l'individuazione delle frequenze
- l'esame del rapporto con i contesti
- l'individuazione dei significati

[6]La preparazione dei testi in questo senso rappresenta una delle fasi cardine su cui poggia l'idea di una filologia digitale, i metodi della quale meriterebbero un discorso a parte che farebbe deviare troppo dall'argomento della presente ricerca

1.1.1 La categorizzazione dei materiali

Poiché lo studio delle associazioni tra una parola specifica e fattori non linguistici appare come un aspetto non trascurabile nella ricerca e la richiesta da parte degli studiosi di lessicografia appare sempre più legata alla necessità di una consultazione delle collezioni di testi orientata alla granularità dei dati[7], risulta chiaro come sia necessario nella costruzione di *corpora* testuali destinati alla ricerca scientifica, una fase di vera e propria suddivisione tassonomica dei materiali testuali, che può essere compiuta solo attraverso uno studio accurato delle collezioni, per evidenziarne tutte le caratteristiche necessarie all'elaborazione di successive richieste. È ormai impensabile che lo strumento di ricerca operi soltanto sul testo e l'arricchimento dei *testi puri* attraverso l'aggiunta di un apparato metatestuale in grado di evidenziare le caratteristiche sincroniche e diacroniche dei materiali contenuti in un *corpus* non solo rende possibile un migliore accesso da parte dello studioso ai testi contenuti nelle raccolte elettroniche, ma anche amplia le possibilità elaborative a cui possono essere sottoposti i testi per l'estrazione automatizzata di dati e l'applicazione di metodi di analisi qualitativa e quantitativa.

Se tradizionalmente nella letteratura riguardante il reperimento dei dati e il reperimento dell'informazione si separano i dati strutturati, di solito conservati in *database relazionali*, e dati semi-strutturati come il testo, che normalmente è l'oggetto su cui operano i sistemi di *Information Retrieval*. Un tempo le due dimensioni erano fortemente separate, in quanto ciascun tipo di strumento supportava metodi di accesso ai dati e strutture peculiari. Oggi la distinzione tra dati strutturati e semistrutturati sta rapidamente svanendo[8] con l'emergere di sistemi ibridi che sfruttano l'organizzazione dei dati tipica dei database e la possibilità di accesso diretto al testo, più legata agli strumenti di *text retrieval*[9]. A testimonianza di questa integrazione

[7] Intendendo con essa la possibilità di differenziare quanto possibile i risultati dell'indagine sui corpora, aggregando i dati restituiti secondo criteri associativi e disgiuntivi

[8] Una accurata disamina dei metodi di integrazione può essere trovata in Grossman e Frieder (2004, pp. 211-255)

[9] In particolare l'avvento e la diffusione dei linguaggi di marcatura ha reso possibile l'annotazione diretta e non ambigua dei fenomeni testuali e la creazione di sistemi semplici per l'integrazione dei dati metatestuali più vari. Un esempio classico di questi sistemi

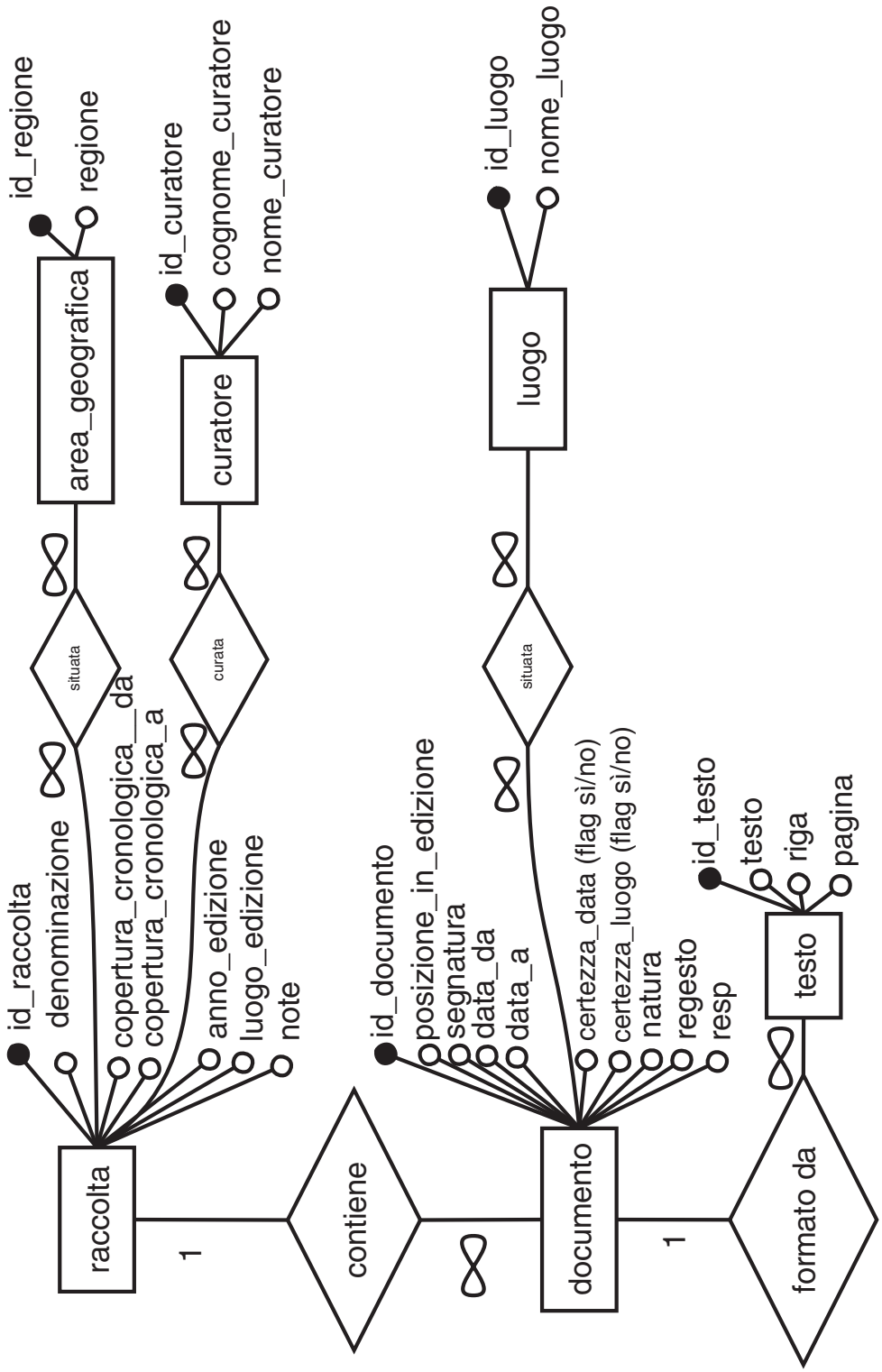

Figura 1.1: Lo schema ER della parte documentale del progetto ALIM

può essere portato l'esempio dell'integrazione tra dati testuali e metatestuali presente nel progetto ALIM che ha come obiettivo la raccolta dei testi latini di area italiana prodotti durante il medioevo. Il sistema di gestione delle ricerche scelto si avvale di un programma di *text-retrieval* collegato ad una base di dati relazionale: le ricerche possono quindi sfruttare le possibilità di aggregazione dei risultati offerti dalla struttura relazionale assieme alla versatilità delle ricerche *full-text*. Nella figura 1.1 è rappresentato lo schema ER della base dati relativa alla gestione dei testi in latino documentario: gli attributi di ciascuna relazione rappresentano i punti di aggregazione sulla base delle quali è possibile organizzare i dati testuali.

1.1.2 L'individuazione delle frequenze

Attraverso la raccolta elettronica è possibile adottare tecniche di analisi computazionale che permettono di ricostruire la frequenza d'uso delle parole. Gli studi lessicografici in questo caso si incontrano con quelli *lessicometrici* che hanno come obiettivo la descrizione della lingua attraverso gli strumenti delle scienze statistiche. L'apporto delle tecniche lessicometriche presuppone un ulteriore grado di formalizzazione e razionalizzazione del metodo di ricerca, un ulteriore passo verso l'uso di strumenti d'analisi che permettono di:

- dare fondamento quantitativo ad elementi che nell'analisi tradizionale, attraverso la lettura sequenziale del testo, rimangono relegati all'ambito dell'ipotesi e dell'intuizione
- porre in evidenza fenomeni linguistici e di contenuto nei quali il ricercatore può ravvisare fatti rilevanti.

Risulta chiaro come, nell'utilizzo di tali metodi, siano presenti, relativamente alla verifica delle ipotesi di ricerca, sia una funzione diagnostica sia una funzione prognostica.

è costituito dal progetto TEI (`http://www.tei-c.org`), che, come si legge nella pagina declaratoria dell'iniziativa, «*enables libraries, museums, publishers, and individual scholars to represent a variety of literary and linguistic texts for online research, teaching, and preservation*»

In ambito lessicografico le tecniche di *text retrieval* assieme a quelle di *text mining* permettono di migliorare l'estrazione e la comprensione dei materiali di spoglio utilizzati per la costituzione di dizionari.

Da un punto di vista quantitativo è possibile individuare nell'analisi frequenziale uno dei metodi per la formalizzazione della competenza linguistica degli autori di testi, fornendone un quadro, per così dire, fotografico. Nell'analisi di *corpora* testuali ampi ed eterogenei per provenienza, paternità e datazione è possibile individuare le permanenze degli usi linguistici di più lunga durata situandole da un punto di vista geografico e cronologico.

Nel caso dei testi in latino e delle lingue flessive uno dei problemi principali nell'allestimento di strumenti di analisi lessicometrica è costituito proprio dalla presenza di *allografi morfologici* che devono essere trattati attraverso opportuni strumenti di lemmatizzazione per non invalidare i risultati statistici[10].

1.1.3 L'esame del rapporto con i contesti

Esaminare il modo in cui le parole si raggruppano, per lo studioso, è preliminare all'indagine sulla distribuzione dei sensi e degli usi e al confronto tra formule. I sistemi di ricerca che vengono associati alle collezioni di testi destinati all'uso lessicografico dovrebbero permettere la possibilità di una rapida visualizzazione del contenuto dei contesti per renderne agevole l'analisi morfosintattica e semantica. Tali procedure si fondano sulle potenzialità del formato digitale nell'aspetto relativo alla presentazione dei dati (*display*) e dal punto di vista informatico in genere non presentano problemi teorici di particolare complessità: esso in genere è comunque uno degli elementi più apprezzati della rivoluzione digitale, pur non costituendo un'innovazione del metodo lessicografico, migliora l'ergonomia del lavoro tradizionale di spoglio.

1.1.4 L'individuazione dei significati

L'individuazione dei significati di una parola, finalità peculiare della lessicografia, può essere perseguita sulla base del repertorio completo delle occor-

[10]Si rimanda al capitolo specifico per una trattazione estesa di questi problemi

renze inserite in contesto accessibili attraverso gli strumenti di ricerca, eliminando l'aleatorietà di procedimenti intuitivi condotti su un numero parziale di occorrenze. La disponibilità e la facilità di accesso a tutto il materiale dovrebbe facilitare l'allestimento del lessico, definendo i significati dei termini, mettendoli in rapporto reciproco, distinguendo tra polisemia e omonimia.

Accanto a questo processo che va in senso tradizionale dalla collezione di testi alla costituzione del dizionario esistono anche altre possibilità di utilizzo delle raccolte testuali sotto il profilo degli studi semantici.

In primo luogo l'evoluzione ricerche sull'Elaborazione del linguaggio naturale (*Natural Language Processing*) ci porta a considerare l'opportunità di migliorare la prospettiva di impiego delle collezioni di testi, sfruttando la possibilità di integrazione di metodi provenienti da questi ambiti per poter ottimizzare la gestione dell'informazione nelle raccolte.

In particolare una suggestione nata da un intervento di Antonio Lamarra (2004) ci ha portato a prendere in considerazione la possibilità di un processo inverso che vada dal dizionario alla collezione di testi per migliorare l'organizzazione delle ricerche. Lamarra affermava che:

> Until now, automatic dictionaries have mostly been considered as analyzers of word-forms [...]. However, it would be extremely useful to make dictionaries work also in the opposite way, especially in connection to retrieval functions

Questa suggestione è stata sviluppata nel presente studio con l'obiettivo di migliorare la ricerca dell'informazione nell'ambito dei *corpora testuali* mediolatini, nella prospettiva di migliorare il reperimento dei dati lessicali attraverso l'introduzione di opportuni trattamenti automatici e di ipotizzare un sistema di reperimento del contenuto semantico.

Nei prossimi paragrafi si cominceranno a delineare nel senso generale dell'Information Retrieval i principali problemi con i quali questa prospettiva di ricerca ha dovuto dialogare.

Dizionari scientifici automatizzati per il latino classico sono disponibili già da lungo tempo, ma essi producono risultati molto modesti se applicati a testi medievali. Ne consegue che i dizionari automatici come quelli sviluppati dal LASLA o presso l'ILC, non sono adeguati per l'utilizzo su testi che

appartengono alla tarda latinità e non classica dove il contenuto dei testi è prevalentemente teologico, filosofico, scientifico e tecnico. La lingua latina medievale come è noto si amplia notevolmente attraverso l'apporto di neologismi semantici e semasiologici: una situazione che durerà almeno fino al diciottesimo secolo, continuando ad essere una lingua vitale produttiva.

1.2 Ritrovamento dei dati e ritrovamento dell'informazione

In primo luogo bisogna chiarire che l'IR presenta due principali aspetti: il *ritrovamento dei dati* e il *ritrovamento dell'informazione*.

Entrambi gli aspetti presuppongono l'esistenza di tre principali attori:

- un utente che formula una richiesta (*query*)
- un sistema automatizzato di gestione dei documenti
- una collezione di *documenti*

In entrambi i casi, alla *query* formulata dall'utente, il sistema di gestione risponde restituendo una lista di risultati ordinati. Il termine *documento* nell'IR è utilizzato per denotare una singola unità informativa di testo in formato digitale. Documento nel senso dell'IR può essere una completa unità logica, come un articolo di ricerca, un libro o un manuale, ma anche essere una parte di un testo più esteso, come un periodo, un paragrafo o una sequenza di paragrafi. Anche la glossa a un lemma di un dizionario è chiamata *documento* nell'ambito dell'IR. La collezione di documenti costituisce un insieme statico o relativamente statico che subisce una procedura di indicizzazione prima di poter essere gestito dal sistema di ricerca. Sul processo di indicizzazione torneremo più avanti, essendo esso il vero cuore dei sistemi per il reperimento dei dati e delle informazioni.

I sistemi orientati al ritrovamento dei dati e quelli orientati al ritrovamento dell'informazione sono fortemente separati dal punto di vista teorico e metodologico e proprio sulla base di questa distinzione, come vedremo, deve essere sviluppata la peculiarità dell'IR su testi latini e mediolatini.

Il *ritrovamento dei dati*, nel contesto dell'IR, si pone come obiettivo quello di determinare quali documenti contengono le parole utilizzate nella *query*, e ciò, nella maggior parte dei casi. Un linguaggio di *data retrieval* permette di recuperare tutti gli oggetti che soddisfano condizioni ben definite, come quelle date da espressioni regolari[11] o da espressioni in algebra relazionale. Dunque, in un processo di *data retrieval*, un singolo oggetto erroneamente recuperato (o non recuperato) su migliaia di oggetti, costituisce la prova di un algoritmo di ricerca difettoso per la tipologia di oggetti che si intende recuperare.

L'utilizzo delle collezioni in ambito umanistico privilegia prevalentemente gli aspetti legati al *data retrieval*, proprio per la natura lessicologica del loro impiego. Esaminiamo da questo punto di vista i primi tre dei punti precedentemente evidenziati che abbiamo attribuito a quel contesto d'uso:

- la categorizzazione dei materiali: è legata al *data retrieval* in quanto si aggiungono al testo elementi di carattere metatestuale (*metadati*) che devono essere recuperati in quanto tali
- l'individuazione delle frequenze: è un recupero di dati quantitativi rispetto al livello lessicale ed eventualmente al livello dei metadati (*quante occorenze di una parola nei testi di un autore, di un periodo, di un genere ecc.*)
- l'esame del rapporto con i contesti: è il recupero di dati testuali o *stringhe*, rispondenti ad un determinato criterio di ricerca

Il *data retrieval* può quindi essere considerato il principale scopo della costruzione e dell'utilizzo di *corpora* digitali nell'ambito umanistico e, più specificamente, lessicografico. È vero che la digitalizzazione e la messa in rete costituisce un efficace mezzo di diffusione "senza carta" di testi altrimenti difficilmente reperibili, ma senz'altro la fortuna delle collezioni elettroniche non è dovuta ad un'improbabile fruizione a video[12], quanto piuttosto alla

[11] Per una trattazione estesa delle proprietà delle espressioni regolari cfr. Baeza-Yates e Ribeiro-Neto (1999, pp.72-73)

[12] Solo per amore di obiezione si potrebbe negare che chiunque di noi, volendo studiare integralmente un'opera disponibile solo in formato digitale, non tenterebbe di stamparla, prima di intraprendere una scomoda e stancante lettura davanti a un monitor

possibilità di operare sul testo ricerche con una facilità e una rapidità altrimenti impensabili. La ricerca lessicale in ogni caso, sia essa operata con finalità di studio lessicografico, o per l'individuazione di contesti d'uso con scopo ermeneutico, si colloca in pieno nell'ambito del *data retrieval*.

Il ritrovamento dell'informazione, sposta invece il punto di vista sul *concetto* che la *query* vuole descrivere, cercando di "interpretarne" il contenuto semantico, per poter restituire i documenti più attinenti a tale argomento. La principale differenza tra questi due approcci è nella modalità di intendere la richiesta: un processo di *ritrovamento dei dati* vede la richiesta come una semplice ricerca di una o più parole all'interno dei documenti, mentre un sistema orientato al *ritrovamento dell'informazione* ha il compito di individuare quale informazione semantica si vuole accedere; mentre la teoria delle basi di dati ha a che fare con richieste sotto forma di precisi predicati, nell'IR si ha a che fare con il nebuloso e mal definito concetto di rilevanza, che dipende in modo intricato dall'intento dell'utente e dalla natura del *corpus*: come evidenziato nell'introduzione, il problema si sposta sul piano semiotico, con tutti i problemi legati all'ambiguità dell'oggetto testo come macchina significante. Per questo, nel momento in cui i sistemi di ritrovamento dell'informazione vengono valutati, spesso si evidenzia l'assenza dai risultati restituiti di numerosi documenti che potrebbero essere ritenuti rilevanti da parte dell'utente, come hanno osservato Blair e Maron (1985) nella loro ricerca sulla valutazione dei sistemi di IR.

Un sistema di IR in senso proprio, dunque, ha a che fare con quella che già da Salton (1969, p.70) veniva indicata come *fuzzyness of natural language*. A tale proposito, Gordon (1997) sottolinea come gli utenti siano piuttosto accondiscendenti nelle loro aspettative nei confronti dei sistemi di reperimento dell'informazione. Un aspetto importante da evidenziare, che ha fortemente influenzato i metodi per la ricerca dei documenti, è il considerevole incremento dei dati da memorizzare e da gestire che si è registrato con la formazione di corpora testuali digitali sempre più vasti. Questo ha comportato lo sviluppo di particolari sistemi software, ovvero i database, in grado di migliorare la gestione dei dati, archiviandoli e strutturandoli in maniera omogenea. Un buona organizzazione dei documenti migliora il sistema di ritrovamento, tuttavia, come è stato precedentemente accennato laddove

si parlava della categorizzazione dei materiali, non sempre è sufficiente a soddisfare le richieste di ricerca inerenti al contenuto di un documento. Questi sistemi infatti, non sono orientati a discriminare i documenti dal punto di vista semantico, ma solo attraverso i metadati associati al testo [13].

Il compito di ricercare un documento che contenga una specifica informazione al suo interno è lasciato all'utente, ma facilmente la mole dei dati può rendere improponibile una ricerca manuale. In questo scenario l'IR è di grande ausilio, perché attraverso il testo dei documenti dovrebbe risalire al livello contenutistico. Tuttavia, con interrogazioni poco selettive, il sistema IR potrebbe dare in risposta numerosi documenti, di cui non si conosce a priori il grado di rilevanza. Anche in questo caso, è l'utente a dover cercare, tra tutti i documenti restituiti, quello più attinente alla sua richiesta. Le difficoltà nel raggiungere questo obiettivo, è nel predisporre un insieme di algoritmi che siano in grado sia di discriminare quale informazione l'utente vuole ritrovare, sia di stabilire, per ogni documento ritrovato, quale possa essere il grado di rilevanza rispetto a tale informazione: per assurdo se questa pagina fosse inclusa in un sistema di IR e si formulasse una *query* per ottenere informazioni su Gregorio Magno, la semplice discriminante testuale porterebbe ad includere questa pagina nei risultati, anche se essa non contiene dati rilevanti sul personaggio. Un successo, quindi, solo sul versante del reperimento dei dati, ma non su quello del reperimento dell'informazione.

Si deve osservare, tornando all'utilizzo delle collezioni elettroniche di testi nell'ambito degli studi umanistici, che gli strumenti di ricerca collegati alle collezioni si sono sviluppati solo nell'ambito del *data retrieval*lessicale, mentre sono del tutto assenti strumenti di interrogazione dell'informazione semantica: qualora lo studioso desideri riscontrare la presenza di determinati argomenti all'interno di un'opera non può fare altro che ricorrere alla lettura integrale, eventualmente ad epitomi se disponibili, o alla ricerca per tentativi in base a parole chiave che come nell'esempio precedente, possono condurre a risultati inaspettati e non voluti.

Vale la pena di ricordare qui che inferenza e significazione sono processi umani che nella *semiotica cognitiva* di Peirce, come essa viene definita dal curatore dell'edizione italiana, Massimo Bonfantini, vengono accomunati:

[13] A tal proposito forniscono chiari criteri discriminanti Grossman e Frieder (2004, p.212)

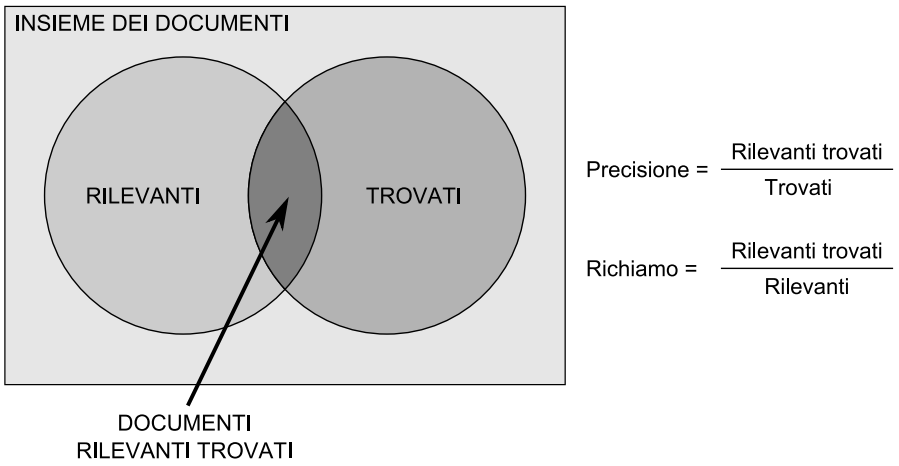

Figura 1.2: Il gruppo di risultati di una query e i parametri di valutazione

> Un segno, o *representamen*, è qualcosa che sta a qualcuno per qualcosa sotto qualche rispetto o capacità [...] Definisco un *Segno* come qualcosa che da un lato è determinato da un *oggetto* e dall'altro determina un'idea nella mente di una persona, in modo tale che quest'ultima determinazione, che io chiamo *interpretante* del segno, è con ciò stesso mediatamente determinata da quell'*oggetto* (Peirce, 1980, p.132 e 194)

Nell'impossibilità di ricostruire il processo di semiosi attributiva presente nella *query* e in generale in tutti gli atti di significazione sta il maggiore ostacolo alla realizzazione di sistemi esperti sia nell'ambito dell'IR sia in quello dell'intelligenza artificiale. È comunque possibile cercare di costruire strutture di dati in grado di favorire dei processi di annotazione automatica o semi-automatica dei testi che possano permettere un più agevole accesso al contenuto, permettendo l'integrazione di metadati che possano in qualche modo descrivere anche gli aspetti semantici dei documenti.

1.3 Valutazione dei sistemi di IR

È proprio nell'ambito dei sistemi di IR in senso proprio che si sviluppa quindi un complesso problema di valutazione dell'efficacia: prima di passare all'analisi dell'efficacia del motore di ricerca associato ad alla *Patrologia La-*

tina[14], è necessario che vengano introdotti alcuni concetti chiave relativi alle modalità di valutazione dei sistemi dal punto di vista dell'IR.

1.3.1 Parametri di valutazione

Nell'immagine 1.2 si illustrano le categorie critiche fondamentali che possono essere applicate ad una *query* eseguita. Il risultato che una *query* produce su una collezione è la restituzione di un insieme di documenti che vengono recuperati, in parte quei documenti risultano *rilevanti* rispetto alla *query*. Nel sistema perfetto questi due gruppi dovrebbero essere equivalenti: in tal caso si recupererebbero solo documenti rilevanti. In una situazione reale, invece, il sistema può recuperare documenti rilevanti e non rilevanti.

Per misurare l'*efficacia* del sistema di IR vengono usati due rapporti: *precisione* (*precision*) e *richiamo* (*recall*).

La *precisione* è data dal rapporto tra il numero di documenti rilevanti richiamati e il numero totale dei documenti richiamati: questo valore fornisce una indicazione quantitativa della qualità del gruppo di documenti di risposta alla *query*, tuttavia questa misurazione non prende in considerazione il numero totale dei documenti rilevanti presenti nella base dati. Ad un sistema viene attribuito un valore di precisione buono se richiamati dieci documenti se ne trovassero nove rilevanti (con una precisione di 0,9). Se i documenti rilevanti della base dati fossero solo nove si potrebbe affermare che si è ottenuto un enorme successo, ma se i documenti fossero milioni , si sarebbe ottenuto un risultato insoddisfacente. Per questo motivo è necessario utilizzare anche un altro parametro nella valutazione del sistema.

Il *richiamo* considera anche il numero totale di documenti rilevanti. Calcolare il numero totale di documenti rilevanti per una *query*, come osserva Kantor (1994) nella sua disamina delle misure di efficacia, non è una operazione banale: il calcolo esatto può essere effettuato solo leggendo l'intera collezione, quindi solo nella fase di messa a punto del sistema di IR, quando in un "ambiente controllato" si predispone una raccolta di prova, con un determinato *set* di *query*. Il rapporto tra *query* e rilevanza di un documento è in gran parte soggettivamente stabilito dall'utente e dipende dall'attribuzione

[14] `http://pld.chadwyck.co.uk/`

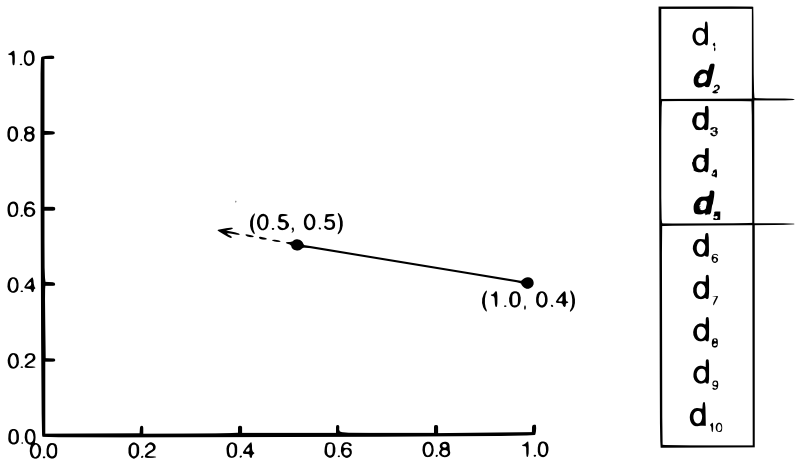

Figura 1.3: Precisione e due punti di richiamo

di significato che l'utente dà alle parole con cui la richiesta viene formulata. All'aumentare dell'ampiezza e della eterogeneità della collezione aumenta la probabilità del recupero di documenti non rilevanti a causa della polisemia. A ciascun livello di richiamo è possibile calcolare la precisione. Prendiamo ad esempio una *query* q per la quale siano stati stimati come rilevanti due documenti della collezione; ipotizzando che l'utente invii al sistema la *query* q e che vengano trovati dieci documenti, inclusi i due rilevanti (indicati come d_2 e d_5) la linea inclinata della figura 1.3 mostra che dopo aver ritrovato due documenti, un solo documento è rilevante, pertanto il valore di richiamo è del 50%, la precisione è anch'essa del 50% in quanto su due documenti trovati uno solo è rilevante. Per ottenere il 100% di richiamo si devono continuare a recuperare documenti fino a quando entrambi i documenti rilevanti non siano stati individuati: nell'esempio è necessario trovare almeno cinque documenti. A questo punto di richiamo la precisione è del 40% perché, sui cinque trovati, sono rilevanti due documenti. Nella figura 1.4 viene mostrata una tipica curva di precisione-richiamo: più alto è il livello di richiamo che si vuole ottenere maggiore è il numero dei documenti che devono essere recuperati dalla collezione. Nel sistema perfetto, vengono recuperati soltanto i documenti rilevanti: ciò significa che a ciascun livello di richiamo la precisione sarà 1,0.

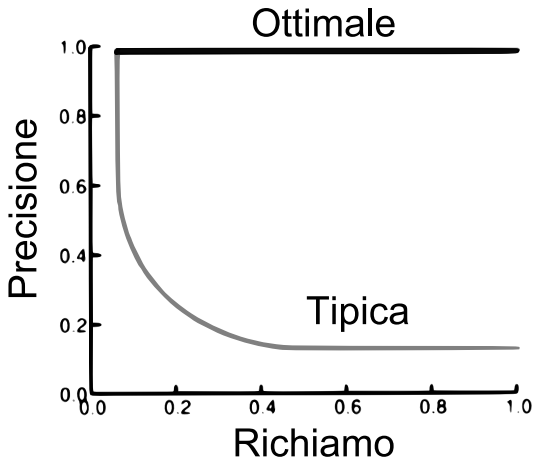

Figura 1.4: Rappresentazione di precisione-richiamo ottimale e tipica

1.3.2 Il caso del *Patrologia Latina Database*

Patrologia Latina Database è l'edizione elettronica a testo completo della prima edizione della Patrologia Latina di Jacques-Paul Migne pubblicata tra il 1844 e il 1855, e dei 4 volumi di indici pubblicati tra il 1862 e il 1865. La Patrologia Latina comprende le opere dei Padri della Chiesa, da Tertulliano (200 d.C.) alla morte del papa Innocenzo III (1216). Il database contiene il testo integrale della Patrologia Latina con l'apparato critico e gli indici, con la possibilità di operare ricerchi testuali attraverso stringhe di ricerca che possono avvalersi di operatori booleani e di caratteri sostitutivi. Il modello del Patrologia Latina Database può essere considerato standard per i sistemi di ricerca orientati al *text retrieval* nell'ambito delle collezioni di testi in latino tardo e medievale.

Prendiamo ora in esame una ricerca lessicale sul verbo *duco*, condotta digitando una *query* nella forma:

```
duc*
```

L'intento è quello di ottenere in risposta tutte le forme derivate dal tema del presente *duc-*. Tale query produce un totale di 404 risultati. Soltanto 118 dei quali sono rilevanti per la *query* eseguita: un livello di precisione molto basso (circa il 29%).

Un altro esperimento può essere fatto con le forme del verbo *fero*, attraverso la query:

```
fer* or tul* or lat*
```

Su 2104 risultati solo il 7% risulta essere forma del verbo *fero*. Questi risultati mostrano come la precisione del sistema di ricerca sia molto bassa, soprattutto nel caso di temi brevi, mentre migliora se si cercano parole con più lunghe[15].

Questi risultati sono del tutto omogenei con quelli ottenuti attraverso altri sistemi di ricerca su collezioni di testi mediolatini, in quanto si tratta di adattamenti di procedure di *text matching* che non tengono conto delle esigenze degli utilizzatori di queste collezioni, ma adattano i principi più generici nel trattamento dei dati.

In particolare si deve evidenziare come gli strumenti di *data retrieval* lessicale che accompagnano collezioni di testi mediolatini, si dimostrino insufficienti almeno sotto due punti di vista:

- non tengono conto dei problemi legati agli aspetti grafico-ortografici dei testi presenti nelle collezioni, fallendo frequentemente davanti alla variabilità grafica di tali testi

- non permettono un accesso alle forme lemmatizzate, cioè raggruppate tenendo conto degli aspetti flessivi della lingua

Operare ricerche su stringhe di caratteri in questo modo, aumenta anche il rischio di perdita dell'informazione. Busa (2000, p.166), riferendosi a ricerche compiute su testi nella propria lingua madre, che l'operazione di ricerca testuale con troncamento:

> incrementa l'ingenua ma errata supponenza di conoscere sufficientemente le possibilità lessicali della propria lingua, pur nativa, tanto da potersene fidare anche alla *consolle* di un computer

[15] Per esempio la ricerca del verbo *ambulare*, attraverso il tema del presente produce 129 risultati, dei quali 94 pertinenti, con una precisione del 73%

La cautela, quindi, con una lingua straniera dovrà essere ancora maggiore. Per esempio, nel caso della ricerca di tutte le forme del verbo abscondo attraverso la stringa abscond*, i risultati sarebbero parziali, in quanto escluderebbero forme (possibili) come apscondit, acondit, abcondit, absconsus, absconsurus,... non considerate dall'utente nel momento in cui scrive la stringa di ricerca abscond*, supponendo di recuperare, in questo modo, tutte le forme desiderate.

1.4 Da un generico sistema di IR a una proposta per i testi mediolatini

Il miglioramento delle funzioni di ricerca deve essere illuminato tenendo conto del modello linguistico peculiare alla lingua latina medievale: è pertanto necessario partire dall'analisi del funzionamento di un generico sistema di IR, per poterlo successivamente adeguare con opportuni adattamenti. Nella figura 1.5 è riportata una rappresentazione funzionale di un sistema di IR paradigmatico[16]. Il sistema funziona sulla base di due processi distinti ma complementari: una prima fase è relativa all'*indicizzazione* dei documenti, cioè l'insieme dei processi che vedono la strutturazione dei dati da parte del sistema per poterne operare la gestione e il recupero(parte alta dello schema), la seconda fase descrive il processo di interrogazione del sistema, dove avviene l'interazione tra l'utente e i dati immagazzinati (in basso nello schema).

Nella fase di indicizzazione, a partire da un documento immesso nel sistema, vengono estratte le parole componenti il testo (con la possibilità di escluderne alcune ritenute irrilevanti), ne viene segnata la posizione, le parole sono eventualmente sottoposte a troncamento (*stemming*) per motivi di compressione e aumento della restituzione, infine vengono inserite in un indice posizionale.

La fase di *query* è simmetrica a quella di indicizzazione a loro volta, le richieste dell'utente vengono processate in modo analogo e comparate a quanto presente nell'indice della base dati e i documenti compatibili con la richiesta (*query*) vengono restituiti secondo un ordine basato su un sistema

[16]Da Baeza-Yates e Ribeiro-Neto (1999, p.7), traduzione nostra

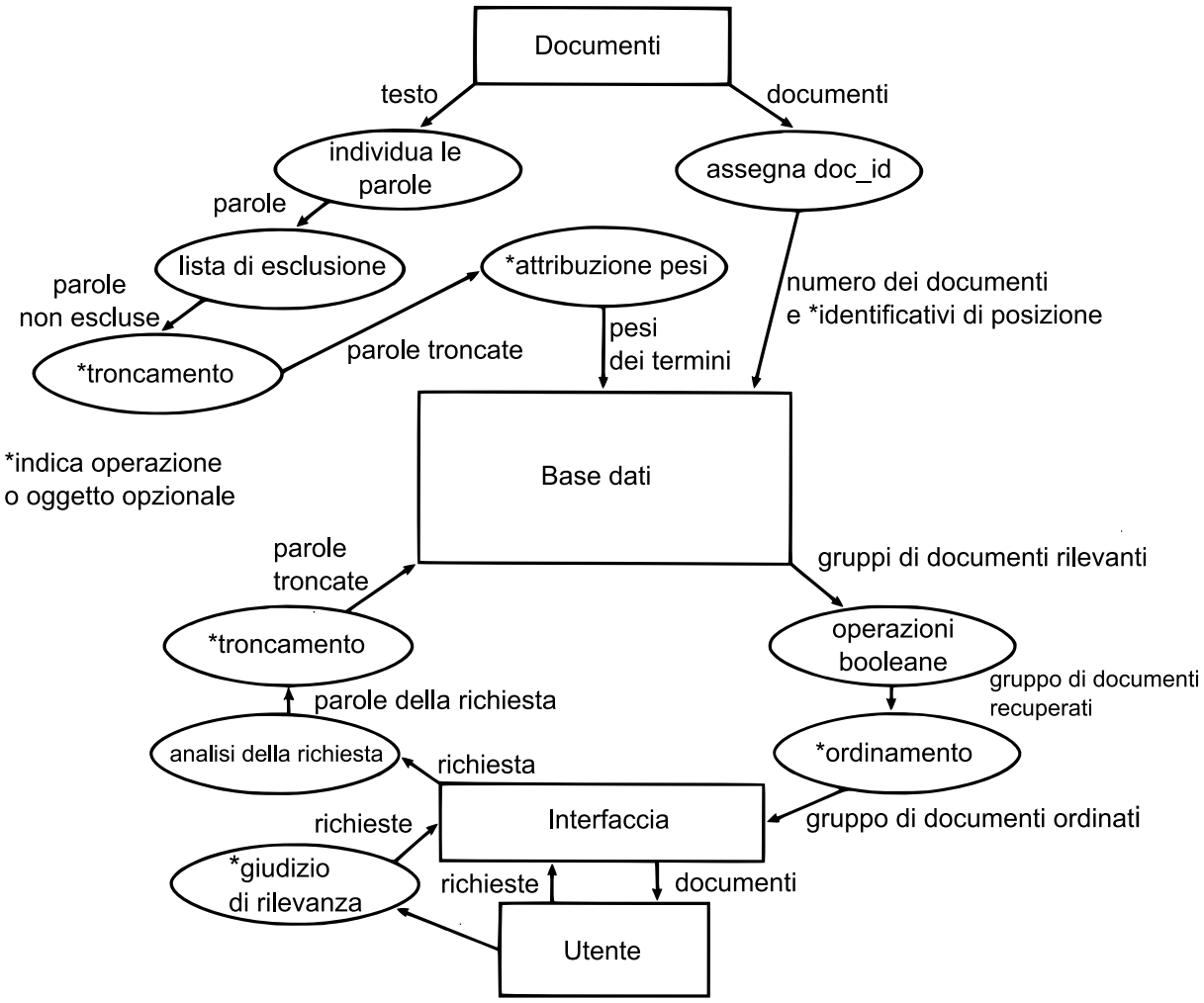

Figura 1.5: Analisi del dominio di un sistema IR generico

di ordinamento (*ranking*) che attribuisce maggiore o minore rilevanza ai dati attraverso l'analisi dei pesi (*weights*) attribuiti alle parole del documento.

Questo modello canonico funziona ottimamente per il reperimento dei dati lessicali nel caso dell'inglese moderno, per il quale è stato da tempo creato un algoritmo di compressione attraverso *stemming*[17]. Data la marginalità dei fenomeni flessivi in questa lingua, riconducibili a poche regole facilmente sintetizzabili con algoritmi non viene mai affrontato il problema della *lemmatizzazione*.

Lemmatizzazione non stemming

Per i testi in latino il sistema di *stemming* deve essere sostituito da un opportuno processo di compressione delle forme allografe e di lemmatizzazione, dato che gli esperimenti di compressione degli indici attraverso *stemming*, fino ad ora tentati[18], risultano non del tutto soddisfacenti sul piano della precisione dei risultati restituiti.

Gli stessi lemmi in latino frequentemente presentano due o più parti invarianti[19]: pertanto il semplice troncamento a destra non produce risultati adeguati; inoltre la le radici latine per la maggior parte tendono ad essere piuttosto brevi (come mostrato nell'esempio relativo alla *query*) e molte radici sono condivise da parole di significato differente[20]. Questi fattori fanno sì che difficilmente la ricerca su parole troncate porti ad individuare solo parole semanticamente collegate alle parole della query.

È pertanto necessario riprogettare un sistema che tenga conto degli aspetti linguistici ed eventualmente migliorare il sistema di Information Retrieval attraverso metodi di compressione specifici per la lingua latina. La figura 1.6 descrive il processo relativo al trattamento delle parole in ingresso nel sistema di IR riprogettato per adattarsi alla lingua latina. Al posto del processo di troncamento è stato inserita una procedura di lemmatizzazione morfologica che permette di migliorare la ricerca lessicale[21]. Un modulo di marcatura semantica, interviene inoltre sulle parole lemmatizzate, legando ciascuna for-

[17] Si tratta dell'algoritmo di Porter (1997) che ha trovato vastissima applicazione nell'ambito dei sistemi di IR inglesi e che ha dato l'avvio a studi analoghi per quasi tutte le lingue moderne

[18] In particolare il sistema presentato in Schinke *et al.* (1997)

[19] Si pensi per esempio alla ai verbi o dei nomi della terza declinazione

[20] Per esempio *port-* è in *portus*, in *porta*, e in *portare*

[21] Sarà descritto estesamente al capitolo 3

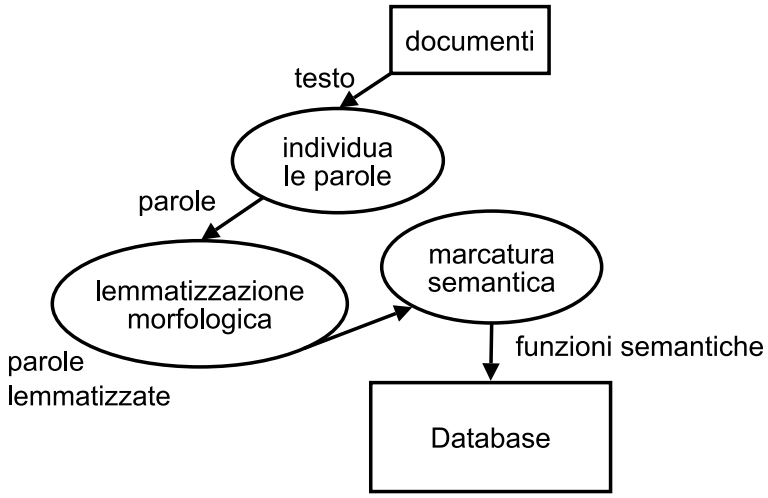

Figura 1.6: Catena di indicizzazione per i testi latini

ma individuata a dei metadati che ne permettono una migliore gestione dal punto di vista del significato: I metadati fanno uso di una base di conoscenza semantica che è stata realizzata per poter consentire una descrizione del contenuto ulteriore rispetto a quella data dal semplice rapporto tra parole e criterio di rilevanza [22]. Da un punto di vista semiotico si tratta della sovrapposizione di un linguaggio formale a una struttura, il testo che viene indicizzato, di linguaggio naturale. In questa sede non verranno approfondite le implicazioni del rapporto tra espressione e contenuto nella struttura che viene prodotta e che sono descritte in parte da Buzzetti (2004):

> La marcatura del testo si comporta come una notazione diacritica e svolge una funzione linguistica essenziale nella rappresentazione digitale del testo. Lo status linguistico del markup è caratterizzato da una ambiguità strutturale costitutiva. Esso, nel medesimo tempo, è parte del testo e dice qualcosa sul testo. Deve essere visto come una parte del linguaggio dell'oggetto, che può essere usato come metalinguaggio per descrivere se stesso.

Altro aspetto fondamentale per la gestione di ricerche su testi in latino medievale è rappresentato dalla variabilità grafico-ortografica che verrà esaminata estesamente nel prossimo capitolo e che necessita di una struttura

[22] Questa parte sarà trattata nei capitoli 4, 5 e 6

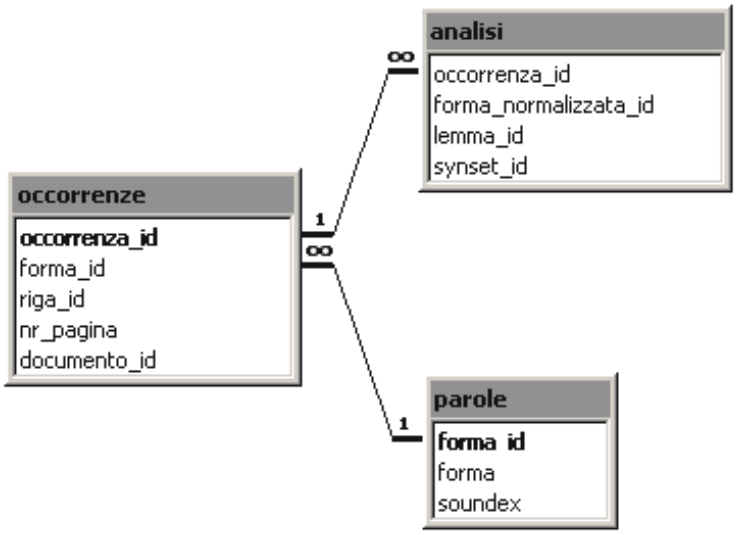

Figura 1.7: Struttura dati indicizzati

di dati atta a separare il livello grafico-ortografico dalla individuazione dei lemmi. La figura 1.7 rappresenta la struttura dei dati indicizzati secondo il metodo sviluppato in questa ricerca: di ogni parola individuata nel testo vengono conservate in struttura indipendente i dati relativi all'*occorrenza*, cioè gli indicatori posizionali come il numero di pagina, di riga e il documento di provenienza, i dati relativi all'*aspetto ortografico*, cioè la *forma* attestata nel testo e la sua *codifica soundex*[23], i dati relativi all'*analisi* lessicale e semantica, vale a dire il riferimento alla forma normalizzata in caso di allografia, il riferimento al lemma di appartenenza e alla collocazione semantica (*synset*) ottenuta attraverso l'ausilio di un *thesaurus semantico*[24]. Questa struttura rende possibile il trattamento dei testi indipendentemente dalla realizzazione ortografica, permettendo ricerche tradizionali attraverso le consuete tecniche di *pattern matching*[25], sia ricerche lessicali a partire dalle forme lemmatizzate, sia ricerche per raggruppamenti concettuali.

I prossimi due capitoli si occuperanno degli aspetti relativi all'analisi lessicale dal punto di vista degli aspetti grafico-ortografici e della lemmatizza-

[23] Cfr. p.31

[24] Alla teoria e alla costruzione di questa struttura sono dedicati i capitoli 3, 4 e 5

[25] Metacaratteri, troncamenti, ricerche per prossimità ecc.

zione, successivamente si approfondiranno gli aspetti collegati alle strutture semantiche.

Capitolo 2

Problemi di gestione degli aspetti lessicali del testo

La costituzione di un sistema di Information Retrieval per il latino medievale porta a compiere una riflessione preliminare sulle caratteristiche di questa lingua, limitandoci al problema dell'utilizzo di testi già editi e adeguatamente convertiti in formato digitale e rimanendo, quindi legati solamente alle peculiarità di carattere lessicale ed ortografico.

In primo luogo appare evidente come in un testo di frasi seguenti, i morfemi adiacenti devono essere interpretati o in maniera congiuntiva come parti della stessa parola o in modo disgiuntivo come parti di parole differenti. Ciò impone un problema nei linguaggi dove le parole non sono perfettamente individualizzate; in questi linguaggi la autonomia della parola non è marcata in senso fonologico da segni segmentali o sovrasegmentali, né lo è grammaticalmente. In questi casi, l'autonomia delle parole non può essere dimostrata da operazioni formali come la sostituzione, la permutazione, l'inserimento o la trasformazione[1]. Nella divisione della frase latina in parole distinte, si deve fare riferimento all'autorità testuale delle fonti utilizzate per ciascun autore e ai dizionari, fatta eccezione per i casi dei suffissi enclitici che potrebbero essere trattati come parole separate.

In secondo luogo, al livello del sistema lessicale, emergono problemi di identificazione e di differenziazione perché esistono numerose forme grafica-

[1]Cfr. Gardner (1971, p.30)

mente identiche che devono essere interpretate come occorrenze di parole diverse, o come varianti di differenti invarianti, cioè come omografi; accanto a queste forme esistono forme graficamente differenti che devono essere interpretate come varianti dello stesso invariante, cioè gli *eterografi* (o con dizione più ampia *allografi*).

2.1 L'eterografia o allografia

Le varianti della stessa invariante possono essere raggruppati insieme dopo un censimento in modo tale da assicurare che non esistano duplicati e ambiguità nell'assegnazione; perché ciò sia possibile è necessaria una operazione di marcatura manuale.

Esistono vari tipi di eterografi:

- eterografi paradigmatici
- eterografi ortografici
- eterografi dialettali
- eterografi stilistici
- eterografi sintagmatici

Eterografi paradigmatici

Flessione

Gli eterografi paradigmatici nelle parole lessicali esprimono le differenze morfologiche delle medesime invarianti e sono proprie delle lingue flessive come il latino. L'identificazione degli eterografi paradigmatici è l'oggetto della lemmatizzazione, di cui si tratterà nel capitolo successivo. L'algoritmo di lemmatizzazione, sviluppato in questa ricerca, opera, in un primo tempo, la *lemmatizzazione morfologica* delle parole, in base all'ausilio di un vasto dizionario e di opportuni meccanismi di *pattern matching*; in una seconda fase, assistita dall'utente, vengono risolti i casi dove è necessaria una disambiguazione sulla base dei contesti ai fini di una *lemmatizzazione morfosintattica*.

Eterografi ortografici

Le varianti ortografiche vanno ricondotte ad un'unico lemma, poiché sono frutto di scelte editoriali e non dipendono dalla tradizione scribale (per esempio la distinzione tra u v e varianti come *jam* e *iam*). Le varianti ortografiche che presentano varianti fonologiche dovrebbero essere documentate, ma ricondotte ad uno stesso lemma (come nel caso di *comprehendo* e *comprendo*).

Scelte editoriali

Eterografi dialettali e stilistici

A causa della relativa mancanza di materiale descrittivo contemporaneo relativamente ai dialetti romani è spesso impossibile accertare se le forme varianti (per esempio forme come *fuere* e *fuerunt*) debbano essere considerate varianti stilistiche. Infatti le forme varianti nella lingua letteraria in genere hanno concomitanti implicazioni stilistiche. In ogni caso, le forme varianti che non riflettono esitazione su una vocale o una consonante o un effetto del *sandhi* devono essere mantenute. Queste varianti possono fornire informazioni interessanti relativamente all'evoluzione del latino medievale nelle forme romanze.

Lingua letteraria

Studio diacronico

Eterografi sintagmatici

Questo tipo di allografo è condizionato dal contesto, dalle parole antecedenti o successive: le regole del *sandhi* producono varianti nella grafia delle sillabe finali e nei confini di parola, riflettendo graficamente l'esito della catena di parole attraverso la contrazione (ecceos, viden, bonast).

Catena parlata

2.1.1 Compressione degli allografi ortografici, dialettali e sintagmatici

Apparentemente l'identificazione degli elementi dell'indice duplicati (o moltiplicati) a causa di grafie alternative potrebbe presentarsi come un tipico problema decisionale: se Σ è l'alfabeto finito utilizzato per la codifica delle parole presenti nei documenti e Σ^* l'insieme delle stringhe finite di simboli di Σ, dato il problema decisionale Π, si individua come D_Π l'insieme di stringhe che codificano un'istanza di Π, i cui elementi sono le forme presenti nell'indice

Problema decisionale?

in esame, e come $Y_\Pi \subseteq D_\Pi$ l'insieme di stringhe che codificano *istanze positive* di Π. Quest'ultimo insieme sarà costituito dalle parole presenti nell'indice in forma allografa. In altre parole, si intenderebbe individuare il linguaggio $L(\Pi)$ costituito dalle stringhe di Σ^* che appartengono a Y_Π:

$$L(\Pi) = \left\{ s \in \Sigma^* \mid s \in Y_\Pi \right\}$$

ciò sarebbe possibile utilizzando una funzione di confronto fra gli elementi appartenenti a D_Π che isolasse le forme che possiedono un allografo.

Fuzzy problem

Ad una più attenta analisi risulta evidente che la creazione di tale funzione attraverso un approccio booleano è impossibile: in primo luogo, le grafie alternative di una stringa, soprattutto per quel che riguarda la situazione dei testi mediolatini, sono frequentemente imprevedibili, pur essendo possibile individuare alcune tendenze[2]; secondariamente, una simile funzione tratterebbe come allografe tutte le forme che costituiscono la flessione di un lemma, restituendo un numero elevato di falsi positivi. Risulta chiaro, inoltre, che decidere se una stringa dell'insieme *è allografa* di un'altra significa sindacare sulla loro similarità. Non necessita di dimostrazione l'affermazione che similarità e differenza tra stringhe si snodano in un continuo, tanto che ogni stringa di un insieme è simile ad un'altra, purché esse abbiano almeno un simbolo (carattere) in comune. Appare evidente che si sta operando nell'ambito di un problema di tipo *fuzzy*[3].

Sia quindi $\chi_{X,Y}$ il confronto di due stringhe appartenenti all'insieme D_Π ed N il numero di stringhe presenti nell'insieme: si avranno $\frac{1}{2}N(N-1)$ operazioni di confronto, per le quali è necessario un algoritmo in grado di restituire un valore indicativo delle differenze tra le due stringhe, per poter successivamente operare raggruppamenti in base ad una soglia di similarità (a) opportunamente scelta. In altre parole, per ciascuna stringa dell'insieme verrà individuato un linguaggio di D_Π che raggruppa tutte le stringhe che

[2]La monottongazione dei dittonghi, l'alternanza delle lettere y e i, l'inserimento occasionale e arbitrario di h, la geminazione o lo scempiamento delle consonanti, gli scambi vocalici e consonantici ecc. Per una descrizione estesa dei fenomeni: Cremaschi (1959); De Prisco (1991); Stotz (1996)

[3]Per quel che riguarda la *fuzzy logic*: Novak (1992); Zadeh (1992)

rientrano nella soglia di similarità:

$$\forall\ s \in D_{\Pi}\ \exists\ A(X) = \left\{ X \in D_{\Pi} \mid dist(X,Y) \leq a \right\}$$

occorre sottolineare che gli insiemi A(X) individuati possono presentare punti di intersezione, problema di cui si discuterà più avanti.

Prendendo in esame i numerosi algoritmi di confronto esistenti[4] è parso opportuno utilizzare l'algoritmo di Levenshtein (1966) conosciuto come *edit distance* ($dist_{Lev}(X,Y)$): esso permette di calcolare il numero minimo di operazioni di cancellazione, inserimento e inversione di simboli necessarie per trasformare una stringa ($X = x_1 \dots x_m$) nella stringa su cui si opera il confronto ($Y = y_1 \dots y_n$), fornendo una valutazione quantitativa della differenza tra due stringhe. La distanza di Levenshtein può essere definita come $\delta(m,n)$: Edit distance

$$\delta(0,0) := 0$$

$$\delta(i,j) := min \begin{cases} \delta(i-1,j)+1 \\ \delta(i,j-1)+1 \\ \delta(i-1,j-1)+costo(x_i,y_j) \end{cases}$$

dove

$$costo(x_i,y_j) := \begin{cases} 0, & \text{se } x_i = y_i \\ 1, & \text{se } x_i \neq y_i \end{cases}$$

La complessità dell'algoritmo è $O(N)$, dovuta alla necessità di operare una ricerca lineare.

Il sistema di confronto, come fino ad ora illustrato, non tiene conto del modello linguistico, pertanto devono essere introdotti alcuni passi correttivi che permettano di tenere conto della flessione: le forme flesse appartenenti al medesimo lemma, senza una correzione di questo tipo, sarebbero considerate tutte allografe tra loro. Inoltre, deve essere individuato un sistema che permetta di abbattere il numero di confronti completi tra le stringhe dell'insieme.

La soluzione qui proposta si serve di una raggruppamento iniziale ope- Soundex

[4]Per una trattazione esaustiva: Knuth (1998); Cormen *et al.* (2001)

rato da un versione opportunamente modificata dell'algoritmo *Soundex*, originalmente sviluppato da Odell e Russell (1918/1922)[5]. L'insieme D_{Π} viene preliminarmente suddiviso in partizioni che raggruppano le stringhe aventi il medesimo codice *Soundex* così ricavato:

1. si isola il primo carattere della stringa, sopprimendo tutte le vocali e la lettera h

2. vengono assegnati i seguenti valori alle lettere rimanenti:

   - b, f, p, v $\rightarrow$ 1
   - c, g, k, q, s, x, z $\rightarrow$ 2
   - d, t $\rightarrow$ 3
   - l $\rightarrow$ 4
   - m, n $\rightarrow$ 5
   - r $\rightarrow$ 6

3. se due lettere con lo stesso codice sono adiacenti, se ne mantiene solo una

4. il codice risultante nella forma *carattere,numero,numero,numero* è dato dalla prima lettera seguita dai primi tre valori numerici individuati; nel caso in cui restino meno di tre cifre, si completa con degli 0 (es: imperator$\rightarrow$i516; lupi$\rightarrow$l100).

I confronti fra stringhe attraverso l'algoritmo di Levenshtein saranno operati solo internamente alle partizioni così costituite e il numero di confronti necessari sarà sempre minore di $\frac{1}{2}N(N-1)$, tranne nel caso in cui tutte le stringhe abbiano il medesimo codice.

Allografie preconosciute

Il confronto puramente computazionale viene affinato grazie ad una tabella di accelerazione, contenente numerose allografie conosciute, che permette

[5]Descritto in Knuth (1998, p.30)

di identificare come equivalenti alcune sequenze di caratteri[6], prima dell'applicazione del confronto algoritmico, che restituirà le probabili allografie non normate. Inoltre, il numero dei falsi positivi viene ridotto attraverso un ulteriore controllo sulla base della parte terminale di ciascuna stringa, identificando le forme che probabilmente appartengono alla medesima flessione: di conseguenza, la forma *rose* sarà correttamente identificata come allografa di *rosae* ma non di *rosis*.

Risultati dell'implementazione

Il metodo illustrato è stato implementato in linguaggio di programmazione Visual Basic[7] e applicato alla collezione dei testi retorici attualmente presente nel database ALIM[8]. Di seguito si dà un campione dei risultati ottenuti: Specimen

```
abarum|avarum
abas|abbas
abati|abbati
abatibus|abbatibus
abatissa|abbatissa
abbas|abas
abbati|abati
abbatibus|abatibus
abbatissa|abatissa
```

[6]es: i fenomeni che investono i dittonghi, *y:i, v:u, cia:tia, cio:tio, cie:tie, x:s, œ:ae, ph:f, y:i, v:u, cia:tia, cio:tio, cie:tie*; o i prefissi *abf-:af-:auf-, abs-:aps-, conb-:comb-, conl-:coll-, conm-:comm-, conn-:con-, conp-:comp-, conr-:corr-, adc-:acc-, adf-:aff-, adg-:agg-, adl-:all-, adm-:amm-, adn-:ann-, adp-:app-, adq-:acq-, adr-:arr-, ads-:ass-, adt-:att-, adgn-:adn-:agn-, adsc-:asc-, adsp-:asp-, adst-:ast-, ecf-:eff-:exf-, exb-:eb-, exd-:ed-, exl-:el-, exrn-:em-, exr-:er-, exs-:ex-, inb-:imb-, inl-:ill-, inm-:imm-, inp-:imp-, inr-:irr-, obc-:occ-, obf-:off-, obg-:ogg-, obrn-:omm-, obp-:opp-, obs-:ops-, obt-:opt-, subc-:succ-:susc-, subg-:sugg-, subf-:suff-, subrn-:summ-, subp-:supp-, subr-:surr-, subs-:sups-:suss- subt-:supt-, subsc-:susc-, subsp-:susp-, transs-:trans-, trans-:tras-* o l'aspirazione *ha:a, he:e, hi:i, ho:o, hu:u, hy:y, ch:c, ph:p, rh:r, th:t*

[7]Cfr. appendice A

[8]`http://www.uan.it/alim/letteratura.nsf`

```
abeo|habeo
ablatiuum|ablativum
ablatiuus|ablativus
ablativum|ablatiuum
abraam|abraham
absencia|absentia...
```

Precisione

L'analisi quantitativa ha evidenziato come su un indice di 22028 forme siano stati individuati 6681 gruppi di possibili allografi, con una percentuale di falsi positivi del 14,9%: ciò permette di valutare l'efficacia dell'algoritmo in termini di precisione, data dal rapporto tra il numero di gruppi individuati e il numero di gruppi rilevanti. Il valore ottenuto (0,851) può essere considerato un ottimo risultato, tenendo conto che si opera prescindendo dal contesto e da informazioni di tipo semantico. Anche nel breve elenco riportato si può notare come sarebbe possibile migliorare la certezza dei risultati potendo operare su base semantica nel contesto-frase: in particolare, le coppie *abbas:abas* e *abeo:habeo* individuate come possibili allografi potrebbero appartenere a lemmi diversi (*Abas, Abantis* in luogo di *abbas, abbatis* e i verbi *abeo* e *habeo*). Mentre una considerazione di tipo frequenziale sconsiglierebbe di ricondurre *abas* al lemma *Abas*, l'ambiguità rimane nel caso della seconda coppia, dove la compressione dell'indice può portare ad una restituzione ambigua di risultati. Ciononostante, si sottolinea come sia tollerabile in strumenti di questo tipo un aumento dei valori di *recall*, anche se ciò può andare a scapito della precisione: i risultati restano in ogni caso più accurati rispetto ad una ricerca che fa uso esclusivo di caratteri jolly, dove il livello di rumore è in genere molto alto.

Ambiguità

2.2 Omografia

Rapporto parola/contesto

L'omografia[9] è la caratteristica di essere ambigue che hanno alcune forme di parola (intese come stringhe di caratteri) qualora siano esaminate fuori

[9]Per uno studio dei fenomeni legati all'omografia dal punto di vista della linguistica computazionale, cfr. Busa (1968, 1987, 1994, 2000); Passarotti e Ruffolo (2004); mentre per l'IR: Hirst (1987); Brown *et al.* (1991); Krovetz e Croft (1992); **?**); Ballesteros e Croft (1998)

contesto, ovvero non incluse nella frase, entro la quale non c'è (o non dovrebbe esserci) omografia. Parliamo di «omografia» e non di «omonimia», in quanto termine più confacente alla linguistica computazionale: esso, infatti, indica una stessa *stringa* di caratteri («omografia») che significa diversamente in contesti diversi. Per «essere ambigua» e «significare diversamente» non intendiamo che una stessa forma di parola ha diversi significati (polisemia), ma che essa non può essere attribuita sempre, a priori, allo stesso lemma fuori contesto (*facies* è forma dei lemmi *facio, -ere* e *facies, -ei*). Inoltre essa può essere portatrice di più valori morfologici contemporaneamente (*puellis* è dativo e ablativo). L'omografia va tenuta distinta dall'*omofonia*: l'omografia riguarda solo la sequenza dei caratteri di una forma, mentre l'omofonia concerne anche la quantità delle vocali presenti in una forma. Esistono parole omografe e omofone (facies) e parole omografe non omofone (*occìdo, -ere* e *òccido, -ere*), mentre non esistono, almeno in latino, parole omofone e non omografe. In ambito di linguistica computazionale, è bene tenere conto solo dell'omografia e non dell'omofonia, evitando di sciogliere la prima ricorrendo alla seconda: le forme *occìdo* e *òccido*, quindi, pur foneticamente diverse, vanno considerate omografe, in quanto solitamente codificate digitalmente con identiche sequenze di caratteri[10].

Tre sono i problemi che pone lo studio dell'omografia latina:

- anzitutto ci si deve chiedere come si articola nel sistema linguistico latino il «significare diversamente» di una stessa stringa di caratteri;
- secondariamente per poter dare una efficace valutazione del fenomeno è necessario chiedersi quanto sia diffuso;
- deve essere inoltre individuata una tassonomia dell'omografia

Una sistematizzazione del genere è necessaria in quanto, per il trattamento computazionale dei dati linguistici, è indispensabile disporre di una modellizzazione delle omografie possibili nel sistema linguistico in vista del perfezionamento dello strumento di lemmatizzazione morfo-sintattica automatica

[10] Ambiguità che potrebbe essere risolta utilizzando una codifica meno ambigua, ma che richiederebbe la riedizione della maggior parte dei materiali editi in forma digitale e cartacea

descritto in queste pagine. Infatti, prima di poter scrivere e applicare regole per la disambiguazione, è utile poter disporre di una recensione completa e di una classificazione esaustiva delle situazioni possibili in cui l'omografia si può presentare, in modo da avere un chiaro e scientificamente preciso quadro del problema che si deve andare a risolvere attraverso l'automatizzazione.

2.2.1 Tipi di omografia in latino

Veniamo quindi a illustrare brevemente una parziale recensione delle tipologie di omografia latina:

omografia esolemmatica

- Omografia tra forme di due, o più lemmi (omografia esolemmatica):
  - di una, o più forme di un lemma con una, o più forme di un altro lemma. Esempio: rosa è nominativo, ablativo e vocativo singolare del lemma di prima declinazione *rosa, -ae* e nominativo ablativo vocativo singolare femminili e nominativo, accusativo, vocativo plurale neutri del participio passato del lemma di terza coniugazione *rodo, -ere*;
  - di tutte le forme di un lemma con tutte le forme di un altro lemma. Esempio: le forme generate dal tema del presente dei verbi *òccido,-ere* e *occìdo, -ere*;
  - causata da enclitiche. Esempio: la forma *tumet* di lemma *tumeo, -ere* (*tum-et*) e di lemma tu (*tu-met*);
  - di una forma appartenente a due lemmi di tema diverso (omografia radicale). Esempio: la forma caro di lemma *caro, -nis* e di lemma *carus, -a, -um*;
  - di una forma appartenente a due lemmi di tema uguale, ma di tipo flessivo diverso. Esempio: tra participi passati e nomi di quarta declinazione, come la forma *intellectus* di lemma *intellectus, -us* e di lemma *intellectus, -a, -um*.
- Omografia tra nome comune/aggettivo e nome proprio. Esempi: i lemmi *clemens, augustus, paulus*.

omografia endolemmatica

- Omografia tra forme dello stesso lemma (*omografia endolemmatica*, o eminentemente morfologica):
  - causata da desinenza. Esempio: la. forma *puella* come nominativo, ablativo e vocativo singolare del lemma *puella, -ae*;
  - causata da enclitiche. Esempio: la forma actione di lemma *actio, -onis* come ablativo singolare (*action-e*), o nominativo e vocativo singolare con l'aggiunta dell'enclitica *-ne* (*actio-ne*).

L'esame computazionale condotto da Passarotti sulle forme contenute nel database CILF[11] ha evidenziato come l'omografia esolemmatica incida massimamente sulle forme costituite da pochi caratteri e che su un campione di 40.014 lemmi analizzati, 12.389 (il 30,96% del totale) sviluppano almeno una forma di omografia esolemmatica.

L'omografia esolemmatica, quindi, prendendo questo dato come statisticamente attendibile può incidere sensibilmente sulla precisione delle ricerche lessicali e dei processi di lemmatizzazione automatica, con una incidenza, nel peggiore dei casi[12], pari a circa il 30%. Le prove effettuate nel corso dello sviluppo dell'algoritmo di disambiguazione hanno mostrato che in una situazione normale l'incidenza media non è mai più alta del 7% del campione di parole che vengono riconosciute. Un dato che comunque resta significativo.

2.3 Trattamento dell'omografia esolemmatica

Omografia esolemmatica

Per far fronte al problema si è studiata la possibilità di disambiguare automaticamente le situazioni di omografia esolemmatica, quei casi, cioè, dove una parola presenti una forma ascrivibile a lemmi di categorie grammaticali differenti a causa dell'omografia.

Poiché non è possibile determinare a priori il lemma o la parte del discorso a cui attribuire una parola fuori contesto, è necessario operare attraverso si-

[11] P. Tombeur, *Thesaurus formarum totius latinitatis - Cetedoc Index of Latin Forms*, Turnhout, 1998

[12] Considerando un campione di testo da analizzare che contenga solo e soltanto le forme ambigue di ciascun lemma

stemi statistici, attraverso un algoritmo che possa sostituirsi alla competenza linguistica, che non può essere riprodotta artificialmente.

Hidden Markov Model

Si è ipotizzato che l'attribuzione della parte del discorso corretta possa essere ottenuta con un margine di errore sufficientemente basso attraverso l'applicazione di un processo probabilistico basato sul modello statistico chiamato *Modello nascosto di Markov*. Si presuppone, quindi, che il sistema della allografia esolemmatica possa essere modellizzato sia un processo markoviano con parametri sconosciuti: l'obiettivo è riuscire a inferire dai parametri osservabili il parametro nascosto. Questo modello è applicato frequentemente in processi di *pattern recognition* e può essere adattato all'analisi automatizzata del discorso.

Modellizzazione

Nel problema esposto i dati che devono essere analizzati sono le categorie grammaticali delle parole che possono essere ricondotte a più di una parte del discorso. Sia quindi (Ω, A, P) uno spazio probabilizzato dove Ω rappresenta l'insieme delle categorie grammaticali della lingua latina, A la tribù delle parti di Ω e P una misura di probabilità su Ω che verrà specificata in seguito.

In un tale spazio si organizza un blocco di variabili aleatorie

$$[C_i]_{1 \leq i \leq k}$$

che, per ogni i, rappresenta la categoria grammaticale della i-esima parola analizzata, essendo k il numero totale delle parole presenti nella frase analizzata (C è la categoria grammaticale della parola i, con i compreso largamente tra uno e il numero totale delle parole della frase). Usando l'ipotesi di Markov si assume che

$$P\{C_i \mid C_{i-i}, C_{i-2}, \dots C_1\} = P\{C_i \mid C_{i-i}, C_{i-2}\}$$

cioè la probabilità che la i-esima parola sia ascrivibile a una determinata categoria grammaticale (C_i) dipende dalle categorie grammaticali delle due parole che la precedono (o, eventualmente, in una modellizzazione più complessa, che verrà discussa più ampiamente nella tesi, dalle categorie delle due parole adiacenti, precedenti o conseguenti). Data la formula della probabilità condizionata:

$$P\{C_i \mid C_{i-1}, C_{i-2}\} = \frac{P\{C_i \cap C_{i-1} \cap C_{i-2}\}}{P\{C_{i-1} \cap C_{i-2}\}}$$

nell'impossibilità di ricavare direttamente la legge di P, deve essere scelto un valore di C_i che massimizzi tale formula.

Perché sia possibile individuare tale valore si ricorre all'analisi automatizzata di un congruo numero di testi raccolti in un *corpus* di natura omogenea, estraendo, attraverso un algoritmo di confronto sulla base di un dizionario completo, l'insieme di tutte le terne che presentano sequenze di parti del discorso attribuibili in maniera univoca e conservando una stima della frequenza. Si ottengono così insiemi di terne accompagnate dalle rispettive occorrenze nel corpus in esame (per esempio Verbo-Preposizione-Sostantivo-80; Aggettivo-Sostantivo-Verbo-56; ecc.). Nel caso di una situazione di ambiguità vengono presi in esame tutti i gruppi che presentino sequenze di categorie grammaticali identiche a quelle che precedono la parola che si deve disambiguare. Alla parola da disambiguare viene assegnata la categoria grammaticale della parola che si trova nella terna più frequente tra quelle considerate.

Corpus

Il procedimento descritto può essere applicato in tutti quei casi in cui sia possibile individuare una sequenza di tre termini tra i quali sia ambigua la categoria grammaticale di uno: il problema viene risolto su base statistica e i risultati sono direttamente influenzati dalla omogeneità delle frasi analizzate con quelle delle strutture presenti nel *corpus* utilizzato come strumento per l'estrazione delle terne.

I dati statistici di partenza sono stati ricavati sulla base dell'analisi automatica dei testi contenuti nel PHI CD-ROM 5.3, che raccoglie testi della latinità classica: la scelta è stata dettata dalla necessità di avere la collezione di testi più vasta possibile per poter ricavare un modello statistico attendibile; non è stato possibile utilizzare come base di partenza solo testi in latino medievale, in quanto le raccolte elettroniche esistenti più corpose sono costituite da testi codificati in modo proprietario e ciò rende impossibile l'estrazione dei dati. La prova, effettuata sulla collezione di testi precedentemente presa in esame per testare l'algoritmo di riconoscimento degli allografi. Gli esperimenti sono stati condotti per un numero complessivo di 25 passaggi che hanno incluso ogni volta porzioni diverse, selezionate casualmente, del campione totale. Il sistema ha permesso di disambiguare in media il 50% delle forme ambigue e uno scarto quadratico medio di 3,87. Il valore piutto-

sto omogeneo tra i vari passaggi ha portato dimostrato anche l'omogeneità della collezione di testi su cui sono state condotte le prove. Il risultato potrà essere ulteriormente migliorato quando si renderanno disponibili raccolte più vaste di testi in latino medievale liberamente accessibili[13].

[13]In particolare fa ben sperare la crescita del progetto ALIM

Capitolo 3

La lemmatizzazione: formalizzazione e aspetti algoritmici

3.0.1 La lemmatizzazione

La lemmatizzazione, secondo la definizione di Busa (1987), è quel complesso di operazioni che conducono a riunire tutte le forme sotto il rispettivo lemma, intendendo per lemma ciascuna parola-titolo o parola-chiave di un dizionario e per forma ogni possibile diversa realizzazione grafica di un lemma. La lemmatizzazione, quindi, consiste nell'attribuire le varianti o flessive (uomini) o grafiche (omo) a una stessa parola (uomo), che funge da lemma: queste varianti sono le forme del lemma.

Definizione

Ciascuna lingua possiede convenzioni di lemmatizzazione proprie: in italiano è uso convenzionale che il lemma verbale sia la forma coniugata all'infinito presente attivo. In latino, invece, il lemma verbale è coniugato alla prima persona singolare dell'indicativo presente attivo: si trova, infatti, sui dizionari la voce *tollo* e non tollere. Il lemma è anche una forma: *edo* è, quindi, sia un lemma (che, da solo, rappresenta tutte le proprie possibili forme), sia una forma. Il contrario non vale: una forma non è necessariamente lemma. Infatti, *edam* è una forma, ma non un lemma.

convenzioni

La lemmatizzazione appare come una pratica facile, se non, addirittura, banale: senza accorgercene e senza sforzo ogni giorno utilizziamo la pratica

Problemi di formalizzazione

della distinzione tra lemma e forma. Tuttavia, al di là di questa apparente intuitività, lemmatizzare richiede l'esercizio di un numero vasto di meccanismi inconsci che devono essere formalizzati qualora si voglia tentare di riprodurli attraverso procedure automatiche.

Esistono due tipi di lemmatizzazione:

- *lemmatizzazione morfologica*: analizza le forme di parole in isolamento, ovvero fuori dal contesto sintattico, fornendone tutti i valori che sono possibili in un dato sistema linguistico. Dal momento che la lemmatizzazione morfologica interessa le parole in sé, ovvero svincolate dalla sintassi, essa resta valida di ciascuna parola sempre e in qualsiasi contesto: ciò è estremamente necessario in informatica umanistica, in quanto è un dato che può essere assunto aprioristicamente, indipendentemente dal testo che, di volta in volta, viene preso in esame.

- *lemmatizzazione morfo-sintattica*: analizza le forme di parole entro il contesto sintattico. Non è mai ambigua, ma sempre univoca, in quanto l'immersione della forma nella frase ne precisa il valore. Quindi, mentre la lemmatizzazione morfologica è indipendente dal testo, la lemmatizzazione sintattica è, invece, legata al testo su cui è applicata.

Perché lemmatizzare

Lemmatizzazione come indice

La lemmatizzazione è, prima di tutto, una forma di organizzazione del materiale lessicale di un testo: essa permette, quindi, di ottenere un filtro ragionato ed estremamente utile alla consultazione. Come già evidenziato, la maggior parte delle ricerche testuali, infatti, ha interesse a reperire tutte le occorrenze di un dato lemma sotto qualsiasi forma si presenti, mentre è certamente più raro il caso che si miri all'indagine di una particolare forma. Ad esempio, il ricercatore ha maggior necessità di individuare quante e quali forme del lemma sembrare occorrono in un testo, piuttosto che di puntare la propria attenzione su una data forma di quello stesso lemma.

Pertanto, la lemmatizzazione permette di disegnare la carta geografica del sistema lessicologico di un testo, primo e fondamentale gradino di qualsiasi analisi linguistico-computazionale.

Indubbia utilità

Come già evidenziato precedentemente[1], analizzando i risultati di una ri-

cerca sul *Patrologia Latina Database*, operare interrogazioni per stringhe di parola (unico mezzo di ricerca possibile, se il testo non è lemmatizzato), provoca una restituzione dei dati imprecisa e rumorosa, con il rischio tutt'altro che remoto di perdere informazione: si pensi a quanto sarebbe articolata e rischiosa la ricerca di tutte le forme di un composto di sum operata attraverso stringhe di caratteri.

Indubbia laboriosità

Che la maggior parte dei testi oggi trasportati su supporto elettronico non sia ancora lemmatizzata è dovuto non tanto a ragioni che giocano a favore di questo uso, quanto a un motivo meramente pratico: la quantità di lavoro umano richiesto dalla lemmatizzazione (in particolare, quella morfosintattica) funge da deterrente.

La maggior parte delle ricerche è interessata più a recuperare, in un testo, tutte le occorrenze di un dato lemma, che non a concentrarsi su ricerche mirate a singole forme di parola. Tuttavia, è probabile il caso di un utente che voglia recuperare tutti i verbi coniugati, ad esempio, al futuro semplice, o alla terza persona singolare. Anche per questo tipo di ricerche, la lemmatizzazione è chiave imprescindibile per accedere ai dati linguistici. Il tipo di lemmatizzazione più utile in casi del genere è, certamente, quella morfosintattica, in quanto essa scioglie le omografie possibili, risparmiando questo lavoro all'utente.

La lemmatizzazione manuale

Evidenziare i problemi relativi alla lemmatizzazione manuale può aiutare a focalizzare gli aspetti operativi che andranno trasformati in processi informatizzati. Un metodo pratico di lemmatizzazione manuale può essere illustrato in otto punti:

Un algoritmo "manuale"

1. Una volta acquisito il testo è necessaria la sua indicizzazione con dei riferimenti (almeno, numero di pagina e di riga).

   Indicizzazione

2. Si produce il sistema grafemico del testo, ovvero una lista di tutte le lettere (differenziate tra maiuscole e minuscole, accentate e non accentate), delle interpunzioni e dei segni grafici presenti.

   Sistema grafemico

[1] Cfr. p. 18

Formario alfabetico e retrogrado

3. Si produce un *formario*, ovvero una lista di tutte le forme presenti nel testo. È preferibile che il formario sia in due redazioni: una elencata alfabeticamente da sinistra a destra (per cui *abacus* precede *rosa*) e una elencata alfabeticamente da destra a sinistra (per cui *rosa* precede abacus, in quanto le due forme sono lette dal computer rispettivamente *asor* e *sucaba*).

Occorrenze

4. Ciascuna forma deve essere seguita dal proprio luogo (o dai propri luoghi) di occorrenza nel testo, ovvero dal numero di pagina e di riga in cui compare.

Cartoncini e regole

5. Si opera la lemmatizzazione morfologica (non andando a controllare nel testo le occorrenze della forma in esame): su dei cartoncini, si riportano la forma (presa dal formario precedentemente prodotto) e il suo lemma. Per evitare ambiguità è utile costituire a parte una lista dei *dubbi*, e delle soluzioni trovate, in modo da uniformare le scelte.

Lemmatizzazione morfosintattica

6. Dopo aver lemmatizzato in questo modo un numero di forme tale da giungere ad aver definito alcune regole di lemmatizzazione, si passa alla lemmatizzazione morfo-sintattica. Questa fase impone il controllo dei contesti di tutte le forme possibilmente omografe listate nel formario, per disambiguarne il valore corretto. È, necessario stabilire quali informazioni devono essere codificate su ciascuna scheda (caso, genere, numero, persona...); decise le informazioni ritenute necessarie, ad ogni forma si attribuiscono una serie di codici: ad ogni posizione di codice si collega un attributo, indicante univocamente una delle informazioni desiderate[2]. I cartoncini di lemmatizzazione devono poi essere raccolti in un apposito contenitore in ordine alfabetico per forma.

Computer come archivio

Certamente anche nella fase di lemmatizzazione manuale il computer può essere utilizzato come strumento di archiviazione, per esempio sostituendo i cartoncini con record di un sistema di gestione dei dati o più semplicemente con delle tabelle separate da spazi.

[2]Ad esempio, se di ciascuna delle forme del testo latino voglio sapere Parte del discorso (1), Declinazione/Coniugazione (2), Modo (3), Tempo (4), Caso (5), Genere (6), Numero (7), Persona (8) e Grado (9), dovrò attribuire a ciascuna forma lemmatizzata 9 codici, in quanto 9 sono le informazioni (attributi) che reputo necessarie in output

7. Conclusa la lemmatizzazione, si crea un *rationarium*, ovvero un documento in cui ciascun lemma è corredato delle seguenti informazioni: Rationarium

   - la parte de discorso del lemma,
   - la frequenza: quante forme diverse del lemma vengono realizzate nel testo,
   - l'occorrenza: quante volte le forme del lemma compaiono nel testo,
   - la lista in ordine alfabetico delle forme del lemma, ciascuna corredata dei codici di lemmatizzazione, del numero delle occorrenze e dei luoghi di occorrenza.

8. infine, si produce un lemmario (lista dei lemmi, ciascuno seguito dalla propria parte del discorso, dalla frequenza e dall'occorrenza) in ordine alfabetico regolare e inverso. A partire da questi dati, è possibile realizzare liste particolari secondo le necessità delle singole ricerche: ad esempio, è possibile produrre la lista di tutte le forme di lemma nominale uscenti in -a e con valore di ablativo singolare maschile. Lemmario

La lemmatizzazione semi-automatica

È possibile supportare il proprio lavoro di lemmatizzazione, facendo uso di un lemmatizzatore automatico, ovvero uno strumento informatico che, ricevendo in *input* un testo, ne lemmatizza morfologicamente un certo numero di forme (la percentuale delle forme lemmatizzate dipende dalla qualità del lemmatizzatore), dando in output tutti i valori che possono essere assunti da quella forma nella lingua in questione[3].

Ad esempio, un lemmatizzatore automatico, ricevendo in input la forma *rosa*, dà in output le seguenti relative alla flessione e alla parte del discorso. Sta al lavoro manuale di disambiguare in contesto quale di questi valori è corretto.

L'utilizzo di un lemmatizzatore di questo tipo ha almeno due vantaggi:

[3]Per il latino classico, per esempio, è disponibile online lo strumento LEMLAT `http://webilc.ilc.cnr.it/~ruffolo/lemlat/index.html` frutto di un progetto ormai più che decennale (Marinone, 1990; Bozzi e Cappelli, 1990; Cappelli e Passarotti, 2003)

- presenta a chi lemmatizza tutti i valori che, in un sistema linguistico, possono essere ricoperti da una data forma di parola;
- lemmatizza una volta per tutte le forme che hanno una sola analisi possibile (ad esempio: puellam, piratam, lupus...).

Tipologie di lemmatizzatori

L'analisi delle forme compiuta da un lemmatizzatore automatico può avvenire secondo almeno due modalità:

- il lemmatizzatore è dotato di un dizionario di macchina comprendente una lista di forme, ciascuna delle quali è seguita dal proprio lemma, o dai propri lemmi e dai valori che essa può assumere. Lo strumento, quindi, ricevendo in input una forma, va a cercarla in questo formario e, se la trova, la lemmatizza, dando in output le informazioni di cui quella forma è corredata nel dizionario-macchina. Questo tipo di archiviazione dei dati ha il vantaggio di autoincrementarsi: ogni volta che si lemmatizza un testo nuovo, le forme non analizzate dal lemmatizzatore e lemmatizzate a mano vengono registrate nel dizionario-macchina e imparate dallo strumento computazionale. Questo sistema ha, però, lo svantaggio di dover collezionare inizialmente una lista di forme che va incrementata dall'utente;
- il lemmatizzatore non è dotato di alcun formario, ma di tabelle contenenti pezzi di parole, ciascuno dei quali è corredato di un codice che ne dice la compatibilità, o meno con gli altri pezzi di parole. C'è, quindi, una tabella di desinenze, una tabella di prefissi, una tabella di radici. Ricevendo in input una forma, il lemmatizzazionezatore la segmenta nelle sue parti e, quando trova una segmentazione compatibile con i dati registrati nelle tabelle, la lemmatizza[4]. Ad esempio, ricevendo in input la forma puellam, un lemmatizzatore di questo tipo trova che la segmentazione *puell-am* è compatibile con i dati registrati nelle sue tabelle, ovvero riconosce la parte *puell* come una radice di un nome di prima declinazione e riconosce la parte am come una desinenza compatibile con una radice di un nome di prima declinazione. Quindi, crea

[4]Questo metodo è fruttuosamente utilizzato per il la lemmatizzazione morfologica del latino classico dal programma LEMLAT (Cappelli e Passarotti, 2003, cfr.)

automaticamente il lemma, aggiungendo la desinenza *-a* alla radice *puell*, in quanto i lemmi di prima declinazione escono regolarmente in *a*. Lo svantaggio di questo sistema è di dover operare in fase di elaborazione un numero elevato di confronti, attraverso l'analisi di tutte le possibilità di confronto.

Nonostante il grosso aiuto fornito da una lemmatizzazione semi-automatica, il lavoro manuale da affrontare per fare una lemmatizzazione morfo-sintattica rimane oneroso. Strumenti che permettano di fare automaticamente una lemmatizzazione morfo-sintattica automatica sono tuttora in fase di studio per quasi tutte le lingue moderne[5], anche se le problematiche di definizione del significato stesso di contesto e delle modalità di riconoscimento sintattico sono quanto mai aperte in quanto come ricorda Wallach (2006) «*text is not a bag of words*».

3.1 Il modulo di lemmatizzazione

Come si è visto precedentemente, è necessario nei sistemi di ricerca lessicale individuare un metodo di compressione degli indici di ricerca, ma nel caso di testi latini non è possibile utilizzare algoritmi automatici di troncamento in quanto ciò deprimerebbe troppo la precisione del sistema e non costituirebbe un significativo progresso rispetto all'utilizzo, sempre possibile, dei metacaratteri. Nel presente progetto di Information Retrieval l'implementazione di un lemmatizzatore automatico costituisce la base di un miglioramento delle funzioni di indicizzazione.

Questo processo di indicizzazione

Svantaggi dei metodi illustrati

Inoltre i due metodi mostrati nel paragrafo precedente presentano alcuni svantaggi in fase di implementazione, che li rendono non applicabili direttamente alle esigenze di un processo di indicizzazione per l'IR: nel primo caso c'è la necessità di dover incrementare la base dati attraverso il suo utilizzo, nel secondo il sistema consuma molte risorse di calcolo per operare i confronti, cosa particolarmente svantaggiosa quando si devono lemmatizzare collezioni di documenti molto ampie, come nel nostro obiettivo.

[5]Per il latino classico è significativo quanto è allo studio da parte di Alberto (2002)

[6]Il metodo di costruzione del lemmatizzatore, pertanto, si è collocato a metà tra i due precedentemente illustrati: dal primo metodo è stata presa la possibilità di gestire un vasto database di forme annotate che contengono le realizzazioni morfologiche della flessione. Questa soluzione permette di minimizzare il numero di confronti necessari ad individuare i candidati d'analisi a cui attribuire la forma in esame. Dal secondo metodo è stato mutuato l'utilizzo delle liste di parti variabili e invariabili dalle quali derivare automaticamente la flessione delle parole.

In pratica, per motivi di rapidità di estrazione dei dati si conservano tutte le possibili forme flesse e la loro flessione è ricavata da una base dati che viene incrementata solamente inserendo le parti invariabili delle parole. Il sistema crea automaticamente le forme flesse in base all'attribuzione della parte invariabile alla parte del discorso indicata dall'utente attraverso un codice.

Alcuni esempi potranno essere utili alla comprensione del processo.

3.1.1 Formalizzazione delle strutture di dati

Categorie morfologiche

Nella seconda appendice vengono mostrati gli elenchi di parti variabili che sono alla base del sistema di declinazione automatica delle parole. Sono stati individuati 7 tipi principali categorie che richiedono un trattamento specifico:

- nomi
- pronomi
- aggettivi
- numerali
- avverbi
- verbi
- participi dei verbi

[6]Caratteristiche della presente implementazione

- supini
- preposizioni
- congiunzioni
- interiezioni

Per ciascuna parte individuata sono state individuate delle modalità di flessione- (*paradigmi morfologici*) che tengono conto di tutte le possibili flessioni di una parola: prendiamo un caso di *rosa*. Essa è stata catalogata all'interno di una base dati contenenti le parti invariabili in questo modo:

paradigmi morfologici

```
LEMMA_ID    31992
stem1       ros
stem2       ros
stem3       #
stem4       #
type        N
decl        1 1
kind        F T
age         X
area        X
geo         X
freq        X
source      0
```

Il codice *LEMMA_ID* è la chiave primaria con cui è identificato univocamente il lemma all'interno della base dati; *stem1* e *stem2* registrano le parti invarianti della parola a cui possono essere unite le parti varianti compatibili[7], *type* la categoria flessiva generale, *decl* indica il codice del paradigma morfologico compatibile, *kind* indica il tipo di parola all'interno della categoria flessiva[8], *age, area, geo* forniscono, quando disponibili, indicazioni sull'utiliz-

[7]Si preferisce la dizione parte variante e parte invariante, in quanto non si tratta esattamente

[8]In questo caso F sta per *femminile* e T per *thing* "cosa"

zo geografico e diacronico della parola, *source* indica il codice del dizionario da cui è stato estratto il lemma[9].

Il sistema di declinazione automatica va a recuperare la tabella contenente il paradigma morfologico corretto e procede all'integrazione degli ulteriori dati presenti in quella tabella:

```
N     1 1 NOM S C  1 1 a        X A
N     1 1 NOM S M  1 2 as       B D
N     1 1 VOC S C  1 1 a        X A
N     1 1 GEN S C  2 2 ae       X A
N     1 1 GEN S C  2 2 ai       B C
N     1 1 LOC S C  2 2 ae       X A
N     1 0 DAT S C  2 2 ae       X A
N     1 0 DAT S C  2 2 ai       B I
N     1 1 ABL S C  2 1 a        X A
N     1 1 ABL S C  2 2 ad       B D
N     1 1 ACC S C  2 2 am       X A
N     1 0 NOM P C  2 2 ae       X A
N     1 0 VOC P C  2 2 ae       X A
N     1 0 GEN P C  2 4 arum     X A
N     1 0 GEN P C  2 2 um       X C
N     1 0 LOC P C  2 2 is       X A
N     1 0 DAT P C  2 2 is       X A
N     1 0 DAT P C  2 4 abus     D B
N     1 0 ABL P C  2 2 is       X A
N     1 0 ABL P C  2 4 abus     D B
N     1 0 ACC P C  2 2 as       X A
```

Organizzazione dei dati morfologici

Il paradigma morfologico richiamato contiene le indicazioni relative non solo al paradigma flessivo regolare, ma anche alle attestazioni storico-linguistiche: pertanto verranno prese in considerazione le possibili uscite registrate nelle varie periodizzazioni[10] del latino, nonché la loro frequenza. Nell'esempio la

[9] O=Oxford Latin Dictionary

[10] Per le attestazioni di grammatica storica e l'individuazione delle parti variabili sono stati usati Ernout (1953); Traina e Perini (1972); Palmer (1977); Stotz (1996); Poccetti *et al.* (1999)

prima colonna riporta il codice della categoria morfologica, il secondo indica il paradigma morfologico (*1 1* è indica i nomi femminili della prima declinazione, ma viene richiamato anche il paradigma *1 0* che indica desinenze comuni), nella terza colonna sono indicati i casi individuati da ciascuna parte variante, *S/P* indicano singolare o plurale, i codici F/M/N/C indicano se la parte invariante va applicata a sostantivi di genere Femminile, Maschile, Neutro o se è Comune ai generi, la settima colonna indica a quale parte invariante va unita ciascuna parte variante, l'ottava colonna indica la lunghezza in carrateri della parte variante[11]. Segue, poi, la parte variante, mentre le ultime due colonne indicano nell'ordine la periodizzazione della desinenza e la frequenza[12].

Indicizzazione e lemmatizzazione

Il sistema di declinazione automatico costruisce successivamente una scheda relativa a ciascuna forma declinata, includendo solo gli elementi compatibili. Periodizzazione e frequenza possono essere utilizzate nel processo di lemmatizzazione dei testi, per adattare le scelte di assegnazione in base all'età dei testi e alla probabilità delle forme.

Il sistema di indicizzazione utilizza poi il metodo del confronto con le parole contenute nella base delle forme flesse per attribuire la corrispondente lemmatizzazione morfologica a ciascuna forma estratta dai testi in esame. In un primo tempo vengono risolti solo i casi di omografia esolemmatica qualora una forma sia attribuita a diverse parti del discorso[13]. La lemmatizzazione-morfosintattica è possibile in un secondo tempo, attraverso una procedura assistita che presenta le funzioni sintattiche individuate come candidate per ciascuna forma all'operatore, perché vengano attribuite in modo univoco.

Risultati dell'implementazione

L'analisi qualitativa operata sulla base di trenta *query* selezionate casualmente in base ai lemmi riconosciuti su un campione di testi selezionati tra quelli presenti in ALIM ha portato ad una precisione del 91,9%, già con la lemmatizzazione morfologica e la procedura di disambiguazione dell'omografia esolemmatica. Il valore potrebbe essere migliorato attraverso l'impiego del processo assistito.

[11] Questo indicatore è introdotto per motivi interni al processo di declinazione

[12] Quest'ultima ricavata limitatamente alle desinenze del latino classico attraverso gli spogli effettuati da Gardner (1971) e Diederich (1939)

[13] Attraverso l'algoritmo statistico descritto precedentemente

Nei prossimi capitoli si tratterà del metodo di ricerca orientato ai contenuti e delle modalità di integrazione nel sistema di una base di conoscenza semantica.

Capitolo 4

Basi di conoscenza per l'IR: *thesauri*, reti semantiche, ontologie

4.1 Thesauri

Uno dei sistemi più intuitivi per migliorare l'efficacia di un sistema di IR è prevedere in esso il supporto di un *thesaurus*: sebbene l'opportunità di incorporare strumenti di questo tipo venga ravvisata dagli studiosi di IR fino dagli anni '60, nessun sistema di IR per testi latini attuale utilizza questa strategia per il miglioramento del richiamo.

Thesaurus: definizione

Con il termine *thesaurus* si intende un dizionario controllato dove vengano identificate relazioni tra le parole; i *thesauri* possono essere suddivisi in due categorie principali: quelli orientati all'archiviazione documentale e quelli terminologici[1]. In un *thesaurus* per l'archiviazione documentale generalmente vengono tracciate solo le relazioni di ipo-iperonimia e sinonimia (termini correlati) [2], nei *thesauri* terminologici vengo presi in considerazione più relazioni relative ad un dominio collegando i concetti definiti in quel dominio con i termini che li realizzano[3].

[1] I confini tra le due tipologie sono stati recentemente descritti in Tsujii e Ananiadou (2005)

[2] In maniera analoga, per esempio, al sistema di archiviazione Dewey

[3] Si illustrerà di seguito quanto il modello di WordNet, che è quello scelto per la realiz-

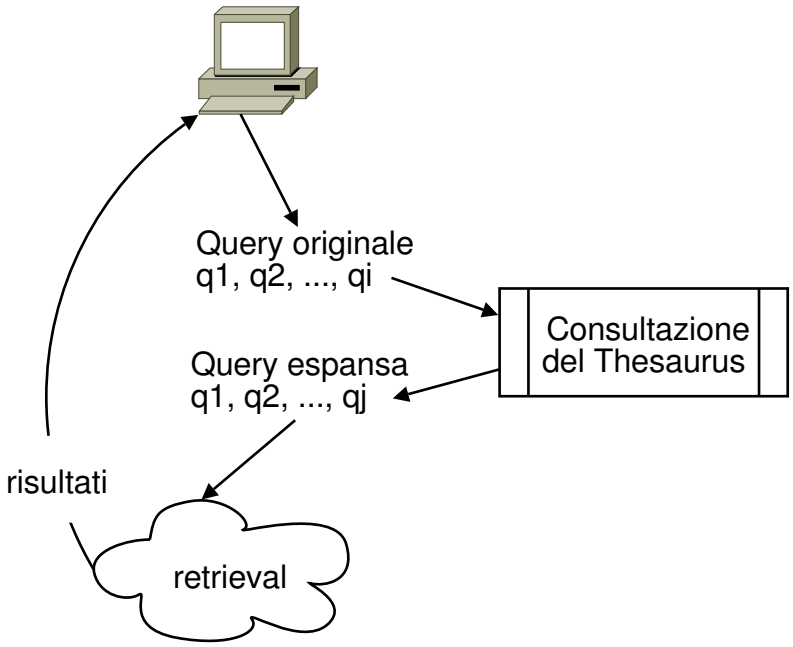

Figura 4.1: Utilizzo di un *thesaurus* per l'espansione di una *query*

Modello

In un sistema di IR assistito attraverso un *thesaurus* la *query* effettuata dall'utente viene riformulata (espansa) attraverso l'inserimento dei termini che nel *thesaurus* risultano essere in relazione con quelli cercati dall'utente (figura 4.1). Poiché raramente due persone descrivono lo stesso concetto con gli stessi termini l'espansione della *query* dovrebbe aumentare le possibilità del richiamo di contesti conformi alle richieste dell'utente. Un documento rilevante per la query dell'utente, infatti, potrebbe non contenere alcuna delle parole presenti in essa contenute: in tal caso un *thesaurus*può essere usato sia per assegnare un termine comune per tutti i sinonimi di una parola, sia per espandere una query per includere tutti i termini sinonimi. Sebbene il funzionamento dello strumento appaia intuitivamente semplice, la modellizzazione dei criteri di attribuzione e di espansione risulta piuttosto complessa, per i noti motivi legati alla semiotica testuale.

Generazione manuale o automatica?

La costruzione di *thesauri* per ambienti di IR può avvenire sostanzialmente in due modi: attraverso l'elaborazione manuale o per mezzo di algoritmi di

zazione della base di conoscenza semantica di questo progetto, si ponga all'incrocio fra le tre tipologie di oggetti: *thesaurus* documentale, terminologico e ontologia

generazione automatica, con quella che, non a torto, è stata definita da Grossman e Frieder (2004, p.123) "the quest for a sort of holy grail of information retrieval"[4].

Generazione automatica

Le metodologie di costruzione automatica sono basate su quattro criteri che qui elenchiamo sommariamente perché una trattazione completa richiederebbe uno spazio ben più vasto:

- **co-occorrenza dei termini**: l'impiego della co-occorrenza come criterio di costruzione di un *thesaurus* è stato trattato inizialmente da Salton e Lesk (1971). L'approccio proposto è quello vettoriale: ciascuna parola viene rappresentata come un vettore e successivamente i termini vengono comparati utilizzando un coefficiente di similarità che misuri la distanza euclidea (o l'angolo) tra i due vettori. Per formare un *thesaurus* per una data parola p le parole correlate di p sono tutte quelle u tali che $SC(p, u)$ sia superiore ad una determinata soglia. Questo processo ha una complessità $O(p^2)$ pertanto la generazione del *thesaurus* viene limitata alle sole parole che presentano una frequenza media.

- **contesto dei termini**: anziché porre l'accento sulla co-occorrenza delle parole all'interno di un intero documento, è possibile sfruttare il contesto (inteso come insieme di parole vicine) per costruire i vettori che rappresentano ciascuna parola; in pratica la similarità tra le parole viene determinata non tanto dal fatto che siano presenti nello stesso documento, quanto piuttosto dalla presenza nello stesso contesto, cioè dall'essere preceduti o seguiti da parole simili.

- **raggruppamento con scomposizione a valore singolo**: un metodo basato sull'espansione in base alle coppie di termini co-occorrenti pubblicato da parte di Schutze e Pedersen (1997)

La costruzione di un *thesaurus* con metodi automatizzati può essere utile laddove si voglia evitare l'utilizzo di numerose basi di conoscenza lessicale dominio-specifiche.

[4]Sulla questione è tornato recentemente e in maniera molto più estesa: vedi libro BNF

La costruzione manuale di *thesauri* è usata tipicamente per la realizzazione di risorse dominio-specifiche. Ghose e Dhawle (1977) sottolineano che la costruzione di questi *thesauri* è particolarmente complessa per quelle discipline dove c'è maggiore disaccordo relativamente al significato dei termini specialistici, come per esempio le scienze sociali. L'uso di *thesauri* generati manualmente è ampiamente descritto da Hines e Harris (1971). Per una descrizione di un metodo per la costruzione di un thesaurus si può prendere a modello l'esempio di Wang *et al.* (1985). Questo modello prevede la costruzione di una serie di thesauri basati sulla relazione IS-A (ipo-iperonimia) tra due termini (*dog* is-a *animal*). I singoli thesauri vengono poi raggruppati in uno che che espliciti sette gruppi di relazioni: l'antinomia, tutte le relazioni eccetto l'antinomia, tutte le relazioni, la meronimia, la co-locazione, tassonomia e sinonimia, relazioni paradigmatiche. L'antinomia identifica i termini in opposizione e la meronimia identifica la relazione parte-tutto. La co-locazione contiene le relazioni tra le parole che occorrono frequentemente nella stessa frase o nello stesso periodo. La tassonomia e sinonimia rappresentano i sinonimi. Le relazioni paradigmatiche mettono in relazione differenti forme di parole che contengono lo stesso nucleo semantico come *canino* e *cane*. Gli esperimenti condotti da Wang hanno dimostrato come, su una piccola collezione di documenti, l'espansione della *query* utente attraverso tutte le relazioni, con l'eccezione dell'antinomia, abbia fatto registrare solo un modesto miglioramento nella precisione e nel richiamo.

D'altro canto lo studio di Kristensen (1993), sulla base di un *thesaurus* contenente tre differenti relazioni,[5] ha rilevato come il richiamo su una vasta collezione di articoli di giornale finlandesi (227.000) presenti, su un numero di trenta *query*, un incremento dal 27% al 100% mentre la precisione sia diminuita soltanto dal 62,5% al 51%. Il *thesaurus* utilizzato conteneva 1.011 concetti e un totale di 1.573 termini.

I contributi fin qui descritti hanno come obiettivo il miglioramento della classifica di rilevanza usando un thesaurus. Lee mostra come l'estensione del sistema di rilevanza booleano attraverso l'inclusione dei dati del thesaurus in una richiesta booleana: i risultati degli esperimenti hanno mostrato la maggiore efficacia di questo approccio (riprenderemo più estesamente il modello

[5]equivalenza (sinonimia), gerarchia (IS-A) e relazioni associative

in quanto il principio è il medesimo adottano nel sistema di espansione delle query su base concettuale).

4.2 Reti semantiche (*Semantic Networks*)

Concetti base

Le reti semantiche sono basate sull'idea che la conoscenza possa essere rappresentata attraverso concetti correlati per mezzo di varie relazioni. Un *network* semantico, dunque, a differenza di un *thesaurus*, pone l'accento sulla creazione di una struttura indipendente dall'espressione e che possa modellizzare i rapporti a livello di contenuto: una rete semantica è quindi composta da un insieme di nodi e archi. Gli archi sono etichettati in base al tipo di relazioni che rappresentano; i dati di fatto relativi ad un determinato nodo, come le sue caratteristiche (colore, dimensione ecc.), sono spesso inseriti in una struttura di dati chiamata cornice (ingl. *frame*). Ciascuna voce di un frame è chiamata zoccolo (ingl. *slot*)[6]. Il *frame* di una rosa può essere così schematizzato:

(**rosa**
    (*ha-colore* rosso)
    (*altezza* 60 cm)
    (*è-un* fiore)
)

In questo caso il *frame rosa* è un singolo nodo di una rete semantica che mostra una relazione IS-A (è-un) con il nodo *fiore*. Gli *slot ha-colore* e *altezza* contengono proprietà individuali della rosa[7].

Caratteristiche e limiti

Fino ad ora sono stati sviluppati numerosi sistemi per la comprensione del linguaggio naturale e per la costruzione automatica di network semantici per la rappresentazione della conoscenza presente in un testo[8]. I problemi più frequenti sono legati alla rappresentazione dei argomenti riguardanti lo

[6] cfr.Minsky (1975)

[7] In questo contesto è d'obbligo il richiamo alla teoria dei database relazionali che utilizzano una modellizzazione molto simile per descrivere i dati trattati. Le reti semantiche specializzano il modello relazionale orientandolo alla rappresentazione dei sistemi linguistici

[8] Per una panoramica si vedano i lavori di

spazio o il tempo: per esempio risulta difficile immagazzinare le informazioni presenti in una frase come "lunedì scorso la rosa è cresciuta di trenta centimetri ed è diventata più alta di tutto nel giardino". Le informazioni che riguardano una caratteristica presente in tempi diversi o la posizione relativa di un oggetto sono difficilmente registrabili in una rete semantica[9].

Reti semantiche e IR

Nonostante i problemi di rappresentazione sino ad ora discussi, le reti semantiche aprono la possibilità di un concreto utilizzo nell'ambito dei sistemi di IR. In particolare, l'impiego di questi strumenti può risultare utile per affrontare i problemi di richiamo dei risultati di una *query* introducendo un livello di astrazione che permetta di superare i limiti delle delle comparazioni operate su stringhe di caratteri: anziché operare un confronto tra i caratteri dei termini di una *query*e quelli presenti in un documento, viene misurata la *distanza semantica* tra i termini. L'idea portante è che i termini che condividono lo stesso significato appaiano relativamente vicini all'interno di una rete semantica.

Senza dubbio esiste una stretta relazione tra i thesauri e le reti semantiche: dal punto di vista di un sistema di IR, un thesaurus può essere usato per espandere una query utente con i termini correlati; una rete semantica ingloba un thesaurus in quanto può rappresentare le relazioni di sinonimia, ma si presenta come un insieme in grado di rappresentare una maggiore complessità di relazioni tra gli elementi collegati.

WordNet

Uno degli esempi più completi di rete semantica[10] è costituito da WordNet[11], un sistema disponibile pubblicamente che contiene *frame* specificamente orientati alla rappresentazione delle parole: a partire dal riconoscimento della natura del tutto accidentale dell'ordinamento dei dizionari attraverso *spelling*, nel modello di *WordNet* le parole sono organizzate per blocchi di significato, denominati *synset*, che raccolgono tutti i lemmi che lessicalizzano lo stesso concetto; i *synset* sono collegati tra loro per mezzo di relazioni che includono,

Relazioni descritte

assieme alla sinonimia, anche l'iponimia, la meronimia e l'antinomia. L'ipoiperonimia mette in relazione significati subordinati e superordinati fornendo

[9] A tale problema, in Lenat e Guha (1989) è dedicata la sezione "Representational Thorns", dove viene descritto *Cyc*, un vasto progetto di rapprentazione della conoscenza

[10] Più avanti si sottolineeranno le analogie con le *ontologie fondazionali*

[11] Miller *et al.* (1990); Fellbaum (1998)

così una struttura gerarchica di concetti. La relazione meronimica induce una gerarchia delle parti sull'insieme dei significati. In questo modo il livello lessicale è chiaramente separato da quello concettuale e questa distinzione è rappresentata dal *medium* semantico-concettuale e dalla relazione semantica che uniscono rispettivamente *synset* e parole. Le relazioni presenti tra i verbi permettono di mettere in luce relazioni di *implicazione* (ingl. *entailment*) e di troponimia. Due verbi sono correlati dall'*implicazione* nel momento in cui il primo verbo implichi il secondo: per esempio la coppia comprare-pagare. La troponimia è la relazione presente nel momento in cui due attività collegate da implicazione avvengono allo stesso tempo: un esempio è la coppia zoppicare-camminare. Il lavoro di Voorhees (1993a) ha mostrato come nel caso dell'espansione delle query attraverso l'utilizzo di WordNet l'ostacolo maggiore sia legato alla specificità delle possibili ricerche, inoltre l'aggiunta di termini aventi un numero elevato di significati può degradare in modo significativo l'efficacia. In Liu *et al.* (2004) è possibile trovare una recente applicazione che ha mostrato attraverso l'utilizzo di WordNet un miglioramento dell'efficacia del 5%.

4.2.1 Formalismi per operazioni sulle reti semantiche

Distanza fra nodi

Per il calcolo della distanza semantica tra i singoli nodi di una rete viene usato un algoritmo di *spreading activation*: un puntatore parte da ciascuno dei nodi iniziali e vengono seguiti i collegamenti finché non si incontra un punto di intersezione; il percorso più breve tra i due nodi viene usato per calcolare la distanza. L'algoritmo semplice del percorso più breve non si applica in questo caso perché potrebbero esserci numerosi collegamenti tra gli stessi due nodi. La distanza tra il nodo a e il nodo b è quindi costituita dal numero minimo di segmenti che separa a e b.

Distanza fra set di nodi

Il calcolo della distanza fra set di nodi si presenta come un problema più complesso. Prendiamo ad esempio le due coppie, formate da un aggettivo e da un sostantivo, "rosa alta" e "fiore grande" in questa situazione alto può essere comparato con grande e rosa con fiore. La difficoltà sta nell'allineare i concetti in modo tale che i concetti correlati siano comparati. Una possibile soluzione è quella fornita da Rada *et al.* (1989).

R-distance

4.2.2 Sviluppare query basate su concetti

Distanza fra nodi

Anziché computare la distanza tra i termini di una *query* e quelli presenti in un documento attraverso la rete semantica e, successivamente, incorporare la distanza nel sistema di misurazione della rilevanza, una rete semantica può essere utilizzata come un thesaurus, con la semplice sostituzione delle parole nella query con quei termini che risultano correlati nella rete semantica. Per rappresentare la query, quindi, al posto di vettori basati sulle parole possono essere generati dei vettori di "concetti". Un algoritmo improntato a questo tipo di approccio è stato descritto da Giger (1988) per migliorare i risultati di un preesistente sistema ricerca booleano. Le parole nel sistema di ricerca originale venivano sostituite dai concetti: questi concetti vengono individuati all'interno di una rete semantica che contiene collegamenti alle parole. A testimoniare la confusione tra i termini thesaurus e rete semantica, Giger nell'articolo si riferisce al sistema con il termine thesaurus, ma le tipologie di relazioni gerarchiche presenti nel modello descritto fanno riportare gli esempi all'ambito delle reti semantiche.

Metodi di confronto

In Chen e Lynch (1992) e Chen *et al.* (1993) viene presentato un altro tipo di approccio che risulta di particolare interesse in quanto basato su una rete generata automaticamente attraverso due algoritmi di raggruppamento (*clustering*): il primo è il consueto algoritmo basato sul calcolo del coseno, mentre il secondo è stato sviluppato dagli autori e si serve di legami asimmetrici tra i nodi della rete semantica. Gli utenti sono in grado di scorrere manualmente la rete per ottenere le parole adatte alla query e, allo stesso tempo, la rete semantica viene usata per trovare termini adatti per indicizzare manualmente nuovi documenti.

4.3 Ontologia

Altro strumento di gestione della conoscenza semantica è l'*ontologia* essa si presenta come il tentativo di formulare uno schema concettuale esaustivo e rigoroso nell'ambito di un dato dominio; si tratta generalmente di una struttura dati gerarchica che contiene tutte le entità rilevanti, le relazioni esistenti fra di esse, le regole, gli assiomi, ed i vincoli specifici del dominio.

L'uso del termine ontologia nell'informatica è derivato dal precedente uso dello stesso termine in filosofia, dove ha il significato dello studio dell'essere o dell'esistere, così come le fondamentali categorie e delle relazioni tra esse.

4.3.1 Uso come glossario di base

Ontologie fondazionali

[12]Una *ontologia fondazionale* è in qualche misura assimilabile ad un glossario di base, anche se al contrario di questo, usualmente la prima è gerarchizzata in due o piú livelli, nei cui termini tutto il resto deve essere descritto. Una analogia può essere vista con il *Basic English*, il dizionario delle 2000 parole della lingua inglese necessarie al dizionario Longman per descrivere le 4000 frasi più comuni nella lingua inglese. Una ontologia fondazionale ha la funzione di un'ontologia di base sia per gli utenti che per i programmi, influenzando la loro prospettiva dei dati e degli eventi.

[13]Una analogia è possibile anche con i linguaggi artificiali. Tutti i programmi per computer si basano su ontologie fondazionali, costituite dal set di istruzioni del processore, dalle librerie di un linguaggio, dai file presenti in un *file system*, o da qualche altra lista di "ciò che esiste". Costruire la rappresentazione di un dominio di conoscenza partendo da delle basi insufficienti può portare a risultati poco corretti, da qui la necessità di disporre di ontologie di base standardizzate[14] e consolidarle come fondamenta del proprio lavoro.

Gruber (1993) ha definito l'ontologia come una specificazione di una concettualizzazione.

Relazioni presenti

Malgrado il termine "ontologia" sia stato utilizzato in modo estremamente generico per contraddistinguere un qualunque schema concettuale di classificazione, una vera ontologia non deve limitarsi ad una gerarchia di concetti organizzati con la relazione di sussunzione ma deve includere anche altre relazioni semantiche che descrivono in che modo i concetti sono interrelati. Una delle relazioni più comuni, oltre a quella di sussunzione, è la relazione di meronimia[15].

[12]Rapporto
con i *thesauri*

[13]Ontologie e linguaggi artificiali

[14]Come la *Dublin Core* per l'SGML `http://dublincore.org/`

[15]Per questo motivo WordNet viene assimilata strutturalmente alle ontologie

Ontologia informatica vs Ontologia filosofica

Questo approccio è differente, anche se correlato, col significato filosofico del termine ontologia, lo studio di ciò che è. Lo scopo di un'ontologia computazionale non è quello di specificare cosa *esiste* e cosa *non esiste*, ma di creare una base di dati, che è un artefatto dell'uomo, contenente concetti riferiti al dominio di indagine dell'ontologo, e che verrà impiegata per eseguire certi tipi di computazione. Per questo motivo, i metodi logici seguiti dagli ontologi in filosofia possono essere utili per individuare ed evitare delle possibili ambiguità, ma quando diverse rappresentazioni ontologiche alternative possono servire ugualmente bene per gli obiettivi dell'ontologo computazionale, i vincoli di tempo impongono normalmente che una alternativa venga scelta, e le altre ignorate. Per certi scopi, è meglio non prendere in considerazione diversi dettagli degli oggetti di interesse. Ne consegue che diverse ontologie computazionali, sviluppate indipendentemente per scopi diversi, per lo stesso dominio di applicazione, possono risultare sensibilmente diverse fra di loro.

4.3.2 Applicazioni nell'informatica

Condivisione della conoscenza

Le ontologie sono applicate comunemente nel campo dell'intelligenza artificiale e nella rappresentazione e nella condivisione della conoscenza. I programmi nei computer possono utilizzare un'ontologia per una varietà di scopi, fra cui il ragionamento deduttivo, la classificazione, diverse tecniche di problem solving, oltre che per facilitare la comunicazione e lo scambio di informazioni fra diversi sistemi.

Un'ontologia che non sia legata ad un particolare dominio di applicazione, ma cerchi di descrivere entità più generali, si definisce ontologia costitutiva, oppure ontologia superiore. In genere è necessario creare degli schemi maggiormente specializzati per rendere i dati utilizzabili in contesti applicativi reali.

Le ontologie costitutive sono importanti per sviluppare, sulla base dei concetti fondanti e delle assiomatizzazioni che contengono, ontologie specializzate che mantengano un disegno integro e coerente.

Sono in corso di studio metodologie specifiche, come *OntoClean*[16], per aiutare gli architetti della conoscenza in questo compito.

[16] `http://protege.stanford.edu/ontologies/ontoClean/`

4.3.3 Ontologie disponibili

Ontologie costitutive possono avere un valore commerciale, creando una competizione per definirle. Murray-Rust (2002) sostiene che questa situazione può portare ad «una guerra nel campo semantico ed ontologico dovuta a diversi standard in competizione», e come conseguenza ogni ontologia costitutiva standard verrà verosimilmente contestata da diverse parti - politiche o commerciali, ognuna con la propria idea di "cosa esiste" (in senso filosofico). Nessuna ontologia superiore è stata finora generalmente riconosciuta come uno standard de facto. Diverse organizzazioni stanno lavorando alla definizione di ontologie standard per specifici domini di applicazione. Il *Process Specification Language* (PSL) creato dal National Institute for Standards and Technology (NIST)[17] è uno di questi esempi.

Competizione

Un'ontologia popolare ed abbastanza esaustiva disponibile è Cyc, un sistema proprietario sviluppato già a partire dal 1985, che consiste in un'ontologia costitutiva e diverse ontologie specializzate per dominio (chiamate microteorie - microtheories in inglese). Un sottoinsieme di questa ontologia è stata rilasciata per uso libero col nome di OpenCyc [18].

Il precedentemente citato WordNet[19] si qualifica come un'ontologia costitutiva perché include sia concetti di tipo generale, sia concetti con un maggior grado di specializzazione, collegati non solo da relazioni di sussunzione, ma anche con relazioni semantiche come quella di meronimia e causa. Tuttavia, a differenza di *Cyc*[20], esso non è stato completamente assiomatizzato.

La Suggested Upper Merged Ontology (*SUMO*[21]) è un altro tentativo di definire un'ontologia superiore, avviato dal gruppo di lavoro IEEE P1600.1, disponibile per uso libero.

Questa iniziativa tende a riservare alcuni termini ed il loro significato per tutti i sistemi basati sullo standard 'P1600.1', nello stesso modo in cui una ontologia generale (in senso filosofico) definisce 'cosa esiste'. Lo stesso uso dell'aggettivo 'superiore', implica una gerarchia che deve essere accettata

[17] `www.nist.gov/psl/`
[18] cfr. `http://opencyc.org/`
[19] `http://wordnet.princeton.edu/`
[20] `http://www.cyc.com/`
[21] `http://ontology.teknowledge.com/`

piuttosto che una base che può essere scelta, e sembra implicare un impatto di tipo culturale.

4.3.4 Una Ontologia o molte ontologie

Realizzazione ambiziosa

La distinzione principale fra l'ontologia in senso filosofico e quella in senso informatico è data dalla pretesa, da parte dell'ontologia filosofica, di spiegare "ciò che è" in assoluto, ovvero tutto l'essere, mentre in informatica la creazione di un'ontologia fondante e totale risulta un'impresa titanica, che richiederà la conciliazione di moltissime esigenze e punti di vista diversi.

Una cosa, invece, già possibile e praticata è la creazione di molte ontologie, ciascuna limitata a un ben preciso dominio e persino ad un ben preciso punto di vista, o scopo, su quel dominio, allora abbiamo qualcosa che può già essere realizzato e può essere utilizzato per molti scopi, come gestire un servizio web o integrare sistemi diversi.

Processi di convergenza

Le ontologie così create potrebbero poi, in caso di necessità, venire mappate le une sulle altre, sfruttando il meccanismo di importazione delle ontologie, in modo da farle interagire senza perdere la complessità e particolarità di ciascuna.

Linguaggi per ontologie

Per essere utili, le ontologie devono essere espresse in una notazione concreta. Un "linguaggio per ontologie" è un linguaggio formale con cui un'ontologia viene costruita. Ciascuna delle ontologie presentate ha sviluppato un suo linguaggio di rappresentazione e proprio la presenza di un linguaggio descritto e variamente implementabile differenzia queste iniziative da quelle più propriamente legate alla costruzione di reti semantiche.

Nel prossimo capito si vedrà come una struttura che si colloca trasversalmente alle definizioni di *thesaurus*, rete semantica e ontologia, cioè il modello di WordNet, può risultare utile nella costruzione di uno strumento di analisi semantica per la lingua latina.

Capitolo 5

La costruzione di una base di conoscenza lessicale latina: il progetto Wordnet latino

In questo capitolo si esporremo il metodo scelto per creare uno strumento di gestione della conoscenza semantica per il latino, a partire da modelli esistenti realizzati per le lingue moderne.

Modelli di realizzazione

Esistono almeno due modelli per creare una WordNet multilingue. Il primo modello, adottato dal progetto EuroWordNet (EWN), consiste nel costruire reti semantiche indipendenti le une dalle altre, cercando in una seconda fase di trovare corrispondenze tra loro [1]. Il secondo modello, adottato dal progetto MultiWordNet (MWN), consiste nel costruire le reti semantiche specifiche per un linguaggio mantenendo il più possibile le relazioni semantiche disponibili nella WordNet di Princeton (PWN). Ciò viene ottenuto costruendo i nuovi *synset* in corrispondenza con i *sysnset* della PWN, ogni volta che ciò sia possibile, e importando le relazioni semantiche dai corrispondenti *synset* inglesi; in questo modo si ipotizza che se esistono due *synset* nella PWN e una relazione che li collega, la stessa relazione leghi i corrispondenti *synset* in una lingua diversa. Secondo Vossen (1996), il modello di MWN (o modello a espansione, expand model) sembra meno complesso e garantisce lo stesso

[1] Per un'analisi delle tecniche si veda Vossen (1996)

grado di compatibilità tra differenti wordnet[2]. Per constatare questo fatto basta considerare che la costruzione di qualsiasi rete semantica necessariamente implica un gran numero di decisioni soggettive (e discutibili). Così se due reti semantiche sono costruite indipendentemente per due diverse lingue, mostreranno differenze che dipendono solo parzialmente dalle differenze tra le due lingue: alcune non banali discrepanze strutturali dipenderanno infatti da scelte soggettive o da criteri di costruzione differenti. Il modello di MWN minimizza queste differenze aderendo strettamente ai modelli di costruzione di PWN.

Migliore allineamento

Inconvenienti

Il modello MWN presenta anche degli inconvenienti potenziali: il rischio più serio è quello di forzare una eccessiva dipendenza sulla struttura lessicale e concettuale di uno dei linguaggi coinvolti (Vossen, 1996, p.718). Questo rischio può essere scongiurato permettendo alla nuova wordnet di divergere, quando necessario, dalla struttura di PWN.

Un altro importante vantaggio del modello MWN è che possono essere utilizzate procedure automatiche per velocizzare la costruzione dei *synset* corrispondenti e per l'individuazione delle divergenze tra PWN e la wordnet che si sta costruendo. In tutte queste procedure la stessa PWN può essere usata utilmente come risorsa.

LatinWordNet

La costruzione di LWN (LatinWordNet) è basata principalmente su due procedure automatiche. La prima può essere chiamata Procedura di assegnazione. Dato il significato di una parola latina la procedura di assegnazione produce una lista pesata dei *synset* di PWN che più probabilmente corrispondono al significato. Tale lista è poi utilizzata dal lessicografo per costuire il *synset* latino. La seconda procedura, fornisce l'individuazione dei gap lessicali (procedura LG), cioè di quelle situazioni in cui un concetto lessicalizzato in un linguaggio non ha un corrispondente nell'altro linguaggio.

5.1 La procedura di assegnazione

Seguendo il modello MWN, il nostro obiettivo è quello di costruire, ogniqualvolta sia possibile, un *synset* latino che sia sinonimo (semanticamente

[2]Utilizzo il minuscolo per indicare una implementazione diversa dall'originale WordNet

corrispondente) con i *synset* di PWN. Se ciò non è possibile, si è individuato una idiosincrasia Inglese-a-latino o Latino-a-inglese.

Strategie di assegnazione

I *synset* sinonimi italiani possono essere costruiti seguendo due differenti strategie:

- La prima strategia è basata sui traducenti dall'inglese al latino. Per ciascun *synset* di PWN S, cerchiamo un gruppo di traducenti che siano i sinonimi delle parole inglesi di S. Se non è possibile costruire nessun *synset* sinonimo italiano di S si è trovata una idiosincrasia lessicale inglese-a-italiano.

- La seconda strategia è basata sui gruppi di traducenti latino-a-inglese. Per ciascun senso σ di una parola latina L, si cerca un *synset* di PWN che includa almeno un traducente inglese di L e si costituisce un legame tra L e S. Quando la procedura è stata applicata a tutti i significati della parola latina, possiamo costruirre la classe di equivalenza di tutti i gruppi di parole latine che sono state collegate con lo stesso *synset* di PWN. Ciascun gruppo nella classe di equivalenza è il *synset* latino sinonimo con alcuni *synset* di PWN. Se per un gruppo di sinonimi latini non c'è alcun *synset* sinonimo in PWN, si è trovata una idiosincrasia lessicale latino-a-italiano.

Il miglior allineamento tra la WordNet di Princeton e quella latina può essere ottenuto utilizzando entrambe le strategie per cercare di validare i risultati incrociandoli.

Creazione dei collegamenti

Trovare collegamenti tra i significati delle parole latine e i *synset* di PWN è un processo complesso e lungo, anche se è sempre molto più rapido rispetto alla costruzione da zero dei *synset* latini, della loro organizzazione in una rete semantica e del metterli in corrispondenza con i *synset* di PWN. Per ciascun significato latino, il lessicografo dovrebbe cercare i gruppi di traducenti equivalenti in un dizionario bilingue, trovare tutti i *synset* che contengono questi traducenti equivalenti, valutare con attenzione il significato di questi synset (sinonimi, glosse, relazioni semantiche) e, infine, decidere quale synset di PWN, se esiste, è sinonimo del significato latino della parola. Per alcuni significati di parola il lessicografo potrebbe dover valutare decine di *synset* di PWN.

Per aiutare il lessicografo nel suo lavoro è stata realizzata una procedura che sceglie, per ciascun significato di una parola latina, i *synset* di PWN che più probabilmente hanno un significato compatibile. Nel miglior caso la procedura sceglie solamente il candidato corretto, e il lessicografo deve solamente confermare la selezione. Nel peggiore dei casi la procedura trova solo candidati erronei o non può trovare alcun candidato e il lessicografo deve operare manualmente. Nella maggior parte dei casi la procedura trova una rosa ristretta di candidati che includono quello corretto, e il lessicografo deve confermare la scelta corretta e rifiutare quelle errate. In altre parole l'algoritmo aiuta il lessicografo a focalizzare quale sia il *synset* di PWN più adatto.

Procedura-assegnazione

La procedura-assegnazione prende come *input* uno dei sensi della sezione latino-a-inglese del dizionario di macchina e fornisce in *output* un gruppo di candidati, ciascuno dei quali è descritto da una coppia del tipo *<PWN synset, punteggio di certezza>*, dove *punteggio di certezza* (PC) misura il grado di certezza nel legame tra il significato della parola latina e il *synset* di PWN. Solo i candidati con un PC più alto di una certa soglia vengono proposti al lessicografo. Scegliere il livello di soglia è una questione di bilanciare precisione e richiamo (vedi capitolo relativo all'information retrieval in generale). Maggiore è la soglia, minore è la probabilità che candidati erronei siano proposti (alta precisione), ma è anche maggiore la possibilità che la scelta più idonea non sia inclusa nel gruppo dei candidati (basso richiamo).

Per un determinato significato di parola listato nella nel dizionario latino-inglese, la procedura-assegnazione considera il gruppo di parole inglesi che vengono proposte come traducenti equivalenti per quel significato e trova tutti i synset contenenti almeno un traducente equivalente. Questi synset costituiscono il gruppo di candidati (GCand) che deve essere collegato con il significato di parola latina dell'*input*. Possiamo riassumere il primo passo dell'algoritmo dicendo che esso calcola i GCand del significato di una determinata parola latina. Il resto dell'algoritmo consiste nell'ordinare i GCand calcolando il PC di ciascuno dei suoi *synset*.

Ordinamento

L'ordinamento dei GCand è basato su una serie di regole per stabilire i legami: ogni regola, se applicata con successo a un candidato, alza il suo PC. Si deve notare che il *PC parziale* contribuito da ciascuna regola varia

a seconda di fattori specifici alla regola. Accanto al dizionario di macchina, vengono utilizzate dalle regole anche altre risorse, come la sezione italiana di Multiwordnet e un dizionario italiano-latino, un dizionario dei sinonimi latini e la stessa PWN.

Le regole di costruzione dei legami possono essere divise in quattro gruppi principali a seconda del principio su cui sono basate: *probabilità generica*, *traduzione incrociata*, *corrispondenza della glossa* e *intersezione di synset*. Regole di costruzione

Probabilità generica La regola di probabilità generica si basa sulla supposizione che solo un elemento nel GCand è il corretto candidato per legare il senso di una parola latina. Di conseguenza si può supporre che maggiore è la cardinalità del GCand, minore è probabilità che ciascun candidato sia quello esatto. La cardinalità del GCand dipende da grado di ambiguità delle parole che sono proposte come traducenti equivalenti del significato della parola di *input*. Se c'è un solo *synset* nel GCand, ciò significa che tutti i traducenti equivalenti della parola di input sono monosemici: è quindi altamente probabile che l'unico *synset* nel GCand sia sinonimo del significato della parola di *input*[3].

Traduzione incrociata Questa regola si basa sulla supposizione che se colleghiamo un significato di parola al corretto *synset* attraverso un traducente equivalente. Quindi è probabile che almeno alcuni dei sinonimi del traducente, presenti PWN, abbiano la parola parola di input come traducente equivalente inglese-latino. Per esempio l'italiano puntura: qaundo riferito a insetti, il Collins lo traduce come sting. Sting, però, appartiene a 4 *synset* di PWN: sting, stinging; pang, sting; sting, bite, insect bite; bunco, bunco game, sting. Solo il terzo *synset* è sinonimo della parola italiana. Se guardiamo ai sinonimi di *sting* nel terzo *synset* possiamo trovare che la sezione inglese-italiano dà puntura come traduzione di bite. Riassumendo, la regola della traduzione incrociata considera i sinonimi presenti in PWN di un traducente che crea il collegamento e calcola un PC parziale che è proporzionale al numero di sinonimi che hanno la parola italiana come traducente dall'inglese all'italiano.

[3]Cfr. il criterio monosemico usato da Atserias *et al.* (1997)

Corrispondenza della glossa Un gruppo di regole di collegamento sfrutta le informazioni contenute nella glossa italiana che introduce la maggior parte del dizionario di macchina. La glossa può contenere un campo semantico specifico, un sinonimo, un iperonimo, o una specificazione di contesto d'uso. Queste informazioni possono essere utilizzate in vario modo.

L'informazione relativa al campo semantico è sfruttata grazie ad una risorsa sviluppata parallelamente a MWN, cioè la marcatura di tutti i *synset* di PWN con una etichetta relativa al campo semantico[4]. La glossa del dizionario contiene una etichetta relativa al campo semantico e se questa etichetta corrisponde a un *synset* individuato come candidato, allora il candidato ottiene un maggiore PC. Le varianti nelle etichette dei campi semantici sono gestite attraverso un tabella di corrispondenze.

Quando le glosse contengono parole o frasi, si cerca un corrispondente tra di esse e le parole contenute nelle glosse di PWN. Per fare ciò, si estraggono i lemmi delle parole inglesi delle glosse, e si controlla la loro presenza nelle glosse del traducente equivalente in PWN. La forza della corrispondenza dipende dal grado di ambiguità del traducente. Maggiore è la polisemia, minore è il peso attribuito alla corrispondenza.

Il meccanismo ha due estensioni basate sul fatto che le glosse spesso specificano il genere della parola che stanno definendo al posto di un sinonimo. La prima estensione cerca una corrispondenza tra una parola latina e un iperonimo del suo traducente equivalente. Il secondo meccanismo cerca una corrispondenza tra una parola latina e una parola inglese contenuta nella glossa di un iperonimo del *synset* candidato. Se la corrispondenza tra la parola latina e la parola inglese viene ottenuta attraverso uno dei meccanismi indiretti il PC parziale sarà più basso rispetto all'individuazione diretta.

Intersezione di *synset* Questa regola sfrutta il fatto che i gruppi di traduzione possono includere più traducenti equivalenti, che sono ovviamente

[4]Cfr. Magnini e Cavaglià (2000)

sinonimi. Se uno dei traducenti equivalenti è ambiguo, possiamo usare gli altri traducenti equivalenti per disambiguare. In pratica la regola prende i differenti gruppi di candidati che sono accessibili attraverso diversi traducenti equivalenti e li interseca. I *synset* che sono nell'intersezione ottengono un PC. Per esempio la parola italiana pilastro è tradotta nel suo senso metaforico come pillar, mainstay. La parola pillar appartiene a 5 *synset* di PWN, mentre mainstay appartiene a tre *synset*. C'è però un solo synset che li contiene entrambi.

Valutazione

Per evidenziare la performance dell'algoritmo si è operata una valutazione basata sui nomi presenti sotto la lettera D del dizionario latino-inglese (la lettera è stata scelta a caso). I gruppi di traduzione sono stati presi in esame come come stima del numero dei significati presenti per i quali l'algoritmo dovrebbe essere in grado di trovare alcuni *synset* candidati. Abbiamo selezionato i candidati con un PC più alto di una determinata soglia, cioè quei candidati che vengono proposti al giudizio del lessicografo. Il numero di tali candidati è 89% del numero di sensi presenti nel dizionario. Dopo il controllo da parte del lessicografo abbiamo calcolato precisione e richiamo dei candidati selezionati dall'algoritmo. La precisione è del 70% calcolata come rapporto tra il numero di candidati accettati e numero di candidati proposti dall'algoritmo. Il richiamo ammonta al 63% calcolato come rapporto tra il numero di candidati accettati e numero dei significati listati nel dizionario.

5.2 La procedura di individuazione dei gap lessicali

Gap lessicali e analisi contrastiva

La letteratura sull'analisi contrastiva mostra che, dati un *linguaggio sorgente* e un *linguaggio bersaglio*, posso sussistere vari tipi di idiosincrasie al livello lessicale. Tra le varie idiosincrasie che possono verificarsi nel livello lessicale solo alcune sono rilevanti per il tipo di informazione codificato all'interno di MWN, che segue strettamente il criterio di costruzione di PWN. In MWN, un *synset* di un linguaggio $L1$ contenente unità lessicali $p_1, \ldots p_n$ ha un corrispondente in un altro linguaggio $L2$ se esistono una o più unità lessicali in $L2$ che sono sinonimi translinguistici di $p_1, \ldots p_n$. Ne consegue

che solo due tipi di idiosincrasie implicano la mancanza di corrispondenza translinguistica in MWN[5]:

- *differenze denotative*
- *gap lessicali*

Le *differenze denotative* si hanno nel momento in cui un traducente equivalente della lingua sorgente esiste ma è più generale o più specifico rispetto al senso della parola nella lingua bersaglio.

Il significato di *gap lessicale* necessita di una spiegazione più estesa, pertanto, di seguito descriveremo la nozione di unità lessicale e di gap lessicale, successivamente mostreremo i passi di una procedura che automaticamente classifica i traducenti di un dizionario elettronico bilingue in tre gruppi: unità lessicali, gap lessicali e traducenti equivalenti che necessitano di essere classificati manualmente come unità lessicali o gap lessicali.

5.2.1 Che cos'è un gap lessicale

Una delle idiosincrasie più comuni, particolarmente rilevante dati i criteri di costruzione di PWN, sono i gap lessicali.

Un gap lessicale si manifesta ogni volta che una lingua esprime un concetto con una unità lessicale laddove l'altra lingua esprime lo stesso concetto con una libera combinazione di parole (Hutchins e Somers, 1992). Seguendo il criterio di costruzione di PWN una unità lessicale può essere costituita da una *parola singola*, o da un *idiotismo* o da una *collocazione ristretta* (Cowie, 1981):

- un *idiotismo* è una frase fatta il cui significato non può essere ricavato attraverso la composizione di dei significati delle parole che la compongono. Inoltre le parole componenti non possono essere sostituite da sinonimi.
- una *collocazione ristretta* è una sequenza di parole che abitualmente co-occorrono e i cui significati possono essere derivati in modo compositivo. Le *collocazioni ristrette* hanno una coesione semantica dovuta

[5]Cfr. Bentivogli e Pianta (2000)

principalmente all'uso, perciò la sostituzione delle parole componenti è fortemente limitata. Di solito le collocazioni ristrette non hanno una traduzione letterale in altri linguaggi. Per esempio l'italiano senso unico corrisponde all'inglese one way.

- una *combinazione libera* è una combinazione di parole che seguono solamente le regole generali della sintassi: gli elementi non sono legati specificamente gli uni agli altri e per questo possono ricorrere liberamente con altri elementi lessicali.

5.2.2 Individuare i gap lessicali

Nella costruzione della rete semantica LWN è stata introdotta una procedura per identificare i gap lessicali in modo semi-automatico: su questo modello è stata realizzata una procedura analoga per la costruzione della rete semantica latina. Tale procedura si basa sull'utilizzo di un dizionario bilingue di medie dimensioni che include circa 38.000 lemmi e 55.000 traducenti.

Idiotismi e collocazioni

La procedura distingue tra idiotismi, collocazioni e libere combinazioni (che implicano gap lessicali). Nella pratica i confini tra idiotismi, collocazioni ristrette e combinazioni libere non sono nettamente delineati. Comunque, in molti casi può essere operata una distinzione in base alla conoscenza contenuta nei dizionari che marcano in modo esplicito idiotismi e collocazioni. Inoltre, tutti e tre i gruppi mostrano determinate regolarità strutturali che possono essere sfruttate automaticamente per distinguere gli uni dagli altri con un buon grado di affidabilità.

La procedura di ricerca dei gap lessicali classifica tutti i gruppi di traducenti del dizionario bilingue in tre classi: unità lessicali, gap lessicali e gruppi di traducenti che devono essere controllati manualmente.

Uso dei gap

L'informazione riguardante i gap lessicali può essere usata in due modi, a seconda se si abbia a che fare con gap dal Latino all'Inglese o vice versa. I gap Latino-a-Inglese mostrano una serie di *synset* latini che devono essere aggiunti manualmente a LWN: si è certi che questi *synset* non possono essere costruiti in corrispondenza di nessun synset inglese e quindi la loro costruzione non può essere affidata ai risultati della procedura-assegnazione. Vice versa le informazioni relative a gap Inglese-a-Latino mostrano *synset*

specifici di PWN che possono essere esclusi a priori da quelli scelti dalla procedura-assegnazione.

Si deve sottolineare che tale procedura è rilevante anche da un punto di vista teoretico. Infatti essa fornisce una ulteriore stima quantitativa dei gap lessicali, mostrando quanto le due lingue siano compatibili e fornendo una base empirica al modello della rete semantica multilingue.

5.3 Il modello di dati di MultiWordNet e di LatinWordNet

Il modello di dati MultiWordNet riflette i principali elementi teorici della rete semantica multilingue. Il database è costruito sull'idea che esiste un gruppo di dati comuni a tutte le lingue e altri specifici di ciascuna lingua. Nell'implementazione le relazioni *semantiche* di PWN sono contenute in un modulo chiamato COMMON-DB, mentre le relazioni *lessicali* per il latino e per l'inglese sono contenute in altri due moduli LATIN-DB e ENGLISH-DB. In altre parole l'informazione relativa a quali lemmi appartengano ai *synset* è contenuta nei database delle lingue, mentre l'informazione relativa alle relazioni tra i synset, che rimangono costanti tra le lingue, è contenuta nel COMMON-DB. Un altro fondamentale gruppo di informazioni, cioè la corrispondenza tra i *synset* realizzati nelle diverse lingue, è ottenuta utilizzando lo stesso identificatore di *synset* nelle diverse lingue. Tutti i synset di lingue diverse che hanno lo stesso identificativo appartengono al medesimo *multisynset*. Il COMMON-DB descrive le relazioni tra i multisynset di MWN. Tutte le informazioni semantiche che sono indipendenti dalla lingua possono essere aggiunte al COMMON-DB[6].

Separazione e indipendenza dei dati

Sì è mostrato come il modello di dati di MWN rappresenti le costanti concettuali presenti in lingue differenti. Tale modello di dati, inoltre, evidenzia anche le divergenze semantiche tra le lingue[7]. Inoltre, anche se si mantengono le relazioni semantiche evidenziate da PWN come base del COMMON-DB, è possibile aggiungere nuove relazioni o modificare quelle esistenti. La pos-

Possibilità di ampliamento

[6] In particolare le relazioni relative ai campi semantici

[7] Nella fattispecie i gap lessicali

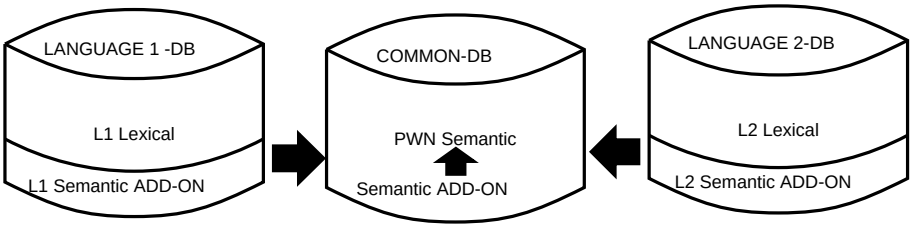

Figura 5.1: Il modello di dati di MultiWordNet (Pianta *et al.*, 2002)

sibilità di modificare le relazioni semantiche di PWN e di rappresentare le idiosincrasie concettuali nei linguaggi specifici è stata implementata attraverso dei moduli aggiuntivi che sovrascrivono, senza modificarli fisicamente, i dati originali di PWN. Il COMMON-DB infatti contiene tutte le relazioni semantiche originali di PWN e una risorsa chiamata COMMON-ADD-ON che ne riscrive un parte. Ciascuna lingua contiene un language-ADD-ON che specifica le relazioni semantiche che sono proprie di quella lingua. La figura 5.1 sintetizza le principali caratteristiche del modello di dati MWN. Le frecce rappresentano le relazioni che vengono sovrascritte. All'interno del COMMON-DB, i dati di PWN sono sovrascritti da una risorsa semantica comune (COMMON ADD-ON), mentre il COMMON-DB viene sovrascritto dalla risorsa semantica aggiuntiva (Semantic ADD-ON) di ciascun database linguistico.

Specificità linguistiche

Le peculiarità lessicali sono codificate all'interno delle aggiunte specifiche di ciascuna lingua. Se c'è prova che la lessicalizzazione di un determinato concetto manchi in una lingua, nella sezione lessicale del database di quella lingua viene inserita una etichetta vuota per quel nodo[8]. Per la rappresentazione delle differenze denotative e dei gap lessicali vengono seguite due diverse strategie: se il nodo vuoto corrisponde a una differenza denotativa, una o più relazioni vicine vengono usate per collegare il nodo ad un *synset* più generico o a molti synset più specifici. Se il nodo vuoto corrisponde a un gap lessicale, viene riportata nella glossa del nodo vuoto una parafrasi di traduzione appropriata, preceduta dalla parola chiave *TE* (*Translating*

[8]Il termine *nodo* è usato in quanto MWN si compone su una struttura di reticolare, dove i lemmi sono inseriti come nodi e le relazioni semantiche costituiscono i collegamenti tra i nodi

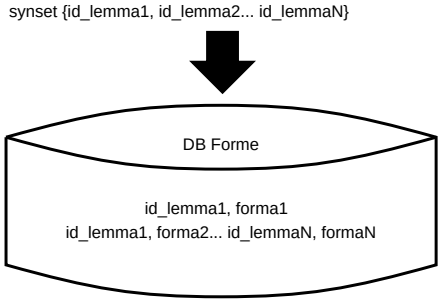

Figura 5.2: La rappresentazione dei lemmi nei *synset* è astratta rispetto alla realizzazione ortografica

Equivalent). Le relazioni più vicine vengono inserite nella risorsa linguistica aggiuntiva specifica della lingua in questione.

Ciascun database linguistico contiene anche un modulo con informazioni lessicografiche relative ai collegamenti tra i sensi delle parole e i *synset*.

Per quel che riguarda le relazioni, tutte quelle semantiche sono state importate da PWN e sono disponibili assieme alle relazioni più vicine, cioè le nuove relazioni specifiche di ciascuna lingua che sono state aggiunte nella MWN per rappresentare le differenze denotative.

Implementazione

L'attuale implementazione della parte latina di MWN si basa sull'aggiunta di un modulo in grado di rendere indipendente il livello grafico/ortografico dall'individuazione dei lemmi (figura 5.2). In pratica per ciascun lemma di dizionario è stata introdotta una grafia normalizzata, associata ad un numero espandibile di grafie alternative. Nei *synset* della parte latina non vengono registrati direttamente i lemmi ma dei codici identificativi: in questo modo possono essere utilizzate diverse grafie per la rappresentazione dello stesso lemma, e sono inoltre collegate all'interno del *synset* anche tutte le realizzazioni morfologiche della flessione dei lemmi.

L'implementazione della base dati è stata effettuata attraverso un database relazionale in modo da permettere l'interfacciamento con il sistema di IR in modo versatile sfruttando le possibilità di consultazione anche attraverso un ambiente distribuito.

Di seguito, sarà descritta la modalità di utilizzazione nell'ambito dell'Information retrieval semantico della rete costruita.

Capitolo 6

IR semantico: un modello per i testi mediolatini

6.1 Sfruttamento del modello WordNet per l'IR

Lo sfruttamento come base di partenza per ricerche di tipo semantico e per il miglioramento del processo di ricerca delle informazioni in una collezione di testi può avvenire utilmente in due forme: in primo luogo i *synset* di WordNet possono essere usati per rappresentare il contenuto dei documenti, in luogo delle parole; secondariamente la rete di WordNet, utilizzata come *thesaurus* semantico, può essere utilizzata come fonte per l'espansione automatica della *query* utente.

6.1.1 Descrizione dell'informazione semantica

Come evidenziato precedentemente, uno dei problemi delle tecniche di reperimento dell'informazione basate sulla corrispondenza di parole è determinato dal fatto che la corrispondenza tra parole e concetti non è una funzione in senso matematico. Nel caso di omografi le parole che sembrano uguali rappresentano concetti diversi, nel caso dei sinonimi, invece, due parole distinte rappresentano lo stesso concetto. Gli omografi rappresentano un ostacolo in quanto diminuiscono la precisione creando *falsi positivi* e i sinonimi diminuiscono i valori di richiamo in quanto si presentano come *falsi negativi*. Nel costruire questo sistema di reperimento dell'informazione si

Falsi negativi e Falsi positivi

parte dall'assunto che l'efficacia degli algoritmi di ricerca dovrebbe migliorare se il confronto non viene operato direttamente sulle parole ma sui concetti che le parole rappresentano.

Approccio strutturato e non strutturato

Questo tipo di confronto fra concetti è stato sperimentato per le lingue moderne in diversi modi. I vocabolari controllati, come si è illustrato precedentemente, sono usati di frequente in quei sistemi che fanno affidamento sull'indicizzazione manuale dei documenti e in genere hanno un termine di descrizione canonico per un dato concetto. Il confronto fra concetti è, per esempio, il cuore del sistema di IR SCISOR (Rau, 1987) che utilizza alcune tecniche di comprensione del testo; in questi sistemi strutture di significati sono utilizzate per rappresentare i concetti e su quelle strutture operano sofisticati algoritmi di confronto. Accanto a questi approcci fortemente strutturati su basi dati di conoscenza lessicale esistono anche tentativi meno legati ad un modello *knowledge-based*: tra questi il sistema basato sull'astrazione di parole in contesti presenti nel modello di indicizzazione semantica latente di Deerwester *et al.* (1990).

WordNet: Uso della base semantica

Le basi di conoscenza lessicale costruite sul modello di WordNet, come è stato fino ad ora mostrato, forniscono un altro modo per definire un concetto: il *synset*. Lo sfruttamento dell'informazione semantica contenuta in WordNet è stato investigato in vari modi, soprattutto per quel che riguarda l'interazione tra *synset* e concetti nelle operazioni di ricerca testuale. Sussna (1993) e Richardson (1994) trattano la struttura creata dai puntatori relazionali di WordNet come una rete semantica e definiscono alcuni modelli di misurazione per calcolare la distanza tra *synset*. La somiglianza tra una *query* e un documento viene poi calcolata dalla similarità tra il gruppo di *synset* della *query* e i *synset* presenti nei documenti. Quest'ultimo tipo di operazione è particolarmente esosa in termini di risorse di calcolo a causa della sua natura combinatoria: vengono infatti individuate tante coppie di *synset* quanti sono gli elementi della *query* e quanti sono i documenti e devono essere valutati tutti i percorsi possibili tra ciascuna coppia.

Metodi di espansione

Per contenere e ottimizzare il numero di confronti negli esperimenti di Chakravarthy (1994) e di Chakravarthy e Haase (1995) usato WordNet per trovare corrispondenze quando sia le *query* sia i documenti sono brevi e strutturalmente prevedibili. Questo interessante approccio è stato applicato all'in-

dicizzazione delle didascalie in collezioni di immagini. La rappresentazione delle *query* e delle didascalie viene realizzata automaticamente e identifica il ruolo delle parole nel testo: il confronto individua corrispondenze se le parole della *query* e quelle della didascalia sono collegate semanticamente nel *database* lessicale e rivestono lo stesso ruolo nei rispettivi testi.

Un terzo tentativo di confronto tra concetti e *synset* è motivato dall'obiettivo di migliorare l'efficienza del sistema di ricerca, mantenendo la robustezza e l'affidabilità del modello vettoriale[1] (Voorhees, 1993b). Il focus di questo approccio è dato dato da una procedura di indicizzazione completamente automatica progettata per scegliere un solo *synset* per ciascuna parola nel testo. Il risultato di questa procedura di indicizzazione è un vettore nei quale alcuni dei termini rappresentano i *synset* anziché le parole. Una volta creato un vettore basato sui *synset* esso viene gestito esattamente come uno basato sulle parole.

Un metodo solo automatico

Disambiguazione dei significati

Nella procedura di indicizzazione è necessario individuare un sistema per determinare la corretta assegnazione dei significati. A tale proposito, ormai classico è l'esempio di Salton e Lesk (1971) che, prendendo in esame il gruppo di sostantivi costituito da *base*, *bat*, *glove* e *hit*, dimostra come, sebbene ciascuna di queste parole prese singolarmente si presenti come polisemica, quando sono utilizzate insieme nello stesso contesto si riferiscono chiaramente al gioco del *baseball*.

Nel caso di WordNet Latino, per sfruttare questa idea in modo automatico, è necessario definire un gruppo di categorie che rappresentino i diversi significati di una parola. Una volta strutturata questa categorizzazione viene contato il numero di parole presenti nel testo da indicizzare che appartengono a ciascuna categoria; il significato che corrisponde alla categoria maggiormente rappresentata viene attribuito alle parole ambigue.

Gruppi di categorie

Nel caso di WordNet si usa per la definizione delle categorie un costrutto chiamato *tetto* (*hood*): un *hood*, come suggerito da Miller *et al.* (1990), è

Generazione automatica delle categorie

[1] v. infra per l'illustrazione del modello vettoriale e del suo utilizzo nell'ambito della presente ricerca

un'area di WordNet nella quale una stringa non è ambigua. Più precisamente, per definire un *hood* di un dato *synset*, s, si considera il gruppo di *synset* e il legame di iponimia di WordNet come un insieme di vertici e bordi diretti di un grafo. L'*hood* di s è il più grande sottografo che contiene s, include solo discendenti di un antenato di s, e non contiene alcun *synset* che abbia un discendente e che includa un'altra istanza di un membro di s come membro.

L'algoritmo proposto per la disambiguazione usa l'*hood* dei *synset* che contengono una parola ambigua w, per definire le categorie che rappresentano i diversi significati di w. Il significato di w in un testo particolare può essere selezionato contando il numero di altre parole presenti nel contesto che sono associate a ciascuno degli *hood* di w, scegliendo l'*hood* con il punteggio più alto. L'idea dietro questo procedimento è di individuare prima di tutto una tendenza generale del rapporto tra *hood* e parole nella collezione di documenti e, successivamente, identificare le deviazioni più significative nei singoli testi.

Disambiguazione tramite *hood*

Un procedura di marcatura che visiti i *synset* e mantenga un conteggio del numero di volte in cui ciascun *synset* viene visitato è fondamentale per entrambi i livelli del processo di disambiguazione. Data una parola w, la procedura di marcatura trova tutte le istanze di w nei synset della base semantica. Per ciascun *synset* identificato, la procedura segue il puntatore dell'iperonimia fino alla radice della gerarchia, incrementando il contatore per ciascun *synset* che viene visitato. Nella prima fase la procedura di marcatura è richiamata una volta per ciascuna occorrenza di parola in tutti i documenti della collezione. Viene registrato il numero di volte che avviene il processo di ricerca della parola, producendo un gruppo di *conteggi globali* (relativi all'intera collezione) per ciascun *synset*. Nella seconda fase la procedura di marcatura viene chiamata una volta per ogni occorrenza di parola all'interno di ogni testo individuale (sia esso un documento o una *query*). Di nuovo si registra il numero totale di volte che la procedura viene eseguita. Questa procedura individua il numero di *conteggi locali* per ciascun *synset*. Dati i conteggi globale e locale, il significato di una parola ambigua w è determinato in questo modo:

La procedura di marcatura

La differenza

$$\frac{numerovisitelocali}{numerototaledichiamatenellafase2} - \frac{numerovisiteglobali}{numerototaledichiamatedellafase1}$$

è calcolata per ciascun *hood* di ogni significato di w. Se un significato non ha un *hood* o se il conteggio locale è inferiore a 2 (cioè non sono presenti altre parole che puntano allo stesso significato) il risultato della sottrazione è considerato 0. Se un significato ha *hood* multipli la differenza è quella massima tra quelle calcolate su ciascun gruppo di *hood*.

Misura di similarità

Come strumento per la ricerca, onde individuare la similarità tra la *query* utente e i documenti presenti nella base dati, al sistema di *information retrieval* della rete semantica è stato applicato il modello vettoriale di Salton *et al.* (1975). In questo modello i documenti e le *query* sono rappresentati da vettori in uno spazio N-dimensionale dove N è il numero di lemmi distinti presenti nella collezione di documenti. Il processo automatico di lemmatizzazione contribuisce ad una compressione dell'indice e a una attribuzione di pesi alle parole indicizzate in maniera direttamente proporzionale al numero di occorrenze della parola nel testo e inversamente proporzionale al numero di documenti in cui la parola è presente. Data una *query* il sistema produce una lista ordinata di documenti in base alla similarità rispetto alla *query*. La misura della similarità è data dal coseno dell'angolo formato tra il vettore della *query* e il vettore che rappresenta il segmento di testo che chiamiamo documento. Se d_i è il peso del termine i nel vettore D e q_i è il peso del corrispondente termine nel vettore della query Q la misura della similarità tra il documento e la query è:

$$sim(Q, D) = \cos(Q, D) = \frac{\sum_{i=1}^{N} q_i d_i}{\sqrt{\sum_{i=1}^{N} q_i^2 \sum_{i=1}^{N} d_i^2}}$$

Il contesto minimo su cui opera l'indicizzazione e quindi la costruzione dei vettori è la frase, intesa come sequenza di simboli tra due segni di interpunzione forte. I vettori, vengono costruiti in base all'assegnazione delle parole contenute nella frase ai *synset* della rete semantica: in questo modo la frase viene considerata come una sequenza di *synset*, vale a dire marche che puntano ad un'area di significati. Ciò consente migliorare l'efficienza del sistema di ricerca.

6.1.2 Espansione di *query* attraverso i *synset*

Rapporto tra query e polisemia

La pratica sperimentale dimostra che la precisione di una query non viene depressa in maniera evidente dalla presenza di omografi in quei casi in cui la *query* sia sufficientemente articolata (Krovetz e Croft, 1992): se un documento possiede un sufficiente numero di termini in comune con una *query*, presentando quindi alta similarità con la richiesta dell'utente, il contesto di ciascuno dei due testi è sufficientemente simile da fare sì che le parole potenzialmente polisemiche puntino al medesimo significato. Inoltre, è esperienza comune di quanti utilizzano sistemi di IR che è molto più deleterio il caso di un sistema di ricerca che produca falsi negativi, piuttosto che quella di un meccanismo che incorpori pochi falsi positivi facilmente individuabili: il più contiene il meno.

Espansione della query utente

Queste considerazioni hanno portato a sperimentare nel corso della ricerca un secondo metodo di sfruttamento della rete semantica durante la fase di indicizzazione. Anziché cercare di selezionare un singolo synset per rappresentare un concetto, con il risultato di ottenere, in caso di assegnazioni erronee, una degradazione dell'efficienza di richiamo, si è pensato di espandere attraverso WordNet Latino l'*input* fornito nella *query* dell'utente, con l'obiettivo di aggiungere nella ricerca tutte le parole che possono essere usate per esprimere il concetto cercato. Qualsiasi parola usata per esprimere un concetto in un documento rilevante troverà una corrispondenza nella *query* e le parole restanti avranno scarso impatto sulla similarità generale. Questo processo, che abbiamo già illustrato nel capitolo relativo alle basi di conoscenza lessicale e che va sotto il nome di *query expansion*, aveva dato prova di poter migliorare l'efficacia delle ricerche su piccole collezioni di testi omogenei (Salton e Lesk, 1968; Wang *et al.*, 1985) e ciò costituiva una base di partenza per una possibile implementazione del metodo su altri tipi di collezione, strada tuttavia non ancora percorsa per testi in lingua latina. Usare la rete semantica come fonte di espansione delle query ha portato anche a verificare l'applicabilità del metodo a collezioni più vaste che coinvolgono diversi domini di conoscenza.

Metodo

Per investigare l'efficacia dell'espansione attraverso la rete semantica, sono stati condotti due gruppi di esperimenti, sulla scorta di quanto suggerito

in Voorhees (1994). Dapprima ci si è concentrati sulla strategia stessa di espansione, attraverso l'utilizzo di *query* che facevano uso di *synset* scelti appositamente come punto d'inizio del processo di espansione: ciò ha permesso di individuare per ciascuna *query* il punto massimo di performance su cui confrontare i risultati di un processo di selezione completamente automatico. La scelta manuale dei *synset* da includere nella *query*, infatti, permette di rimuovere gli inevitabili errori di una selezione di parole poco significativa. Un secondo gruppo di esperimenti, poi, ha esteso la strategia di espansione includendo un algoritmo di selezione automatica dei *synset* di partenza.

Espansione per mezzo di *synset* selezionati dall'utente

Dopo che il testo della *query* è stato annotato manualmente per mezzo dei *synset*, l'indicizzazione e l'elaborazione della *query* sono automatiche. Le parole presenti nella *query* vengono cercate sulla base del processo di compressione determinato precedentemente dall'algoritmo di lemmatizzazione morfologica e in una prima fase della ricerca l'algoritmo di confronto cercherà solo i termini presenti nella *query*.

Modalità di espansione

In un secondo momento viene richiamata la procedura di espansione che dato un *synset* può operare in vari modi: è possibile aggiungere solo le parole appartenenti al *synset*, oppure tutte le parole collegate gerarchicamente per iponimia o per iperonimia, o tutte le parole che sono collegate al *synset* aventi la distanza di un unico arco indipendentemente dal tipo di legame. La procedura di espansione è stata parametrizzata per facilitare la comparazione dell'efficienza delle varie possibilità. Il gruppo di parametri impostato determina per ciascun tipo di relazione presente in WordNet la lunghezza massima della catena di legami che deve essere seguita nel processo di espansione: poiché i *synset* sono tra loro collegati gerarchicamente e relazionalmente è necessario stabilire un punto di ingresso e un punto di arresto del processo di espansione. Tutti i sinonimi contenuti in ciascun *synset* della catena di espansione vengono aggiunti alla query, facendo uso del sistema di descrizione indipendente dalla realizzazione morfologica, che è stato descritto nella parte relativa al problema della lemmatizzazione: la *query* ottenuta cercherà le parole aggiunte utilizzando i codici di lemmatizzazione, in modo tale da

restituire le occorrenze indipendentemente dalla funzione morfologica.

Utilizzo delle relazioni

Per esemplificare il processo di espansione considerato il *synset mobile* si assume che ci possa essere un rapporto di meronimia con il synset *legno e legname*. Se i parametri del processo di espansione sono settati per seguire solo i collegamenti di iponimia, una occorrenza di *mobile* farà aggiungere ai termini di ricerca i codici di *tavolo desco tavola*. Se invece l'espansione avviene attraverso l'aggiunta di qualsiasi link con profondità 1, la parola *mobile* farà includere nella *query* anche *tavolo commercio legname articolo legno*.

Le varie aggiunte alla query possono essere tenute separate nel processo di ricerca attraverso diversi sottovettori. Ciascun vettore è composto da 11 sottovettori: 1 per i termini originali, 1 per i sinonimi e uno per ciascuno dei puntatori relazionali contenuti nella sezione dei sostantivi della rete semantica. Poiché i vettori del documento contengono solo un sottovettore la similarità generale è calcolata per queste *query* è la somma ponderata delle similarità per ciascun sottovettore presente nella query, calcolata contro il singolo vettore rappresentato dal documento.

Altre strategie di espansione

Le strategie di espansione possono essere molteplici: espansione solo attraverso i sinonimi, espansione attraverso i sinonimi e i legami di parentela nella gerarchia, espansione attraverso i sinonimi e tutti i *synset* direttamente collegati (aventi distanza 1 all'interno della catena dei collegamenti).

Gli esperimenti condotti hanno portato a rilevare come il livello di precisione delle query espanse sia direttamente proporzionale al numero di parole specifiche presenti nella query originale: l'espansione attraverso i *synset* è stata combinata in modo da operare sull'intersezione degli insiemi costituiti dai risultati delle *query* individuali. L'intersezione dei risultati ha portato ad un aumento del richiamo nella maggior parte dei casi senza una eccessiva degradazione della precisione.

Espansione completamente automatica

Il successo ottenuto nell'espansione di semplici query con il metodo precedente ha portato a verificare l'ipotesi di una espansione non assistita. Poiché sebbene sia possibile presentare all'utente una lista di possibili *synset* per

l'espansione di ogni *query*, spesso la scelta diventa un processo laborioso e tedioso. Pertanto è stato possibile elaborare un processo ricorsivo che per ciascuna parola di una *query* ne analizza prima di tutto la ricorrenza nel set di documenti: se la ricorrenza è più bassa del numero dei documenti della collezione espande tutti i *synset* contenenti la parola e per ciascuna parola aggiunta che ricorre in più di una lista aggiunge tutte le parole collegate.

Ciò permette di selezionare i significati più importanti perché in questo modo le parole che si aggiungono alla *query* devono essere collegate con più di una parola presente nella *query* originale, individuando un'area di significati omogenei.

Gli esperimenti condotti hanno mostrato che c'è un discreto miglioramento dei valori di richiamo, con una decremento della precisione lievemente più alto rispetto ai risultati della *query* dove i *synset* vengono espansi attraverso l'intervento dell'utente: su 30 query selezionate i valori di restituzione del campione risultano aumentati tra il 15% e il 25% rispetto alle query non espanse, con una precisione che si mantiene intorno al 70% e un richiamo, rispetto alla collezione di testi individuati come rilevanti, per le query di circa il 74%.

Conclusioni

Alla fine della descrizione del percorso di ricerca, intendo sottolineare che ci si rende conto che la capacità di rappresentazione e, di *produzione* di senso propria dell'oggetto testo non è limitata, e non si esaurisce, nella sua dimensione lessicale, ma si estende anche nella dimensione sintattica e nelle capacità attributive del soggetto, a tal proposito, rimando a quanto detto precedentemente, introducendo il lavoro, sulla necessità di operare una *scelta di scala* nella costruzione del modello.

Il percorso fin qui descritto ha portato alla realizzazione di una serie di strumenti di analisi automatica del testo, come aspetto applicativo (o, se vogliamo, come sottoprodotto) della ricerca: il sistema di identificazione delle allografie e il meccanismo di indicizzazione e lemmatizzazione descritto nel relativo capitolo. Si segnala inoltre che recentemente è stato rilasciato al pubblico dominio il un dizionario latino compatibile con il *software* WinEdt per il sistema di editoria elettronica LaTeX: il dizionario composto da una lista di 1243950 forme latine, permette il controllo ortografico dei testi ed è utilizzabile, con lievi adattamenti, in applicazioni di riconoscimento dei caratteri, per migliorare la qualità dei risultati di acquisizione.

Inoltre il progetto del *thesaurus* semantico descritto nella presente tesi di dottorato ha incontrato l'interesse dell'Istituto Trentino di cultura e si è concretizzato in una convenzione interdipartimentale tra il Dipartimento di Linguistica Letteratura e Scienze della Comunicazione dell'Università di Verona e la Divisione di ricerca per le Tecnologie Cognitive e della Comunicazione del Centro per la Ricerca Scientifica e Tecnologica di tale Istituto (IRST-ITC).

Al termine di questo percorso, che ha voluto proporre un metodo per il rinvenimento automatico del significato, mi siano concessi un'ultima breve

citazione e un augurio. Prima la citazione:

> So d'una regione barbarica i cui bibliotecari ripudiano la superstiziosa e vana abitudine di cercare un senso nei libri, e la paragonano a quella di cercare un senso nei sogni o nelle linee caotiche della mano... Ammettono che gli inventori della scrittura imitarono i venticinque simboli naturali, ma sostengono che questa applicazione è casuale, e che i libri non significano nulla *di per sé*.

L'augurio è che i libri possano sempre, e comunque, significare *per noi*.

Appendice A

Prima Appendice

In questa parte si presentano alcuni esempi di codice di programmazione in linguaggio VisualBasic che costituiscono la realizzazione informatica di quanto espresso teoricamente nei capitoli precenti.

A.1 Algoritmo di confronto degli allografi

```
Private Sub Command1_Click()

'Vengono istanziati gli oggetti ADO Connection e Recordset Dim cn As
New ADODB.Connection, rs As New ADODB.Recordset Dim percorso As
String Dim soundx As String Dim allografi As String Dim parola1 As
String Dim parola2 As String Dim destra1 As String Dim destra2 As
String

'Viene creata la connessione al DataBase MDB cn.Open
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
"\parole.mdb"

fnum = FreeFile() Open "allografi.txt" For Output As #fnum

For i = 0 To List1.ListCount - 1
       If List1.Selected(i) And Len(List1.List(i)) > Text2.Text Then

'Se la parola è selezionata carica le parole con medesimo soundex
soundx = getSoundex(List1.List(i))

Select Case Left$(soundx, 1)

Case "V", "B"
query = "SELECT word.word_word FROM word WHERE
word.soundex='B" & Right$(soundx, 3) & "' OR word.soundex='V" &
```

```
Right$(soundx, 3) & "' ORDER BY word.word_word;" rs.Open query, cn,
adOpenStatic, adLockReadOnly, adCmdText

Case "H"
query = "SELECT word.word_word FROM word WHERE
word.soundex='H" & Right$(soundx, 3) & "' OR word.soundex='A" &
Right$(soundx, 3) & "' OR word.soundex='E" & Right$(soundx, 3) & "'
OR word.soundex='I" & Right$(soundx, 3) & "' OR word.soundex='O" &
Right$(soundx, 3) & "' OR word.soundex='U" & Right$(soundx, 3) & "'
OR word.soundex='Y" & Right$(soundx, 3) & "' ORDER BY
word.word_word;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

Case "A"
query = "SELECT word.word_word FROM word WHERE
word.soundex='A" & Right$(soundx, 3) & "' OR word.soundex='H" &
Right$(soundx, 3) & "' OR word.soundex='Æ" & Right$(soundx, 3) & "'
ORDER BY word.word_word;" rs.Open query, cn, adOpenStatic,
adLockReadOnly, adCmdText

Case "E" query = "SELECT word.word_word FROM word WHERE
word.soundex='E" & Right$(soundx, 3) & "' OR word.soundex='H" &
Right$(soundx, 3) & "' OR word.soundex='Æ" & Right$(soundx, 3) & "'
ORDER BY word.word_word;" rs.Open query, cn, adOpenStatic,
adLockReadOnly, adCmdText

Case "Æ"
query = "SELECT word.word_word FROM word WHERE
word.soundex='E" & Right$(soundx, 3) & "' OR word.soundex='A" &
Right$(soundx, 3) & "' OR word.soundex='Æ" & Right$(soundx, 3) & "'
OR word.soundex='H" & Right$(soundx, 3) & "' ORDER BY
word.word_word;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

Case "I", "Y"
query = "SELECT word.word_word FROM word WHERE
word.soundex='I" & Right$(soundx, 3) & "' OR word.soundex='H" &
Right$(soundx, 3) & "' OR word.soundex='Y" & Right$(soundx, 3) & "'
ORDER BY word.word_word;" rs.Open query, cn, adOpenStatic,
adLockReadOnly, adCmdText

Case "O"
query = "SELECT word.word_word FROM word WHERE
word.soundex='O" & Right$(soundx, 3) & "' OR word.soundex='H" &
Right$(soundx, 3) & "' ORDER BY word.word_word;" rs.Open query, cn,
adOpenStatic, adLockReadOnly, adCmdText

Case "U"
query = "SELECT word.word_word FROM word WHERE
word.soundex='U" & Right$(soundx, 3) & "' OR word.soundex='H" &
Right$(soundx, 3) & "' ORDER BY word.word_word;" rs.Open query, cn,
```

```
adOpenStatic, adLockReadOnly, adCmdText

Case Else 'tutti gli altri casi
query = "SELECT word.word_word FROM
word WHERE word.soundex='" & soundx & "' ORDER BY word.word_word;"
rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText End
Select

'Scorre il recordset per stampare gli allografi
        rs.MoveFirst
            Do Until rs.EOF
            If Not (List1.List(i) = rs("word_word")) Then

            parola1 = List1.List(i)
            parola2 = rs("word_word")
            parola1 = Replace(parola1, "x", "s")
            parola1 = Replace(parola1, "æ", "ae")
            parola1 = Replace(parola1, "ph", "f")
            parola1 = Replace(parola1, "y", "i")
            parola1 = Replace(parola1, "v", "u")
            parola1 = Replace(parola1, "cia", "tia")
            parola1 = Replace(parola1, "cio", "tio")
            parola1 = Replace(parola1, "cie", "tie")
            parola2 = Replace(parola2, "x", "s")
            parola2 = Replace(parola2, "æ", "ae")
            parola2 = Replace(parola2, "ph", "f")
            parola2 = Replace(parola2, "y", "i")
            parola2 = Replace(parola2, "v", "u")
            parola2 = Replace(parola2, "cia", "tia")
            parola2 = Replace(parola2, "cio", "tio")
            parola2 = Replace(parola2, "cie", "tie")

            destra1 = Right$(parola1, 1)
            destra2 = Right$(parola2, 1)

'stampa l'output: il rumore è ridotto sulla base dell'ultima lettera
         If destra1 = destra2 Then
         If LD(parola1, parola2) < Text1.Text Then
         allografi = allografi & "|" & rs("word_word")
         End If

         If (destra1 = "d" And destra2 = "t") Then
         If LD(parola1, parola2) < Text1.Text Then
         allografi = allografi & "|" & rs("word_word")
            End If

            If (destra1 = "t" And destra2 = "d") Then
            If LD(parola1, parola2) < Text1.Text Then
            allografi = allografi & "|" & rs("word_word")
            End If
```

```
            If (destra1 = "p" And destra2 = "f") Then
            If LD(parola1, parola2) < Text1.Text Then
            allografi = allografi & "|" & rs("word_word")
            End If

            If (destra1 = "f" And destra2 = "p") Then
            If LD(parola1, parola2) < Text1.Text Then
            allografi = allografi & "|" & rs("word_word")
            End If

            End If
               rs.MoveNext
            Loop
            If Not (allografi = "") Then
            Print #fnum, List1.List(i); allografi; vbCrLf;
            rs.Close
       allografi = ""
       End If
Next Close #fnum cn.Close

End Sub
```

A.2 Declinatore automatico

In questa parte si riporta il codice del declinatore automatico progettato per incrementare il database di forme del declinatore.

```
Private Sub Command1_Click() Dim cn As New ADODB.Connection, rs As New
ADODB.Recordset, rs2 As New ADODB.Recordset Dim query As String Dim query2 As
String Dim pippo As String Dim riga As String Dim fnum As Integer Dim
colonna_radice As String

'queste sono tutte le variabili di tutte le tabelle e della tabella "forme" Dim
lemma_id As String Dim forma As String Dim tipo As String Dim kind As String
Dim decl As String Dim conj As String Dim mode As String Dim tense As String
Dim person As String Dim number As String Dim gender As String Dim caso As
String Dim age As String Dim area As String Dim geo As String Dim freq_lemma As
String Dim freq_des As String Dim source As String


"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=F:\Documents and
Settings\Stefano\Desktop\dizionario_autogenerato\nuovo_dizionario.mdb"


'**************************Nomi************************** If N.Value = 1
Then
```

```
'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM N;" rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "nomideclinato.txt" For Output As #fnum
        'Print #fnum, "lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;f

'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF

  'trattamento desinenze comuni

  If Not (Ultime2(rs("decl")) = " 0") Then
   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
               Do Until rs2.EOF

                 colonna_radice = "stem" & rs("stem")
                   If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then

                       lemma_id = Trim(rs2("lemma_id"))
                       forma = rs2(colonna_radice) & rs("des")
                       soundex = GetSoundex(forma)
                       tipo = rs2("type")
                       kind = rs2("kind")
                       decl = rs2("decl")
                       conj = ""
                       mode = ""
                       tense = ""
                       person = ""
                       number = rs("number")
                       gender = rs("gender")
                       caso = rs("case")
                       age = rs2("age")
                       area = rs2("area")
                       geo = rs2("geo")
                       freq_lemma = rs2("freq")
                       freq_des = rs("freq")
                       source = rs2("source")

                       Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;
                   End If
                       rs2.MoveNext
               Loop
```

```
        rs2.Close
Else


    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl LIKE '" & Primo(rs("decl")) & "
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")

                If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                End If

                    rs2.MoveNext
            Loop

        rs2.Close


End If
            rs.MoveNext
        Loop

        rs.Close
    Close #fnum
```

```
End If

'*********************************verbi****************** If V.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM V;" rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "verbideclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sou
'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF

  'trattamento desinenze comuni

  If Not (Ultime2(rs("decl")) = " 0") Then
   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='V' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
               Do Until rs2.EOF

                 colonna_radice = "stem" & rs("stem")
               If Not (rs2(colonna_radice) = "zzz") Then

                       lemma_id = Trim(rs2("lemma_id"))
                       forma = rs2(colonna_radice) & rs("des")
                       soundex = GetSoundex(forma)
                       tipo = rs2("type")
                       kind = rs2("kind")
                       decl = rs2("decl")
                       conj = rs("conj")
                       mode = rs("mode")
                       tense = rs("tense")
                       person = rs("person")
                       number = rs("number")
                       gender = ""
                       caso = ""
                       age = rs2("age")
                       area = rs2("area")
                       geo = rs2("geo")
                       freq_lemma = rs2("freq")
                       freq_des = rs("freq")
                       source = rs2("source")

                       Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;
```

```
            End If
                    rs2.MoveNext
            Loop

        rs2.Close
Else

If (rs("decl") = "0 0") Then
    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='V';"
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = rs("conj")
                    mode = rs("mode")
                    tense = rs("tense")
                    person = rs("person")
                    number = rs("number")
                    gender = ""
                    caso = ""
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

              End If
                    rs2.MoveNext
            Loop

        rs2.Close


End If

    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='V' AND decl LIKE '" & Primo(rs("decl")) & "%';"
```

```
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

        'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = rs("conj")
                    mode = rs("mode")
                    tense = rs("tense")
                    person = rs("person")
                    number = rs("number")
                    gender = ""
                    caso = ""
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

              End If
                    rs2.MoveNext
            Loop

          rs2.Close


    End If
            rs.MoveNext
        Loop

        rs.Close
      Close #fnum

End If '*********************************aggettivi****************** If
ADJ.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM ADJ;" rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText

'prepara il file di output
```

```
      fnum = FreeFile()
      Open "aggettdeclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sourc
'scorre il recordset desinenza per desinenza rs.MoveFirst
           Do Until rs.EOF
  'trattamento desinenze comuni

  If Not (Ultime2(rs("decl")) = " 0") Then
   'crea secondo recordset con le radici per ciascuna desinenza
               query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
               rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
             Do Until rs2.EOF

                colonna_radice = "stem" & rs("stem")
                      If Not (rs2(colonna_radice) = "zzz") Then

                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = rs("kind")
                      decl = rs2("decl")
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = rs("number")
                      gender = rs("gender")
                      caso = rs("case")
                      age = rs2("age")
                      area = rs2("area")
                      geo = rs2("geo")
                      freq_lemma = rs2("freq")
                      freq_des = rs("freq")
                      source = rs2("source")

                      Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                      End If
                      rs2.MoveNext
             Loop

          rs2.Close

    Else

    If (rs("decl") = "0 0") Then
```

```
'crea secondo recordset con le radici per ciascuna desinenza
        query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND (kind = 'X');"
        rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

'attacca le desinenze alla radice giusta e stampa un file di testo
        Do Until rs2.EOF

          colonna_radice = "stem" & rs("stem")

          If Not (rs2(colonna_radice) = "zzz") Then

                lemma_id = Trim(rs2("lemma_id"))
                forma = rs2(colonna_radice) & rs("des")
                soundex = GetSoundex(forma)
                tipo = rs2("type")
                kind = rs("kind")
                decl = rs2("decl")
                conj = ""
                mode = ""
                tense = ""
                person = ""
                number = rs("number")
                gender = rs("gender")
                caso = rs("case")
                age = rs2("age")
                area = rs2("area")
                geo = rs2("geo")
                freq_lemma = rs2("freq")
                freq_des = rs("freq")
                source = rs2("source")

                Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

          End If
                rs2.MoveNext
        Loop

    rs2.Close

    'crea secondo recordset con le radici per ciascuna desinenza questo per declinare i comparativi e superla
        query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND (kind ='" & rs("kind") & "');"
        rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

'attacca le desinenze alla radice giusta e stampa un file di testo
        Do Until rs2.EOF

          colonna_radice = "stem1"

          If Not (rs2(colonna_radice) = "zzz") Then

                lemma_id = Trim(rs2("lemma_id"))
```

```
                forma = rs2(colonna_radice) & rs("des")
                soundex = GetSoundex(forma)
                tipo = rs2("type")
                kind = rs("kind")
                decl = rs2("decl")
                conj = ""
                mode = ""
                tense = ""
                person = ""
                number = rs("number")
                gender = rs("gender")
                caso = rs("case")
                age = rs2("age")
                area = rs2("area")
                geo = rs2("geo")
                freq_lemma = rs2("freq")
                freq_des = rs("freq")
                source = rs2("source")

                Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

          End If
                rs2.MoveNext
        Loop

    rs2.Close

    'questo per declinare i comparativi e superlativi di tutti i normali aggettivi
        query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND (kind ='POS');"
        rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

'attacca le desinenze alla radice giusta e stampa un file di testo
        Do Until rs2.EOF

          colonna_radice = "stem2"

          If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "") Then
               If rs("kind") = "SUPER" Then

                lemma_id = Trim(rs2("lemma_id"))
                forma = rs2(colonna_radice) & "issi" & rs("des")
                soundex = GetSoundex(forma)
                tipo = rs2("type")
                kind = rs("kind")
                decl = rs2("decl")
                conj = ""
                mode = ""
                tense = ""
                person = ""
                number = rs("number")
                gender = rs("gender")
```

```
                caso = rs("case")
                age = rs2("age")
                area = rs2("area")
                geo = rs2("geo")
                freq_lemma = rs2("freq")
                freq_des = rs("freq")
                source = rs2("source")

                Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

               End If
                If rs("kind") = "COMP" Then

                lemma_id = Trim(rs2("lemma_id"))
                forma = rs2(colonna_radice) & "i" & rs("des")
                soundex = GetSoundex(forma)
                tipo = rs2("type")
                kind = rs("kind")
                decl = rs2("decl")
                conj = ""
                mode = ""
                tense = ""
                person = ""
                number = rs("number")
                gender = rs("gender")
                caso = rs("case")
                age = rs2("age")
                area = rs2("area")
                geo = rs2("geo")
                freq_lemma = rs2("freq")
                freq_des = rs("freq")
                source = rs2("source")

                Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

               End If
          End If
                rs2.MoveNext
        Loop

    rs2.Close

 Else

    'crea secondo recordset con le radici per ciascuna desinenza
        query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl LIKE '" & Primo(rs("decl")) &
        rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

'attacca le desinenze alla radice giusta e stampa un file di testo
        Do Until rs2.EOF
```

```
                colonna_radice = "stem" & rs("stem")
                      If Not (rs2(colonna_radice) = "zzz") Then


                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = rs("kind")
                      decl = rs2("decl")
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = rs("number")
                      gender = rs("gender")
                      caso = rs("case")
                      age = rs2("age")
                      area = rs2("area")
                      geo = rs2("geo")
                      freq_lemma = rs2("freq")
                      freq_des = rs("freq")
                      source = rs2("source")

                      Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                      End If
                      rs2.MoveNext
                Loop

            rs2.Close
      End If
    End If
                rs.MoveNext
            Loop

            rs.Close
         Close #fnum
End If '*********************************pronomi****************** If
PRON.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM PRON;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "pronomideclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sourc
'scorre il recordset desinenza per desinenza rs.MoveFirst
```

```
          Do Until rs.EOF
'trattamento desinenze comuni

If Not (Ultime2(rs("decl")) = " 0") Then
 'crea secondo recordset con le radici per ciascuna desinenza
               query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
               rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


 'attacca le desinenze alla radice giusta e stampa un file di testo
             Do Until rs2.EOF

               colonna_radice = "stem" & rs("stem")
             If Not (rs2(colonna_radice) = "zzz") Then
                 If (rs2("decl") = "4 2") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des") & "dem"
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

                  Else

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
```

```
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                End If
            End If
                    rs2.MoveNext
            Loop

        rs2.Close
Else

    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl LIKE '" & Primo(rs("decl")) & "
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz") Then
                   If (rs2("decl") = "4 2") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des") & "dem"
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs2("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";
```

```
                    Else

                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = rs2("kind")
                      decl = rs2("decl")
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = rs("number")
                      gender = rs("gender")
                      caso = rs("case")
                      age = rs2("age")
                      area = rs2("area")
                      geo = rs2("geo")
                      freq_lemma = rs2("freq")
                      freq_des = rs("freq")
                      source = rs2("source")

                      Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

                  End If
                End If
                      rs2.MoveNext
              Loop

          rs2.Close


    End If
              rs.MoveNext
          Loop

          rs.Close
       Close #fnum

End If '*********************************avverbi****************** If ADV.Value
= 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM ADV;" rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "avverbideclinato.txt" For Output As #fnum
        'Print #fnum, "lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;f
'scorre il recordset desinenza per desinenza rs.MoveFirst
```

```
          Do Until rs.EOF
  'trattamento desinenze comuni

   'crea secondo recordset con le radici per ciascuna desinenza
               query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
               rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
             Do Until rs2.EOF

Select Case rs("decl") Case "X": Select Case rs("stem") Case 1:
                colonna_radice = "stem" & rs("stem")

                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = ""
                      decl = "POS"
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = ""
                      gender = ""
                      caso = ""
                      age = rs2("age")
                      area = rs2("area")
                      geo = rs2("geo")
                      freq_lemma = rs2("freq")
                      freq_des = rs("freq")
                      source = rs2("source")
Case 2:
                colonna_radice = "stem" & rs("stem")

                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = ""
                      decl = "COMP"
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = ""
                      gender = ""
                      caso = ""
                      age = rs2("age")
                      area = rs2("area")
```

```
                        geo = rs2("geo")
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")

Case 3:
                  colonna_radice = "stem" & rs("stem")

                        lemma_id = Trim(rs2("lemma_id"))
                        forma = rs2(colonna_radice) & rs("des")
                        soundex = GetSoundex(forma)
                        tipo = rs2("type")
                        kind = ""
                        decl = "SUPER"
                        conj = ""
                        mode = ""
                        tense = ""
                        person = ""
                        number = ""
                        gender = ""
                        caso = ""
                        age = rs2("age")
                        area = rs2("area")
                        geo = rs2("geo")
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")
End Select

Case Else:
                        colonna_radice = "stem" & rs("stem")

                        lemma_id = Trim(rs2("lemma_id"))
                        forma = rs2(colonna_radice) & rs("des")
                        soundex = GetSoundex(forma)
                        tipo = rs2("type")
                        kind = ""
                        decl = rs2("decl")
                        conj = ""
                        mode = ""
                        tense = ""
                        person = ""
                        number = ""
                        gender = ""
                        caso = ""
                        age = rs2("age")
                        area = rs2("area")
                        geo = rs2("geo")
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")
```

```
End Select
                        If rs2(colonna_radice) <> "zzz" Then
                        Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";
                        End If
                        rs2.MoveNext
                Loop

            rs2.Close

                rs.MoveNext
            Loop

            rs.Close
         Close #fnum


End If

'*******************************preposizioni****************** If PREP.Value
= 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM PREP;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "preposizionideclinato.txt" For Output As #fnum
       'Print #fnum, "lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq
'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni

   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
                Do Until rs2.EOF

                  colonna_radice = "stem" & rs("stem")

                        lemma_id = Trim(rs2("lemma_id"))
                        forma = rs2(colonna_radice) & rs("des")
                        soundex = GetSoundex(forma)
                        tipo = rs2("type")
                        kind = ""
                        decl = rs2("decl")
                        conj = ""
                        mode = ""
```

```
                        tense = ""
                        person = ""
                        number = ""
                        gender = ""
                        caso = ""
                        age = rs2("age")
                        area = rs2("area")
                        geo = rs2("geo")
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")

                        Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

                        rs2.MoveNext
                    Loop

                rs2.Close

                    rs.MoveNext
                Loop

                rs.Close
            Close #fnum


End If

'*********************************participi****************** If VPAR.Value = 1
Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM VPAR;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "vpardeclinato.txt" For Output As #fnum
       'Print #fnum, "lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;fr
'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni

  If Not (Ultime2(rs("decl")) = " 0") Then
   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='V' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
                Do Until rs2.EOF
```

```
              colonna_radice = "stem" & rs("stem")
            If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
                    conj = rs("conj")
                    mode = "PART"
                    tense = rs("tense")
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

            End If
                    rs2.MoveNext
            Loop

        rs2.Close
Else

If (rs("decl") = "0 0") Then
    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='V';"
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then


                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
```

```
                    conj = rs("conj")
                    mode = "PART"
                    tense = rs("tense")
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

              End If
                    rs2.MoveNext
            Loop

        rs2.Close


End If

    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='V' AND decl LIKE '" & Primo(rs("decl")) & "%';"
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
                    conj = rs("conj")
                    mode = "PART"
                    tense = rs("tense")
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
```

```
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")

                        Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                       End If
                        rs2.MoveNext
                Loop

            rs2.Close


    End If
                rs.MoveNext
            Loop

            rs.Close
         Close #fnum

End If '*********************************supini****************** If
SUPINE.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM SUPINE;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText

'prepara il file di output
       fnum = FreeFile()
       Open "supinideclinato.txt" For Output As #fnum
        'Print #fnum, "lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;fre
'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni

   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='V';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
                Do Until rs2.EOF

                 colonna_radice = "stem" & rs("stem")
                 If Not (rs2(colonna_radice) = "zzz" Or rs2(colonna_radice) = "zzzz") Then

                       lemma_id = Trim(rs2("lemma_id"))
                       forma = rs2(colonna_radice) & rs("des")
                       soundex = GetSoundex(forma)
                       tipo = rs2("type")
                       kind = rs2("kind")
```

```
                    decl = rs2("decl")
                    conj = ""
                    mode = "SUPINE"
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

                End If
                    rs2.MoveNext
                Loop

            rs2.Close

                rs.MoveNext
            Loop

            rs.Close
        Close #fnum


End If '*********************************cong****************** If CONG.Value =
1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM CONJ;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText 'prepara il file di output
       fnum = FreeFile()
       Open "congiunzdeclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sou
'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni

   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
                Do Until rs2.EOF
```

```
                colonna_radice = "stem" & rs("stem")

                      lemma_id = Trim(rs2("lemma_id"))
                      forma = rs2(colonna_radice) & rs("des")
                      soundex = GetSoundex(forma)
                      tipo = rs2("type")
                      kind = ""
                      decl = rs2("decl")
                      conj = ""
                      mode = ""
                      tense = ""
                      person = ""
                      number = ""
                      gender = ""
                      caso = ""
                      age = rs2("age")
                      area = rs2("area")
                      geo = rs2("geo")
                      freq_lemma = rs2("freq")
                      freq_des = rs("freq")
                      source = rs2("source")

                      Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                      rs2.MoveNext
                Loop

            rs2.Close

                rs.MoveNext
            Loop

            rs.Close
        Close #fnum

End If '*********************************numerali****************** If
NUM.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM NUM;" rs.Open query, cn, adOpenStatic, adLockReadOnly, adCmdText
'prepara il file di output
       fnum = FreeFile()
       Open "numdeclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sourc

'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni
```

```
If Not (Ultime2(rs("decl")) = " 0") Then
 'crea secondo recordset con le radici per ciascuna desinenza
              query2 = "SELECT * from radici WHERE type='NUM' AND decl='" & rs("decl") & "';"
              rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


 'attacca le desinenze alla radice giusta e stampa un file di testo
             Do Until rs2.EOF

               colonna_radice = "stem" & rs("stem")
             If Not (rs2(colonna_radice) = "zzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

             End If
                    rs2.MoveNext
             Loop

        rs2.Close
 Else

 If (rs("decl") = "0 0") Then
     'crea secondo recordset con le radici per ciascuna desinenza
             query2 = "SELECT * from radici WHERE type='NUM';"
             rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

     'attacca le desinenze alla radice giusta e stampa un file di testo
             Do Until rs2.EOF

               colonna_radice = "stem" & rs("stem")
               If Not (rs2(colonna_radice) = "zzz") Then
```

```
                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

              End If
                    rs2.MoveNext
            Loop

        rs2.Close


End If

    'crea secondo recordset con le radici per ciascuna desinenza
            query2 = "SELECT * from radici WHERE type='NUM' AND decl LIKE '" & Primo(rs("decl")) & "%';"
            rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText

    'attacca le desinenze alla radice giusta e stampa un file di testo
            Do Until rs2.EOF

              colonna_radice = "stem" & rs("stem")
              If Not (rs2(colonna_radice) = "zzz") Then

                    lemma_id = Trim(rs2("lemma_id"))
                    forma = rs2(colonna_radice) & rs("des")
                    soundex = GetSoundex(forma)
                    tipo = rs2("type")
                    kind = rs("kind")
                    decl = rs2("decl")
                    conj = ""
                    mode = ""
                    tense = ""
```

```
                    person = ""
                    number = rs("number")
                    gender = rs("gender")
                    caso = rs("case")
                    age = rs2("age")
                    area = rs2("area")
                    geo = rs2("geo")
                    freq_lemma = rs2("freq")
                    freq_des = rs("freq")
                    source = rs2("source")

                    Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj;

                    End If
                    rs2.MoveNext
                Loop

            rs2.Close


    End If
                rs.MoveNext
            Loop

            rs.Close
        Close #fnum

End If '*********************************interiezioni****************** If
INTERJ.Value = 1 Then

'tira fuori tutte le desinenze di una determinata parte del discorso query =
"SELECT * FROM INTERJ;" rs.Open query, cn, adOpenStatic, adLockReadOnly,
adCmdText 'prepara il file di output
       fnum = FreeFile()
       Open "interiezionideclinato.txt" For Output As #fnum
'Print #fnum,
"lemma_id;forma;soundex;type;kind;decl;conj;mode;tense;person;number;gender;case;age;area;geo;freq_lemma;freq_des;sou

'scorre il recordset desinenza per desinenza rs.MoveFirst
            Do Until rs.EOF
  'trattamento desinenze comuni

   'crea secondo recordset con le radici per ciascuna desinenza
                query2 = "SELECT * from radici WHERE type='" & rs("type") & "' AND decl='" & rs("decl") & "';"
                rs2.Open query2, cn, adOpenStatic, adLockReadOnly, adCmdText


   'attacca le desinenze alla radice giusta e stampa un file di testo
                Do Until rs2.EOF

                 colonna_radice = "stem" & rs("stem")
```

```
                        lemma_id = Trim(rs2("lemma_id"))
                        forma = rs2(colonna_radice) & rs("des")
                        soundex = GetSoundex(forma)
                        tipo = rs2("type")
                        kind = ""
                        decl = rs2("decl")
                        conj = ""
                        mode = ""
                        tense = ""
                        person = ""
                        number = ""
                        gender = ""
                        caso = ""
                        age = rs2("age")
                        area = rs2("area")
                        geo = rs2("geo")
                        freq_lemma = rs2("freq")
                        freq_des = rs("freq")
                        source = rs2("source")

                        Print #fnum, lemma_id; ";"; forma; ";"; soundex; ";"; tipo; ";"; kind; ";"; decl; ";"; conj; ";

                        rs2.MoveNext
                    Loop

                rs2.Close

                    rs.MoveNext
                Loop

                rs.Close
            Close #fnum


    End If '*********************************pack****************** If PACK.Value =
    1 Then

    End If

    cn.Close

    End Sub
```

Appendice B

Seconda Appendice

Di seguito vengono presentate le formalizzazioni delle categorie grammaticali e dei paradigmi morfologici usati dalla procedura di declinazione/lemmatizzazione.

```
--  SENZA PARTI VARIANTI
ADV    X 1 0         X A
ADV    X 2 0         X A
ADV    X 3 0         X A
ADV    POS 1 0       X A
ADV    COMP 1 0      X A
ADV    SUPER 1 0     X A
PREP   ACC 1 0       X A
PREP   ABL 1 0       X A
CONJ   1 0           X A
INTERJ 1 0           X A
--                        PRIMA DECLINAZIONE
--  N 1 1
--  Es: aqua aquae  =>  aqu aqu
N     1 1 NOM S C  1 1 a         X A
N     1 1 NOM S M  1 2 as        B D
N     1 1 VOC S C  1 1 a         X A
N     1 1 GEN S C  2 2 ae        X A
N     1 1 GEN S C  2 2 ai        B C
N     1 1 LOC S C  2 2 ae        X A
N     1 0 DAT S C  2 2 ae        X A
N     1 0 DAT S C  2 2 ai        B I
N     1 1 ABL S C  2 1 a         X A
N     1 1 ABL S C  2 2 ad        B D
N     1 1 ACC S C  2 2 am        X A
N     1 0 NOM P C  2 2 ae        X A
N     1 0 VOC P C  2 2 ae        X A
N     1 0 GEN P C  2 4 arum      X A
N     1 0 GEN P C  2 2 um        X C
N     1 0 LOC P C  2 2 is        X A
N     1 0 DAT P C  2 2 is        X A
```

```
N     1 0 DAT P C  2 4 abus      D B
N     1 0 ABL P C  2 2 is        X A
N     1 0 ABL P C  2 4 abus      D B
N     1 0 ACC P C  2 2 as        X A
--  NOMI GRECI
--  Ex:  epitome, epitomes  =>   epitom epitom
--  Ex:  musice, musices    =>   music  music
N     1 6 NOM S C  1 1 e         X A
N     1 6 VOC S C  1 1 e         X A
N     1 6 GEN S C  2 2 es        X A
N     1 6 LOC S C  2 2 es        X D
N     1 6 ABL S C  2 1 e         X A
N     1 6 ACC S C  2 2 en        X A
--  Ex:  cometes, cometae   =>   comet comet
N     1 7 NOM S C  1 2 es        X A
N     1 7 VOC S C  1 1 e         X A
N     1 7 VOC S C  1 1 a         X B
N     1 7 GEN S C  2 2 ae        X A
N     1 7 GEN S C  2 2 ai        B B
N     1 7 LOC S C  2 2 ae        X A
N     1 7 ABL S C  2 1 e         X A
N     1 7 ABL S C  2 1 a         X B
N     1 7 ACC S C  2 2 en        X A
N     1 7 ACC S C  2 2 am        X B
--  Ex:  Archias, Archiae   =>   Archi Archi
--  Ex:  Aeneas, Aeneae     =>   Aene  Aene
N     1 8 NOM S M  1 2 as        X A
N     1 8 NOM S F  1 1 a         X A
N     1 8 VOC S C  1 1 a         X A
N     1 8 GEN S C  2 2 ae        X A
N     1 8 GEN S C  2 2 ai        B B
N     1 8 LOC S C  2 2 ae        X A
N     1 8 ACC S C  2 2 an        X A
N     1 8 ACC S C  2 2 am        X B
N     1 8 ABL S C  2 1 a         X A
-- Tiridates -is e -ae mia aggiunta
N     1 9 NOM S C  1 2 es        X A
N     1 9 VOC S C  1 2 es        X A
N     1 9 GEN S C  2 2 ae        X A
N     1 9 GEN S C  2 2 is        B B
N     1 9 DAT S C  2 1 i         B B
N     1 9 ACC S C  2 2 em        X A
N     1 9 ACC S C  2 2 am        X B
N     1 9 ABL S C  2 1 a         X A
N     1 9 ABL S C  2 1 e         X A
--                        SECONDA DECLINAZIONE
--  Ex: amicus amici  =>  amic amic
N     2 1 NOM S X  1 2 us        X A
N     2 1 NOM S C  1 2 os        A B
N     2 1 VOC S X  1 1 e         X A
N     2 1 GEN S X  2 1 i         X A
```

```
N     2 0 LOC S X  2 1 i         X A
N     2 0 DAT S X  2 1 o         X A
N     2 0 ABL S X  2 1 o         X A
N     2 0 ACC S C  2 2 um        X A
N     2 1 ACC S N  2 2 us        X A
N     2 0 ACC S C  2 2 om        A B
N     2 0 NOM P C  2 1 i         X A
N     2 0 NOM P N  2 1 e         X A
N     2 0 VOC P C  2 1 i         X A
N     2 0 GEN P X  2 4 orum      X A
N     2 0 GEN P X  2 2 um        X C
N     2 0 LOC P X  2 2 is        X A
N     2 0 DAT P X  2 2 is        X A
N     2 0 ABL P X  2 2 is        X A
N     2 0 ACC P C  2 2 os        X A
N     2 0 ACC P N  2 1 e         X A
--  Ex: verbum verbi  =>  verb verb
N     2 2 NOM S N  1 2 um        X A
N     2 2 NOM S N  1 2 om        A B
N     2 2 VOC S N  1 2 um        X A
N     2 2 GEN S N  2 1 i         X A
N     2 2 ACC S N  2 2 um        X A
N     2 0 NOM P N  2 1 a         X A
N     2 0 VOC P N  2 1 a         X A
N     2 0 ACC P N  2 1 a         X A
--  Ex; puer pueri  =>  puer puer
--  Ex: ager agri   =>  ager agr
N     2 3 NOM S C  1 0           X A
N     2 3 VOC S C  1 0           X A
N     2 3 GEN S C  2 1 i         X A
--  Ex: radius rad(i)i  => radi radi        M
--  Ex: atrium atr(i)i  =>  atri atri       N
N     2 4 NOM S C  1 2 us        X A
N     2 4 VOC S C  1 1 e         X A
N     2 4 NOM S N  1 2 um        X A
N     2 4 VOC S N  1 2 um        X A
N     2 4 GEN S X  2 1 i         D B
N     2 4 GEN S X  2 0           B A
N     2 4 ACC S X  2 2 um        X A
--  Ex: filius fili  =>  fili  fili
--  Ex: Lucius Lucii  =>  Luci  Luci
N     2 5 NOM S M  1 2 us        X A
N     2 5 VOC S M  1 0           X A
N     2 5 GEN S M  2 0           X A
N     2 5 GEN S M  2 1 i         E A
N     2 5 ACC S M  2 2 um        X A
--  Ex:  barbitos barbiti   =>   barbit barbit
N     2 6 NOM S X  1 2 os        X A
N     2 6 VOC S X  1 1 e         X A
N     2 6 GEN S X  2 1 i         X A
N     2 6 ACC S C  2 2 on        X A
```

```
N     2 6 ACC S N  2 2 os        X A
N     2 6 NOM P X  1 2 oe        B C
--  Ex:  Androgeos  Androgeo  =>  Andregeos  Andrege
N     2 7 NOM S X  1 0           X A
N     2 7 VOC S X  1 0           X A
N     2 7 GEN S X  2 1 o         X A
N     2 7 GEN S X  2 1 i         X B
N     2 7 ACC S X  2 2 on        X A
N     2 7 ACC S X  2 1 o         X B
--  Ex:  Ilion Ilii    =>  Ili  Ili
N     2 8 NOM S N  1 2 on        X A
N     2 8 VOC S N  2 2 on        X A
N     2 8 GEN S N  2 1 i         X A
N     2 8 ACC S N  2 2 on        X A
--  Ex:  Panthus, Panthi => Panth Panth
--  Ex:  Teucrus
N     2 9 NOM S C  1 2 us        X A
N     2 9 VOC S C  2 1 u         X A
N     2 9 GEN S C  2 1 i         X A
N     2 9 ACC S C  2 2 un        X A
--- altro nome greco per Perseus -ei -eos
N     2 10 NOM S C  1 2 us        X A
N     2 10 VOC S C  2 1 u         X A
N     2 10 GEN S C  2 2 os        X A
N     2 10 GEN S C  2 1 i         X A
N     2 10 ACC S C  2 2 a         X A
--  Ex: miles militis  =>  miles milit
--  Ex: lex legis  =>  lex leg
--  Ex: frater fratris  =>  frater fratr
--  Ex: soror sororis  =>  soror soror
--  Ex: pulcritudo pulcritudinis  =>  plucritudo pulcritudin
--  Ex: legio legionis  =>  legio legion
--  Ex: varietas varietatis  =>  varietas varietat
--  Ex: radix radicis  =>  radix  radic
N     3 0 NOM S X  1 0           X A
N     3 0 VOC S X  1 0           X A
N     3 0 GEN S X  2 2 is        X A
N     3 0 LOC S X  2 1 i         B A
N     3 0 LOC S X  2 1 e         X A
N     3 0 DAT S X  2 1 i         X A
N     3 0 DAT S X  2 1 e         B B
N     3 0 ABL S C  2 1 e         X A
N     3 1 ACC S C  2 2 em        X A
N     3 0 NOM P C  2 2 es        X A
N     3 0 NOM P C  2 2 is        A B
N     3 0 VOC P C  2 2 es        X A
N     3 1 GEN P C  2 2 um        X A
N     3 1 GEN P C  2 3 ium       X B
N     3 0 LOC P X  2 4 ibus      X A
N     3 0 DAT P X  2 4 ibus      X A
N     3 0 ABL P X  2 4 ibus      X A
```

```
N     3 0 ACC P C  2 2 es        X A
--  Ex: nomen nomenis  =>  nomen nomen
--  Ex: iter itineris =>  iter itiner
--  Ex: tempus temporis  =>  tempus  tempor
N     3 2 ABL S N  2 1 e         X A
N     3 2 ACC S N  1 0           X A
N     3 2 NOM P N  2 1 a         X A
N     3 2 VOC P N  2 1 a         X A
N     3 2 GEN P N  2 2 um        X A
N     3 2 GEN P N  2 3 ium       X B
N     3 2 ACC P N  2 1 a         X A
--  Ex: hostis hostis  =>  hostis host
--  Ex: finis finis  =>  finis fin
--  Ex: urbs urbis  =>  urbs urb
--  Ex: mons montis  =>  mons mont
N     3 3 ABL S C  2 1 i         X B
N     3 3 ACC S C  2 2 em        X A
N     3 3 ACC S C  2 2 im        X A
N     3 3 NOM P C  2 2 is        B C
N     3 3 GEN P C  2 3 ium       X A
N     3 3 GEN P C  2 2 um        X B
N     3 3 ACC P C  2 2 is        B A
N     3 3 ACC P C  2 3 eis       A C
--  Ex: mare maris  =>  mare mar
--  Ex: animal animalis  =>  animal animal
--  Ex: exemplar exemplaris  =>  exemplar exemplar
N     3 4 ABL S N  2 1 i         X A
N     3 4 ACC S N  1 0           X A
N     3 4 NOM P N  2 2 ia        X A
N     3 4 VOC P N  2 2 ia        X A
N     3 4 GEN P N  2 3 ium       X A
N     3 4 GEN P N  2 2 um        X B
N     3 4 ACC P N  2 2 ia        X A
--  Ex:  aer aeris  =>  aer aer
N     3 6 GEN S X  2 2 is        X A
N     3 6 ACC S X  2 1 a         X A
N     3 6 ACC S X  2 2 em        X B
N     3 6 GEN P X  2 2 um        X A
N     3 6 ACC P X  2 2 as        X A
--  Ex:  lampas lampados  =>  lampas  lampad
--  Atlantis, Atlantidos  =>  Atlantis Atlantid
N     3 7 GEN S X  2 2 os        X A
N     3 7 ACC S X  2 1 a         X A
N     3 7 ACC S X  2 2 em        E B
N     3 7 GEN P X  2 2 um        X A
N     3 7 ACC P X  2 2 as        X A
--  Ex:  tigris tigris/tigridis  =>  tigris tigr/tigrid
--  Ex:  praxis praxios  =>  prax praxi
--  Ex:  haeresis haereseos  =>  haeres haerese
--  Ex:  pater patros  =>  pater patr
N     3 9 NOM S X  1 2 is        X A
```

```
N     3 9 GEN S X  2 2 os        X A
N     3 9 ABL S X  2 1 i         X A
N     3 9 ACC S X  1 2 in        X A
N     3 9 ACC S X  2 2 in        X A
N     3 9 ACC S X  1 2 on        X B
N     3 9 ACC S X  2 2 on        X B
N     3 9 ACC S X  1 1 a         X B
N     3 9 ACC S X  2 1 a         X B
N     3 9 GEN P X  2 3 ium       X A
N     3 9 GEN P X  2 2 on        X A
N     3 9 ACC P X  2 2 is        X A
N     3 9 ACC P X  2 2 as        X B
--  Ex: passus passus  =>  pass pass
--  Ex: manus manus  =>  man man
N     4 1 NOM S C  1 2 us        X A
N     4 1 VOC S C  1 2 us        X A
N     4 1 GEN S X  2 2 us        X A
N     4 1 GEN S X  2 2 os        A C
N     4 1 GEN S X  2 1 i         A B
N     4 1 DAT S C  2 2 ui        X A
N     4 1 DAT S C  2 1 u         D C
N     4 0 ABL S X  2 1 u         X A
N     4 1 ACC S C  2 2 um        X A
N     4 1 NOM P C  2 2 us        X A
N     4 1 NOM P C  2 3 uus       D C
N     4 1 VOC P C  2 2 us        X A
N     4 1 VOC P C  2 3 uus       D C
N     4 0 GEN P X  2 3 uum       X A
N     4 0 GEN P X  2 2 um        C C
N     4 0 DAT P X  2 4 ibus      X A
N     4 0 DAT P X  2 4 ubus      B C
N     4 0 ABL P X  2 4 ibus      X A
N     4 0 ABL P X  2 4 ubus      B C
N     4 1 ACC P C  2 2 us        X A
N     4 1 ACC P C  2 3 uus       D C
--  Ex: genu genus  =>  gen gen
--  Ex: cornu cornus  =>  corn corn
N     4 2 NOM S N  1 1 u         X A
N     4 2 VOC S N  1 1 u         X A
N     4 2 GEN S X  2 2 us        X A
N     4 2 DAT S N  2 1 u         X A
N     4 2 ACC S N  2 1 u         X A
N     4 2 NOM P N  2 2 ua        X A
N     4 2 VOC P N  2 2 ua        X A
N     4 2 ACC P N  2 2 ua        X A
--  Ex: Jesus Jesu    =>  Ies Ies  (Jes Jes)
N     4 3 NOM S C  1 2 us        X A
N     4 3 VOC S C  1 1 u         X A
N     4 3 GEN S C  2 1 u         X A
N     4 3 DAT S C  2 1 u         X A
N     4 3 ACC S C  2 2 em        X A
```

```
N     4 3 ACC S C  2 2 um        X A
-- questa la metto per i nomi della quarta greci tipo sappho
N     4 4 NOM S C  1 0           X A
N     4 4 VOC S C  1 0           X A
N     4 4 GEN S C  2 2 us        X A
N     4 4 DAT S C  1 0           X A
N     4 4 ACC S C  1 0           X A
N     4 4 ABL S C  1 0           X A
--  Ex: dies diei  =>  di di
--  Ex: res rei  =>  r r
N     5 1 NOM S C  1 2 es        X A
N     5 1 VOC S C  1 2 es        X A
N     5 1 GEN S C  2 2 ei        X A
N     5 1 GEN S C  2 1 e         B B
N     5 1 DAT S C  2 2 ei        X A
N     5 1 DAT S C  2 1 e         B C
N     5 1 DAT S C  2 1 i         B F
N     5 1 ABL S C  2 1 e         X A
N     5 1 ACC S C  2 2 em        X A
N     5 1 NOM P C  2 2 es        X A
N     5 1 VOC P C  2 2 es        X A
N     5 1 GEN P C  2 4 erum      X A
N     5 1 DAT P C  2 4 ebus      X A
N     5 1 ABL P C  2 4 ebus      X A
N     5 1 ABL P C  2 3 eis       X B
N     5 1 ACC P C  2 2 es        X A
--  abbreviazioni, indeclinabili
N     9 8 X X X  1 0     X A
--  nomi non declinati: fas
N     9 9 X X X  1 0     X A
--  AGGETTIVI
--  Ex: malus mala malum  => mal mal pei pessi
--  Ex: altus alta altum  =>  alt alt alti altissi
ADJ   1 1 NOM S M POS   1 2 us    X A
ADJ   1 1 GEN S M POS   2 1 i     X A
ADJ   1 1 DAT S M POS   2 1 o     X A
ADJ   1 0 ACC S M POS   2 2 um    X A
ADJ   1 0 ABL S M POS   2 1 o     X A
ADJ   1 1 VOC S M POS   1 1 e     X A
ADJ   1 0 NOM P M POS   2 1 i     X A
ADJ   1 0 GEN P M POS   2 4 orum  X A
ADJ   1 0 DAT P X POS   2 2 is    X A
ADJ   1 0 ACC P M POS   2 2 os    X A
ADJ   1 0 ABL P X POS   2 2 is    X A
ADJ   1 0 VOC P M POS   2 1 i     X A
ADJ   1 0 NOM S F POS   2 1 a     X A
ADJ   1 1 GEN S F POS   2 2 ae    X A
ADJ   1 1 DAT S F POS   2 2 ae    X A
ADJ   1 0 ACC S F POS   2 2 am    X A
ADJ   1 0 ABL S F POS   2 1 a     X A
ADJ   1 0 VOC S F POS   2 1 a     X A
```

```
ADJ   1 0 NOM P F POS   2 2 ae    X A
ADJ   1 0 GEN P F POS   2 4 arum  X A
ADJ   1 0 ACC P F POS   2 2 as    X A
ADJ   1 0 VOC P F POS   2 2 ae    X A
ADJ   1 1 NOM S N POS   2 2 um    X A
ADJ   1 1 GEN S N POS   2 1 i     X A
ADJ   1 1 DAT S N POS   2 1 o     X A
ADJ   1 1 ACC S N POS   2 2 um    X A
ADJ   1 0 ABL S N POS   2 1 o     X A
ADJ   1 1 VOC S N POS   2 2 um    X A
ADJ   1 0 NOM P N POS   2 1 a     X A
ADJ   1 0 GEN P N POS   2 4 orum  X A
ADJ   1 0 ACC P N POS   2 1 a     X A
ADJ   1 0 VOC P N POS   2 1 a     X A
ADJ   0 0 NOM S C COMP  3 2 or    X A
ADJ   0 0 GEN S C COMP  3 4 oris  X A
ADJ   0 0 DAT S X COMP  3 3 ori   X A
ADJ   0 0 ACC S C COMP  3 4 orem  X A
ADJ   0 0 ABL S X COMP  3 3 ore   X A
ADJ   0 0 ABL S X COMP  3 3 ori   X A
ADJ   0 0 VOC S C COMP  3 2 or    X A
ADJ   0 0 NOM P C COMP  3 4 ores  X A
ADJ   0 0 GEN P X COMP  3 4 orum  X A
ADJ   0 0 DAT P X COMP  3 6 oribus    X A
ADJ   0 0 ACC P C COMP  3 4 ores      X A
ADJ   0 0 ABL P X COMP  3 6 oribus    X A
ADJ   0 0 VOC P C COMP  3 4 ores      X A
ADJ   0 0 NOM S N COMP  3 2 us        X A
ADJ   0 0 GEN S N COMP  3 4 oris      X A
ADJ   0 0 ACC S N COMP  3 2 us        X A
ADJ   0 0 VOC S N COMP  3 2 us        X A
ADJ   0 0 NOM P N COMP  3 3 ora       X A
ADJ   0 0 ACC P N COMP  3 3 ora       X A
ADJ   0 0 VOC P N COMP  3 3 ora       X A
ADJ   0 0 NOM S M SUPER 4 3 mus       X A
ADJ   0 0 GEN S M SUPER 4 2 mi        X A
ADJ   0 0 DAT S M SUPER 4 2 mo        X A
ADJ   0 0 ACC S M SUPER 4 3 mum       X A
ADJ   0 0 ABL S M SUPER 4 2 mo        X A
ADJ   0 0 VOC S M SUPER 4 2 me        X A
ADJ   0 0 NOM P M SUPER 4 2 mi        X A
ADJ   0 0 GEN P M SUPER 4 5 morum     X A
ADJ   0 0 DAT P X SUPER 4 3 mis       X A
ADJ   0 0 ACC P M SUPER 4 3 mos       X A
ADJ   0 0 ABL P X SUPER 4 3 mis       X A
ADJ   0 0 VOC P M SUPER 4 2 mi        X A
ADJ   0 0 NOM S F SUPER 4 2 ma        X A
ADJ   0 0 GEN S F SUPER 4 3 mae       X A
ADJ   0 0 DAT S F SUPER 4 3 mae       X A
ADJ   0 0 ACC S F SUPER 4 3 mam       X A
ADJ   0 0 ABL S F SUPER 4 2 ma        X A
```

```
ADJ   0 0 VOC S F SUPER 4 2 ma        X A
ADJ   0 0 NOM P F SUPER 4 3 mae       X A
ADJ   0 0 GEN P F SUPER 4 5 marum     X A
ADJ   0 0 ACC P F SUPER 4 3 mas       X A
ADJ   0 0 VOC P F SUPER 4 3 mae       X A
ADJ   0 0 NOM S N SUPER 4 3 mum      X A
ADJ   0 0 GEN S N SUPER 4 2 mi       X A
ADJ   0 0 DAT S N SUPER 4 2 mo       X A
ADJ   0 0 ACC S N SUPER 4 3 mum      X A
ADJ   0 0 ABL S N SUPER 4 2 mo       X A
ADJ   0 0 VOC S N SUPER 4 3 mum      X A
ADJ   0 0 NOM P N SUPER 4 2 ma       X A
ADJ   0 0 GEN P N SUPER 4 5 morum    X A
ADJ   0 0 ACC P N SUPER 4 2 ma       X A
ADJ   0 0 VOC P N SUPER 4 2 ma       X A
--  Ex: miser misera miserum  =>  miser miser miseri miserri
--  Ex: sacer sacra sacrum  =>  sacer sacr zzz  sacerri     --  no COMP
--  Ex: pulcher pulchri  =>  pulcher pulchr pulchri pulcherri
ADJ   1 2 NOM S M POS   1 0          X A
ADJ   1 2 GEN S M POS   2 1 i        X A
ADJ   1 2 DAT S M POS   2 1 o        X A
ADJ   1 2 VOC S M POS   1 0          X A
ADJ   1 2 GEN S F POS   2 2 ae       X A
ADJ   1 2 DAT S F POS   2 2 ae       X A
ADJ   1 2 NOM S N POS   2 2 um       X A
ADJ   1 2 GEN S N POS   2 1 i        X A
ADJ   1 2 DAT S N POS   2 1 o        X A
ADJ   1 2 ACC S N POS   2 2 um       X A
ADJ   1 2 VOC S N POS   2 2 um       X A
--  nullus (gen) nullius  =>  null null zzz zzz   --  no COMP or SUPER
ADJ   1 3 NOM S M POS   1 2 us       X A
ADJ   1 3 GEN S X POS   2 3 ius      X A
ADJ   1 3 DAT S X POS   2 1 i        X A
ADJ   1 3 VOC S M POS   1 2 us       X A
ADJ   1 3 NOM S N POS   2 2 um       X A
ADJ   1 3 ACC S N POS   2 2 um       X A
ADJ   1 3 VOC S N POS   2 2 um       X A
--  alter, altera, alterum   =>  alter   alter
--  neuter, neutra, neutrum  =>  neuter  neutr
ADJ   1 4 NOM S M POS   1 0          X A
ADJ   1 4 GEN S X POS   2 3 ius      X A
ADJ   1 4 DAT S X POS   2 1 i        X A
ADJ   1 4 VOC S M POS   1 0          X A
ADJ   1 4 NOM S N POS   2 2 um       X A
ADJ   1 4 ACC S N POS   2 2 um       X A
ADJ   1 4 VOC S N POS   2 2 um       X A
--  alius, alia, aliud => ali ali
ADJ   1 5 NOM S M POS   1 2 us       X A
ADJ   1 5 GEN S X POS   2 2 us       X A
ADJ   1 5 GEN S M POS   2 1 i        X A
ADJ   1 5 GEN S M POS   2 0          X A
```

```
ADJ   1 5 DAT S M POS   2 1 o        X A
ADJ   1 5 VOC S M POS   1 2 us       X A
ADJ   1 5 GEN S F POS   2 2 ae       X A
ADJ   1 5 DAT S M POS   2 1 o        X A
ADJ   1 5 NOM S N POS   2 2 ud       X A
ADJ   1 5 NOM S N POS   2 1 d        X A
ADJ   1 5 NOM S N POS   2 2 ut       X A
ADJ   1 5 GEN S N POS   2 1 i        X A
ADJ   1 5 GEN S N POS   2 0          X A
ADJ   1 5 DAT S M POS   2 0          X A
ADJ   1 5 ACC S N POS   2 2 ud       X A
ADJ   1 5 ACC S N POS   2 2 ut       X A
ADJ   1 5 VOC S N POS   2 2 ud       X A
ADJ   1 5 ABL P N POS   2 3 eis      X A
ADJ   2 1 NOM S F X 1 1 e         X A
ADJ   2 1 VOC S F X 1 1 e         X A
ADJ   2 1 GEN S F X 2 2 es        X A
ADJ   2 1 LOC S F X 2 2 es        X A
ADJ   2 1 DAT S F X 2 2 ae        X A
ADJ   2 1 ABL S F X 2 1 e         X A
ADJ   2 1 ACC S F X 2 2 en        X A
ADJ   2 2 NOM S F POS   2 1 a     X A
ADJ   2 2 GEN S F POS   2 2 ae    X A
ADJ   2 2 DAT S F POS   2 2 ae    X A
ADJ   2 2 ACC S F POS   2 2 am    X A
ADJ   2 2 ABL S F POS   2 1 a     X A
ADJ   2 2 VOC S F POS   2 1 a     X A
--  es, es, es adjectives
ADJ   2 3 NOM S X X 1 2 es        X A
ADJ   2 3 VOC S X X 1 1 e         X A
ADJ   2 3 VOC S X X 1 1 a         X B
ADJ   2 3 GEN S X X 2 2 ae        X A
ADJ   2 3 LOC S X X 2 2 ae        X A
ADJ   2 3 DAT S X X 2 2 ae        X A
ADJ   2 3 ABL S X X 2 1 e         X A
ADJ   2 3 ABL S X X 2 1 a         X B
ADJ   2 3 ACC S X X 2 2 en        X A
ADJ   2 3 ACC S X X 2 2 am        X B
--  os, os, -
ADJ   2 6 NOM S C X 1 2 os        X A
ADJ   2 6 VOC S C X 1 1 e         X A
ADJ   2 6 GEN S C X 2 1 i         X A
ADJ   2 6 DAT S C X 2 1 o         X A
ADJ   2 6 ABL S C X 2 1 o         X A
ADJ   2 6 ACC S C X 2 2 on        X A
--  os, -, -
ADJ   2 7 NOM S M X 1 2 os        X A
ADJ   2 7 VOC S M X 1 1 e         X A
ADJ   2 7 GEN S M X 2 1 i         X A
ADJ   2 7 DAT S M X 2 1 o         X A
ADJ   2 7 ABL S M X 2 1 o         X A
```

```
ADJ   2 7 ACC S M X 2 2 on        X A
-- -, -, on
ADJ   2 8 NOM S N X 1 2 on        X A
ADJ   2 8 VOC S N X 2 2 on        X A
ADJ   2 8 GEN S X X 2 1 i         X A
ADJ   2 8 DAT S X X 2 1 o         X A
ADJ   2 8 ABL S X X 2 1 o         X A
ADJ   2 8 ACC S X X 2 2 on        X A
--  Plurals are like regular Latin ADJ 1 1
ADJ   2 0 NOM P M POS   2 1 i     X A
ADJ   2 0 GEN P M POS   2 4 orum  X A
ADJ   2 0 DAT P X POS   2 2 is    X A
ADJ   2 0 ACC P M POS   2 2 os    X A
ADJ   2 0 ABL P X POS   2 2 is    X A
ADJ   2 0 VOC P M POS   2 1 i     X A
ADJ   2 0 NOM P F POS   2 2 ae    X A
ADJ   2 0 GEN P F POS   2 4 arum  X A
ADJ   2 0 ACC P F POS   2 2 as    X A
ADJ   2 0 VOC P F POS   2 2 ae    X A
ADJ   2 0 NOM P N POS   2 1 a     X A
ADJ   2 0 GEN P N POS   2 4 orum  X A
ADJ   2 0 ACC P N POS   2 1 a     X A
ADJ   2 0 VOC P N POS   2 1 a     X A
--  AGGETTIVI SECONDA CLASSE
--  Ex: audax (gen) audacis  =>  audax audac audaci audacissi
--  Ex: prudens prudentis  =>  prudens prudent prudenti prudentissi
ADJ   3 1 NOM S X POS   1 0          X A
ADJ   3 0 GEN S X POS   2 2 is       X A
ADJ   3 0 DAT S X POS   2 1 i        X A
ADJ   3 0 ACC S C POS   2 2 em       X A
ADJ   3 0 ABL S X POS   2 1 i        X A
ADJ   3 0 ABL S X POS   2 1 e        X A
ADJ   3 1 VOC S X POS   1 0          X A
ADJ   3 0 NOM P C POS   2 2 es       X A
ADJ   3 0 GEN P X POS   2 3 ium      X A
ADJ   3 0 GEN P X POS   2 2 um       X A
ADJ   3 0 DAT P X POS   2 4 ibus     X A
ADJ   3 0 ACC P C POS   2 2 es       X A
ADJ   3 0 ACC P C POS   2 2 is       X A
ADJ   3 0 ABL P X POS   2 4 ibus     X A
ADJ   3 0 VOC P C POS   2 2 es       X A
ADJ   3 1 ACC S N POS   1 0          X A
ADJ   3 0 NOM P N POS   2 2 ia       X A
ADJ   3 0 ACC P N POS   2 2 ia       X A
ADJ   3 0 VOC P N POS   2 2 ia       X A
--  Ex: brevis breve  =>  brev brev brevi brevissi
--  Ex: facil facil   =>  facil facil facili facilli
ADJ   3 2 NOM S C POS   1 2 is       X A
ADJ   3 2 VOC S C POS   1 2 is       X A
ADJ   3 2 NOM S N POS   2 1 e        X A
ADJ   3 2 ACC S N POS   2 1 e        X A
```

```
ADJ   3 2 VOC S N POS   2 1 e        X A
ADJ   3 2 NOM P C POS   2 3 eis      F B
ADJ   3 2 ACC P C POS   2 3 eis      F B
--  Ex: celer celeris  celere  =>  celer celer celeri celerri
--  Ex: acer acris acre  =>  acer acr acri acerri
ADJ   3 3 NOM S M POS   1 0          X A
ADJ   3 3 VOC S M POS   1 0          X A
ADJ   3 3 NOM S F POS   2 2 is       X A
ADJ   3 3 VOC S F POS   2 2 is       X A
ADJ   3 3 NOM S N POS   2 1 e        X A
ADJ   3 3 ACC S N POS   2 1 e        X A
ADJ   3 3 VOC S N POS   2 1 e        X A
--  Ex: amethystizon amethystizontos => amethystizon amethystizont
ADJ   3 6 NOM S X POS   1 0          X A
ADJ   3 6 GEN S X POS   2 2 os       X A
ADJ   3 6 ACC S C POS   2 1 a        X A
ADJ   3 6 ACC S N POS   2 0          X A
ADJ   3 6 ACC P C POS   2 2 as       X A
--  Non declinati
ADJ   9 9 X X X X 1 0            X A
-- VERBI
--  Ex: voco vocare vocavi vocatus  =>  voc voc vocav vocat
--  Ex: porto portave portavi portatus  =>  port port portav portat
V     1 1 PRES  ACTIVE  IND  1 S  1 1 o             X A
V     1 1 PRES  ACTIVE  IND  2 S  2 2 as            X A
V     1 1 PRES  ACTIVE  IND  3 S  2 2 at            X A
V     1 1 PRES  ACTIVE  IND  1 P  2 4 amus          X A
V     1 1 PRES  ACTIVE  IND  2 P  2 4 atis          X A
V     1 1 PRES  ACTIVE  IND  3 P  1 3 ant           X A
V     1 1 IMPF  ACTIVE  IND  1 S  1 4 abam          X A
V     1 1 IMPF  ACTIVE  IND  2 S  1 4 abas          X A
V     1 1 IMPF  ACTIVE  IND  3 S  1 4 abat          X A
V     1 1 IMPF  ACTIVE  IND  1 P  1 6 abamus        X A
V     1 1 IMPF  ACTIVE  IND  2 P  1 6 abatis        X A
V     1 1 IMPF  ACTIVE  IND  3 P  1 5 abant         X A
V     1 1 FUT   ACTIVE  IND  1 S  1 3 abo           X A
V     1 1 FUT   ACTIVE  IND  2 S  1 4 abis          X A
V     1 1 FUT   ACTIVE  IND  3 S  1 4 abit          X A
V     1 1 FUT   ACTIVE  IND  1 P  1 6 abimus        X A
V     1 1 FUT   ACTIVE  IND  2 P  1 6 abitis        X A
V     1 1 FUT   ACTIVE  IND  3 P  1 5 abunt         X A
V     0 0 PERF  ACTIVE  IND  1 S  3 1 i             X A
V     0 0 PERF  ACTIVE  IND  2 S  3 4 isti          X A
V     0 0 PERF  ACTIVE  IND  3 S  3 2 it            X A
V     0 0 PERF  ACTIVE  IND  1 P  3 4 imus          X A
V     0 0 PERF  ACTIVE  IND  2 P  3 5 istis         X A
V     0 0 PERF  ACTIVE  IND  3 P  3 5 erunt         X A
V     0 0 PERF  ACTIVE  IND  3 P  3 3 ere           X B
V     0 0 PLUP  ACTIVE  IND  1 S  3 4 eram          X A
V     0 0 PLUP  ACTIVE  IND  2 S  3 4 eras          X A
V     0 0 PLUP  ACTIVE  IND  3 S  3 4 erat          X A
```

```
V     0 0 PLUP  ACTIVE  IND  1 P  3 6 eramus        X A
V     0 0 PLUP  ACTIVE  IND  2 P  3 6 eratis        X A
V     0 0 PLUP  ACTIVE  IND  3 P  3 5 erant         X A
V     0 0 FUTP  ACTIVE  IND  1 S  3 3 ero           X A
V     0 0 FUTP  ACTIVE  IND  2 S  3 4 eris          X A
V     0 0 FUTP  ACTIVE  IND  3 S  3 4 erit          X A
V     0 0 FUTP  ACTIVE  IND  1 P  3 6 erimus        X A
V     0 0 FUTP  ACTIVE  IND  2 P  3 6 eritis        X A
V     0 0 FUTP  ACTIVE  IND  3 P  3 5 erint         X A
V     1 1 PRES  PASSIVE IND  1 S  1 2 or            X A
V     1 1 PRES  PASSIVE IND  2 S  2 4 aris          X A
V     1 1 PRES  PASSIVE IND  2 S  2 3 are           B B
V     1 1 PRES  PASSIVE IND  3 S  2 4 atur          X A
V     1 1 PRES  PASSIVE IND  1 P  2 4 amur          X A
V     1 1 PRES  PASSIVE IND  2 P  2 5 amini         X A
V     1 1 PRES  PASSIVE IND  3 P  1 5 antur         X A
V     1 1 IMPF  PASSIVE IND  1 S  1 4 abar          X A
V     1 1 IMPF  PASSIVE IND  2 S  1 6 abaris        X A
V     1 1 IMPF  PASSIVE IND  2 S  1 5 abare         X A
V     1 1 IMPF  PASSIVE IND  3 S  1 6 abatur        X A
V     1 1 IMPF  PASSIVE IND  1 P  1 6 abamur        X A
V     1 1 IMPF  PASSIVE IND  2 P  1 7 abamini       X A
V     1 1 IMPF  PASSIVE IND  3 P  1 7 abantur       X A
V     1 1 FUT   PASSIVE IND  1 S  1 4 abor          X A
V     1 1 FUT   PASSIVE IND  2 S  1 6 aberis        X A
V     1 1 FUT   PASSIVE IND  2 S  1 5 abere         X A
V     1 1 FUT   PASSIVE IND  3 S  1 6 abitur        X A
V     1 1 FUT   PASSIVE IND  1 P  1 6 abimur        X A
V     1 1 FUT   PASSIVE IND  2 P  1 7 abimini       X A
V     1 1 FUT   PASSIVE IND  3 P  1 7 abuntur       X A
V     1 1 PRES  ACTIVE  SUB  1 S  1 2 em            X A
V     1 1 PRES  ACTIVE  SUB  2 S  1 2 es            X A
V     1 1 PRES  ACTIVE  SUB  3 S  1 2 et            X A
V     1 1 PRES  ACTIVE  SUB  1 P  1 4 emus          X A
V     1 1 PRES  ACTIVE  SUB  2 P  1 4 etis          X A
V     1 1 PRES  ACTIVE  SUB  3 P  1 3 ent           X A
V     1 1 IMPF  ACTIVE  SUB  1 S  2 4 arem          X A
V     1 1 IMPF  ACTIVE  SUB  2 S  2 4 ares          X A
V     1 1 IMPF  ACTIVE  SUB  3 S  2 4 aret          X A
V     1 1 IMPF  ACTIVE  SUB  1 P  2 6 aremus        X A
V     1 1 IMPF  ACTIVE  SUB  2 P  2 6 aretis        X A
V     1 1 IMPF  ACTIVE  SUB  3 P  2 5 arent         X A
V     0 0 PERF  ACTIVE  SUB  1 S  3 4 erim          X A
V     0 0 PERF  ACTIVE  SUB  2 S  3 4 eris          X A
V     0 0 PERF  ACTIVE  SUB  3 S  3 4 erit          X A
V     0 0 PERF  ACTIVE  SUB  1 P  3 6 erimus        X A
V     0 0 PERF  ACTIVE  SUB  2 P  3 6 eritis        X A
V     0 0 PERF  ACTIVE  SUB  3 P  3 5 erint         X A
V     0 0 PLUP  ACTIVE  SUB  1 S  3 5 issem         X A
V     0 0 PLUP  ACTIVE  SUB  2 S  3 5 isses         X A
V     0 0 PLUP  ACTIVE  SUB  3 S  3 5 isset         X A
```

```
V     0 0 PLUP  ACTIVE  SUB  1 P  3 7 issemus       X A
V     0 0 PLUP  ACTIVE  SUB  2 P  3 7 issetis       X A
V     0 0 PLUP  ACTIVE  SUB  3 P  3 6 issent        X A
V     1 1 PRES  PASSIVE SUB  1 S  1 2 er            X A
V     1 1 PRES  PASSIVE SUB  2 S  1 4 eris          X A
V     1 1 PRES  PASSIVE SUB  2 S  1 3 ere           X A
V     1 1 PRES  PASSIVE SUB  3 S  1 4 etur          X A
V     1 1 PRES  PASSIVE SUB  1 P  1 4 emur          X A
V     1 1 PRES  PASSIVE SUB  2 P  1 5 emini         X A
V     1 1 PRES  PASSIVE SUB  3 P  1 5 entur         X A
V     1 1 IMPF  PASSIVE SUB  1 S  2 4 arer          X A
V     1 1 IMPF  PASSIVE SUB  2 S  2 6 areris        X A
V     1 1 IMPF  PASSIVE SUB  2 S  2 5 arere         X A
V     1 1 IMPF  PASSIVE SUB  3 S  2 6 aretur        X A
V     1 1 IMPF  PASSIVE SUB  1 P  2 6 aremur        X A
V     1 1 IMPF  PASSIVE SUB  2 P  2 7 aremini       X A
V     1 1 IMPF  PASSIVE SUB  3 P  2 7 arentur       X A
V     1 1 PRES  ACTIVE  IMP  2 S  2 1 a             X A
V     1 1 PRES  ACTIVE  IMP  2 P  2 3 ate           X A
V     1 1 FUT   ACTIVE  IMP  2 S  2 3 ato           X A
V     1 1 FUT   ACTIVE  IMP  3 S  2 3 ato           X A
V     1 1 FUT   ACTIVE  IMP  2 P  2 5 atote         X A
V     1 1 FUT   ACTIVE  IMP  3 P  2 4 anto          X A
V     1 1 PRES  PASSIVE IMP  2 S  2 3 are           X A
V     1 1 PRES  PASSIVE IMP  2 P  2 5 amini         X A
V     1 1 FUT   PASSIVE IMP  2 S  2 4 ator          X A
V     1 1 FUT   PASSIVE IMP  3 S  2 4 ator          X A
V     1 1 FUT   PASSIVE IMP  3 P  2 5 antor         X A
V     1 1 PRES  ACTIVE  INF  0 X  2 3 are           X A
V     1 1 PERF  ACTIVE  INF  0 X  3 4 isse          X A
V     1 1 PRES  PASSIVE INF  0 X  2 3 ari           X A
V     1 1 PRES  PASSIVE INF  0 X  2 5 arier         B B
--  V 2 1
--  Ex:  moneo monere monui monitum  =>  mon mon monu monit
--  Ex:  habeo habere habui habitus  =>  hab hab habu habit
--  Ex:  deleo delere delevi deletus  =>  del del delev delet
--  Ex:  iubeo iubere iussi iussus  =>   iub iub iuss iuss
--  Ex:  video videre vidi visus  =>  vid vid vid vis
V     2 1 PRES  ACTIVE  IND  1 S  1 2 eo            X A
V     2 1 PRES  ACTIVE  IND  2 S  2 2 es            X A
V     2 1 PRES  ACTIVE  IND  3 S  2 2 et            X A
V     2 1 PRES  ACTIVE  IND  1 P  2 4 emus          X A
V     2 1 PRES  ACTIVE  IND  2 P  2 4 etis          X A
V     2 1 PRES  ACTIVE  IND  3 P  1 3 ent           X A
V     2 1 IMPF  ACTIVE  IND  1 S  1 4 ebam          X A
V     2 1 IMPF  ACTIVE  IND  2 S  1 4 ebas          X A
V     2 1 IMPF  ACTIVE  IND  3 S  1 4 ebat          X A
V     2 1 IMPF  ACTIVE  IND  1 P  1 6 ebamus        X A
V     2 1 IMPF  ACTIVE  IND  2 P  1 6 ebatis        X A
V     2 1 IMPF  ACTIVE  IND  3 P  1 5 ebant         X A
V     2 1 FUT   ACTIVE  IND  1 S  1 3 ebo           X A
```

```
V     2 1 FUT   ACTIVE  IND  2 S  1 4 ebis          X A
V     2 1 FUT   ACTIVE  IND  3 S  1 4 ebit          X A
V     2 1 FUT   ACTIVE  IND  1 P  1 6 ebimus        X A
V     2 1 FUT   ACTIVE  IND  2 P  1 6 ebitis        X A
V     2 1 FUT   ACTIVE  IND  3 P  1 5 ebunt         X A
V     2 1 PRES  PASSIVE IND  1 S  1 3 eor           X A
V     2 1 PRES  PASSIVE IND  2 S  2 4 eris          X A
V     2 1 PRES  PASSIVE IND  2 S  2 3 ere           X A
V     2 1 PRES  PASSIVE IND  3 S  2 4 etur          X A
V     2 1 PRES  PASSIVE IND  1 P  2 4 emur          X A
V     2 1 PRES  PASSIVE IND  2 P  2 5 emini         X A
V     2 1 PRES  PASSIVE IND  3 P  1 5 entur         X A
V     2 1 IMPF  PASSIVE IND  1 S  1 4 ebar          X A
V     2 1 IMPF  PASSIVE IND  2 S  1 6 ebaris        X A
V     2 1 IMPF  PASSIVE IND  2 S  1 5 ebare         X A
V     2 1 IMPF  PASSIVE IND  3 S  1 6 ebatur        X A
V     2 1 IMPF  PASSIVE IND  1 P  1 6 ebamur        X A
V     2 1 IMPF  PASSIVE IND  2 P  1 7 ebamini       X A
V     2 1 IMPF  PASSIVE IND  3 P  1 7 ebantur       X A
V     2 1 FUT   PASSIVE IND  1 S  1 4 ebor          X A
V     2 1 FUT   PASSIVE IND  2 S  1 6 eberis        X A
V     2 1 FUT   PASSIVE IND  2 S  1 5 ebere         X A
V     2 1 FUT   PASSIVE IND  3 S  1 6 ebitur        X A
V     2 1 FUT   PASSIVE IND  1 P  1 6 ebimur        X A
V     2 1 FUT   PASSIVE IND  2 P  1 7 ebimini       X A
V     2 1 FUT   PASSIVE IND  3 P  1 7 ebuntur       X A
V     2 1 PRES  ACTIVE  SUB  1 S  1 3 eam           X A
V     2 1 PRES  ACTIVE  SUB  2 S  1 3 eas           X A
V     2 1 PRES  ACTIVE  SUB  3 S  1 3 eat           X A
V     2 1 PRES  ACTIVE  SUB  1 P  1 5 eamus         X A
V     2 1 PRES  ACTIVE  SUB  2 P  1 5 eatis         X A
V     2 1 PRES  ACTIVE  SUB  3 P  1 4 eant          X A
V     2 1 IMPF  ACTIVE  SUB  1 S  2 4 erem          X A
V     2 1 IMPF  ACTIVE  SUB  2 S  2 4 eres          X A
V     2 1 IMPF  ACTIVE  SUB  3 S  2 4 eret          X A
V     2 1 IMPF  ACTIVE  SUB  1 P  2 6 eremus        X A
V     2 1 IMPF  ACTIVE  SUB  2 P  2 6 eretis        X A
V     2 1 IMPF  ACTIVE  SUB  3 P  2 5 erent         X A
V     2 1 PRES  PASSIVE SUB  1 S  1 3 ear           X A
V     2 1 PRES  PASSIVE SUB  2 S  1 5 earis         X A
V     2 1 PRES  PASSIVE SUB  2 S  1 4 eare          B B
V     2 1 PRES  PASSIVE SUB  3 S  1 5 eatur         X A
V     2 1 PRES  PASSIVE SUB  1 P  1 5 eamur         X A
V     2 1 PRES  PASSIVE SUB  2 P  1 6 eamini        X A
V     2 1 PRES  PASSIVE SUB  3 P  1 6 eantur        X A
V     2 1 IMPF  PASSIVE SUB  1 S  2 4 erer          X A
V     2 1 IMPF  PASSIVE SUB  2 S  2 6 ereris        X A
V     2 1 IMPF  PASSIVE SUB  2 S  2 5 erere         X A
V     2 1 IMPF  PASSIVE SUB  3 S  2 6 eretur        X A
V     2 1 IMPF  PASSIVE SUB  1 P  2 6 eremur        X A
V     2 1 IMPF  PASSIVE SUB  2 P  2 7 eremini       X A
```

```
V     2 1 IMPF  PASSIVE SUB  3 P  2 7 erentur       X A
V     2 1 PRES  ACTIVE  IMP  2 S  2 1 e             X A
V     2 1 PRES  ACTIVE  IMP  2 P  2 3 ete           X A
V     2 1 FUT   ACTIVE  IMP  2 S  2 3 eto           X A
V     2 1 FUT   ACTIVE  IMP  3 S  2 3 eto           X A
V     2 1 FUT   ACTIVE  IMP  2 P  2 5 etote         X A
V     2 1 FUT   ACTIVE  IMP  3 P  2 4 ento          X A
V     2 1 PRES  PASSIVE IMP  2 S  2 3 ere           X A
V     2 1 PRES  PASSIVE IMP  2 P  2 5 emini         X A
V     2 1 FUT   PASSIVE IMP  2 S  2 4 etor          X A
V     2 1 FUT   PASSIVE IMP  3 S  2 4 etor          X A
V     2 1 FUT   PASSIVE IMP  3 P  2 5 entor         X A
V     2 1 PRES  ACTIVE  INF  0 X  2 3 ere           X A
V     2 1 PERF  ACTIVE  INF  0 X  3 4 isse          X A
V     2 1 PRES  PASSIVE INF  0 X  2 3 eri           X A
V     2 1 PRES  PASSIVE INF  0 X  2 5 erier         B B
--  Ex: rego regere rexi rectum  =>  reg reg rex rect
--  Ex: pono ponoere posui positus  =>  pon pon posu posit
--  Ex: capio capere cepi captus  => capi cap cep capt   --  I-stem too w/KEY
V     3 0 PRES  ACTIVE  IND  1 S  1 1 o             X A
V     3 1 PRES  ACTIVE  IND  2 S  2 2 is            X A
V     3 1 PRES  ACTIVE  IND  3 S  2 2 it            X A
V     3 1 PRES  ACTIVE  IND  1 P  2 4 imus          X A
V     3 1 PRES  ACTIVE  IND  2 P  2 4 itis          X A
V     3 0 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     3 0 IMPF  ACTIVE  IND  1 S  1 4 ebam          X A
V     3 0 IMPF  ACTIVE  IND  2 S  1 4 ebas          X A
V     3 0 IMPF  ACTIVE  IND  3 S  1 4 ebat          X A
V     3 0 IMPF  ACTIVE  IND  1 P  1 6 ebamus        X A
V     3 0 IMPF  ACTIVE  IND  2 P  1 6 ebatis        X A
V     3 0 IMPF  ACTIVE  IND  3 P  1 5 ebant         X A
V     3 0 FUT   ACTIVE  IND  1 S  1 2 am            X A
V     3 0 FUT   ACTIVE  IND  2 S  1 2 es            X A
V     3 0 FUT   ACTIVE  IND  3 S  1 2 et            X A
V     3 0 FUT   ACTIVE  IND  1 P  1 4 emus          X A
V     3 0 FUT   ACTIVE  IND  2 P  1 4 etis          X A
V     3 0 FUT   ACTIVE  IND  3 P  1 3 ent           X A
V     3 0 PRES  PASSIVE IND  1 S  1 2 or            X A
V     3 1 PRES  PASSIVE IND  2 S  2 4 eris          X A
V     3 1 PRES  PASSIVE IND  2 S  2 3 ere           B C
V     3 1 PRES  PASSIVE IND  3 S  2 4 itur          X A
V     3 0 PRES  PASSIVE IND  1 P  2 4 imur          X A
V     3 0 PRES  PASSIVE IND  2 P  2 5 imini         X A
V     3 0 PRES  PASSIVE IND  3 P  1 5 untur         X A
V     3 0 IMPF  PASSIVE IND  1 S  1 4 ebar          X A
V     3 0 IMPF  PASSIVE IND  2 S  1 6 ebaris        X A
V     3 0 IMPF  PASSIVE IND  2 S  1 5 ebare         X A
V     3 0 IMPF  PASSIVE IND  3 S  1 6 ebatur        X A
V     3 0 IMPF  PASSIVE IND  1 P  1 6 ebamur        X A
V     3 0 IMPF  PASSIVE IND  2 P  1 7 ebamini       X A
V     3 0 IMPF  PASSIVE IND  3 P  1 7 ebantur       X A
```

```
V     3 0 FUT   PASSIVE IND  1 S  1 2 ar           X A
V     3 0 FUT   PASSIVE IND  2 S  1 4 eris         X A
V     3 0 FUT   PASSIVE IND  2 S  1 3 ere          X A
V     3 0 FUT   PASSIVE IND  3 S  1 4 etur         X A
V     3 0 FUT   PASSIVE IND  1 P  1 4 emur         X A
V     3 0 FUT   PASSIVE IND  2 P  1 5 emini        X A
V     3 0 FUT   PASSIVE IND  3 P  1 5 entur        X A
V     3 0 PRES  ACTIVE  SUB  1 S  1 2 am           X A
V     3 0 PRES  ACTIVE  SUB  2 S  1 2 as           X A
V     3 0 PRES  ACTIVE  SUB  3 S  1 2 at           X A
V     3 0 PRES  ACTIVE  SUB  1 P  1 4 amus         X A
V     3 0 PRES  ACTIVE  SUB  2 P  1 4 atis         X A
V     3 0 PRES  ACTIVE  SUB  3 P  1 3 ant          X A
V     3 1 IMPF  ACTIVE  SUB  1 S  2 4 erem         X A
V     3 1 IMPF  ACTIVE  SUB  2 S  2 4 eres         X A
V     3 1 IMPF  ACTIVE  SUB  3 S  2 4 eret         X A
V     3 1 IMPF  ACTIVE  SUB  1 P  2 6 eremus       X A
V     3 1 IMPF  ACTIVE  SUB  2 P  2 6 eretis       X A
V     3 1 IMPF  ACTIVE  SUB  3 P  2 5 erent        X A
V     3 0 PRES  PASSIVE SUB  1 S  1 2 ar           X A
V     3 0 PRES  PASSIVE SUB  2 S  1 4 aris         X A
V     3 0 PRES  PASSIVE SUB  2 S  1 3 are          X A
V     3 0 PRES  PASSIVE SUB  3 S  1 4 atur         X A
V     3 0 PRES  PASSIVE SUB  1 P  1 4 amur         X A
V     3 0 PRES  PASSIVE SUB  2 P  1 5 amini        X A
V     3 0 PRES  PASSIVE SUB  3 P  1 5 antur        X A
V     3 1 IMPF  PASSIVE SUB  1 S  2 4 erer         X A
V     3 1 IMPF  PASSIVE SUB  2 S  2 6 ereris       X A
V     3 1 IMPF  PASSIVE SUB  2 S  2 5 erere        X A
V     3 1 IMPF  PASSIVE SUB  3 S  2 6 eretur       X A
V     3 1 IMPF  PASSIVE SUB  1 P  2 6 eremur       X A
V     3 1 IMPF  PASSIVE SUB  2 P  2 7 eremini      X A
V     3 1 IMPF  PASSIVE SUB  3 P  2 7 erentur      X A
V     3 1 PRES  ACTIVE  IMP  2 S  2 1 e            X A
V     3 1 PRES  ACTIVE  IMP  2 S  2 0              X A
V     3 1 PRES  ACTIVE  IMP  2 P  2 3 ite          X A
V     3 1 FUT   ACTIVE  IMP  2 S  2 3 ito          X A
V     3 1 FUT   ACTIVE  IMP  3 S  2 3 ito          X A
V     3 1 FUT   ACTIVE  IMP  2 P  2 5 itote        X A
V     3 1 FUT   ACTIVE  IMP  3 P  2 4 unto         X A
V     3 1 PRES  PASSIVE IMP  2 S  2 3 ere          X A
V     3 1 PRES  PASSIVE IMP  2 P  2 5 imini        X A
V     3 1 FUT   PASSIVE IMP  2 S  2 4 itor         X A
V     3 1 FUT   PASSIVE IMP  3 S  2 4 itor         X A
V     3 1 FUT   PASSIVE IMP  3 P  2 5 untor        X A
V     3 1 PRES  ACTIVE  INF  0 X  2 3 ere          X A
V     3 1 PERF  ACTIVE  INF  0 X  3 4 isse         X A
V     3 1 PRES  PASSIVE INF  0 X  2 1 i            X A
V     3 1 PRES  PASSIVE INF  0 X  2 3 ier          B B
--  Irregulari
--  Ex: fero ferre tuli latus  =>  fer ferr tul lat
```

```
V     3 2 PRES  ACTIVE  IND  2 S  1 1 s            X A
V     3 2 PRES  ACTIVE  IND  3 S  1 1 t            X A
V     3 2 PRES  ACTIVE  IND  1 P  1 4 imus         X A
V     3 2 PRES  ACTIVE  IND  2 P  1 3 tis          X A
V     3 2 PRES  PASSIVE IND  2 S  2 2 is           X A
V     3 2 PRES  PASSIVE IND  2 S  2 1 e            B C
V     3 2 PRES  PASSIVE IND  3 S  1 3 tur          X A
V     3 2 IMPF  ACTIVE  SUB  1 S  2 2 em           X A
V     3 2 IMPF  ACTIVE  SUB  2 S  2 2 es           X A
V     3 2 IMPF  ACTIVE  SUB  3 S  2 2 et           X A
V     3 2 IMPF  ACTIVE  SUB  1 P  2 4 emus         X A
V     3 2 IMPF  ACTIVE  SUB  2 P  2 4 etis         X A
V     3 2 IMPF  ACTIVE  SUB  3 P  2 3 ent          X A
V     3 2 IMPF  PASSIVE SUB  1 S  2 2 er           X A
V     3 2 IMPF  PASSIVE SUB  2 S  2 4 eris         X A
V     3 2 IMPF  PASSIVE SUB  2 S  2 3 ere          X A
V     3 2 IMPF  PASSIVE SUB  3 S  2 4 etur         X A
V     3 2 IMPF  PASSIVE SUB  1 P  2 4 emur         X A
V     3 2 IMPF  PASSIVE SUB  2 P  2 5 emini        X A
V     3 2 IMPF  PASSIVE SUB  3 P  2 5 entur        X A
V     3 2 PRES  ACTIVE  IMP  2 S  1 0              X A
V     3 2 PRES  ACTIVE  IMP  2 P  1 2 te           X A
V     3 2 PRES  PASSIVE IMP  2 S  2 1 e            X A
V     3 2 PRES  PASSIVE IMP  2 P  1 5 imini        X A
V     3 2 FUT   ACTIVE  IMP  2 S  1 2 to           X A
V     3 2 FUT   ACTIVE  IMP  3 S  1 2 to           X A
V     3 2 FUT   ACTIVE  IMP  2 P  1 4 tote         X A
V     3 2 FUT   ACTIVE  IMP  3 P  1 4 unto         X A
V     3 2 FUT   PASSIVE IMP  2 S  1 3 tor          X A
V     3 2 FUT   PASSIVE IMP  3 S  1 3 tor          X A
V     3 2 FUT   PASSIVE IMP  3 P  1 5 untor        X A
V     3 2 PRES  ACTIVE  INF  0 X  2 1 e            X A
V     3 2 PERF  ACTIVE  INF  0 X  3 4 isse         X A
V     3 2 PRES  PASSIVE INF  0 X  2 1 i            X A
V     3 2 PRES  PASSIVE INF  0 X  2 3 ier          B B
--  Ex: fio fieri factus sum   =>  fi fi zzz zzz
V     3 3 PRES  ACTIVE  IND  2 S  1 1 s            X A
V     3 3 PRES  ACTIVE  IND  3 S  1 1 t            X A
V     3 3 PRES  ACTIVE  IND  1 P  1 3 mus          X A
V     3 3 PRES  ACTIVE  IND  2 P  1 3 tis          X A
V     3 3 PRES  PASSIVE IND  2 S  2 4 eris         X A
V     3 3 PRES  PASSIVE IND  3 S  2 4 itur         X A
V     3 3 IMPF  ACTIVE  SUB  1 S  2 4 erem         X A
V     3 3 IMPF  ACTIVE  SUB  2 S  2 4 eres         X A
V     3 3 IMPF  ACTIVE  SUB  3 S  2 4 eret         X A
V     3 3 IMPF  ACTIVE  SUB  1 P  2 6 eremus       X A
V     3 3 IMPF  ACTIVE  SUB  2 P  2 6 eretis       X A
V     3 3 IMPF  ACTIVE  SUB  3 P  2 5 erent        X A
V     3 3 IMPF  PASSIVE SUB  1 S  2 4 erer         X A
V     3 3 IMPF  PASSIVE SUB  2 S  2 6 ereris       X A
V     3 3 IMPF  PASSIVE SUB  3 S  2 6 eretur       X A
```

```
V     3 3 IMPF  PASSIVE SUB  1 P  2 6 eremur        X A
V     3 3 IMPF  PASSIVE SUB  2 P  2 7 eremini       X A
V     3 3 IMPF  PASSIVE SUB  3 P  2 7 erentur       X A
V     3 3 PRES  ACTIVE  IMP  2 S  2 0               X A
V     3 3 PRES  ACTIVE  IMP  2 P  2 2 te            X A
V     3 3 FUT   ACTIVE  IMP  2 S  2 2 to            X A
V     3 3 FUT   ACTIVE  IMP  3 S  2 2 to            X A
V     3 3 PRES  ACTIVE  INF  0 X  2 3 eri           X A
--  Ex: audio audire audivi auditus  =>  audi aud audiv audit
V     3 4 PRES  ACTIVE  IND  2 S  2 2 is            X A
V     3 4 PRES  ACTIVE  IND  3 S  2 2 it            X A
V     3 4 PRES  ACTIVE  IND  1 P  2 4 imus          X A
V     3 4 PRES  ACTIVE  IND  2 P  2 4 itis          X A
V     3 4 IMPF  ACTIVE  IND  1 S  2 3 bam           E D
V     3 4 IMPF  ACTIVE  IND  2 S  2 3 bas           E D
V     3 4 IMPF  ACTIVE  IND  3 S  2 3 bat           E D
V     3 4 IMPF  ACTIVE  IND  1 P  2 5 bamus         E D
V     3 4 IMPF  ACTIVE  IND  2 P  2 5 batis         E D
V     3 4 IMPF  ACTIVE  IND  3 P  2 4 bant          E D
V     3 4 FUT   ACTIVE  IND  1 S  1 2 bo            X D
V     3 4 FUT   ACTIVE  IND  2 S  1 3 bis           X D
V     3 4 FUT   ACTIVE  IND  3 S  1 3 bit           X D
V     3 4 FUT   ACTIVE  IND  1 P  1 5 bimus         X D
V     3 4 FUT   ACTIVE  IND  2 P  1 5 bitis         X D
V     3 4 FUT   ACTIVE  IND  3 P  1 4 bunt          X D
V     3 4 PRES  PASSIVE IND  2 S  2 4 iris          X A
V     3 4 PRES  PASSIVE IND  2 S  2 3 ire           B D
V     3 4 PRES  PASSIVE IND  3 S  2 4 itur          X A
V     3 4 FUT   PASSIVE IND  1 S  1 3 bor           E E
V     3 4 FUT   PASSIVE IND  2 S  1 5 beris         E E
V     3 4 FUT   PASSIVE IND  3 S  1 5 berit         E E
V     3 4 FUT   PASSIVE IND  1 P  1 5 bimur         E E
V     3 4 FUT   PASSIVE IND  2 P  1 6 bimini        E E
V     3 4 FUT   PASSIVE IND  3 P  1 6 buntur        E E
V     3 4 IMPF  ACTIVE  SUB  1 S  2 4 irem          X A
V     3 4 IMPF  ACTIVE  SUB  2 S  2 4 ires          X A
V     3 4 IMPF  ACTIVE  SUB  3 S  2 4 iret          X A
V     3 4 IMPF  ACTIVE  SUB  1 P  2 6 iremus        X A
V     3 4 IMPF  ACTIVE  SUB  2 P  2 6 iretis        X A
V     3 4 IMPF  ACTIVE  SUB  3 P  2 5 irent         X A
V     3 4 IMPF  PASSIVE SUB  1 S  2 4 irer          X A
V     3 4 IMPF  PASSIVE SUB  2 S  2 6 ireris        X A
V     3 4 IMPF  PASSIVE SUB  2 S  2 5 irere         X A
V     3 4 IMPF  PASSIVE SUB  3 S  2 6 iretur        X A
V     3 4 IMPF  PASSIVE SUB  1 P  2 6 iremur        X A
V     3 4 IMPF  PASSIVE SUB  2 P  2 7 iremini       X A
V     3 4 IMPF  PASSIVE SUB  3 P  2 7 irentur       X A
V     3 4 PRES  ACTIVE  IMP  2 S  2 1 i             X A
V     3 4 PRES  ACTIVE  IMP  2 P  2 3 ite           X A
V     3 4 FUT   ACTIVE  IMP  2 S  2 3 ito           X A
V     3 4 FUT   ACTIVE  IMP  3 S  2 3 ito           X A
```

```
V     3 4 FUT   ACTIVE  IMP  2 P  2 5 itote         X A
V     3 4 FUT   ACTIVE  IMP  3 P  1 4 unto          X A
V     3 4 PRES  PASSIVE IMP  2 S  2 3 ire           X A
V     3 4 PRES  PASSIVE IMP  2 P  2 5 imini         X A
V     3 4 FUT   PASSIVE IMP  2 S  2 4 itor          X A
V     3 4 FUT   PASSIVE IMP  3 S  2 4 itor          X A
V     3 4 FUT   PASSIVE IMP  3 P  1 5 untor         X A
V     3 4 PRES  ACTIVE  INF  0 X  2 3 ire           X A
V     3 4 PERF  ACTIVE  INF  0 X  3 4 isse          X A
V     3 4 PRES  PASSIVE INF  0 X  2 3 iri           X A
V     3 4 PRES  PASSIVE INF  0 X  2 5 irier         B B
--  Ex: sum esse fui futurus  =>  s . fu fut
--  Ex: adsum adesse adfui adfuturus  =>  ads ad adfu adfut
V     5 0 PRES  ACTIVE  IND  1 S  1 2 um            X A
V     5 0 PRES  ACTIVE  IND  2 S  2 2 es            X A
V     5 0 PRES  ACTIVE  IND  3 S  2 3 est           X A
V     5 0 PRES  ACTIVE  IND  1 P  1 4 umus          X A
V     5 0 PRES  ACTIVE  IND  2 P  2 5 estis         X A
V     5 0 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     5 0 IMPF  ACTIVE  IND  1 S  2 4 eram          X A
V     5 0 IMPF  ACTIVE  IND  2 S  2 4 eras          X A
V     5 0 IMPF  ACTIVE  IND  3 S  2 4 erat          X A
V     5 0 IMPF  ACTIVE  IND  1 P  2 6 eramus        X A
V     5 0 IMPF  ACTIVE  IND  2 P  2 6 eratis        X A
V     5 0 IMPF  ACTIVE  IND  3 P  2 5 erant         X A
V     5 0 FUT   ACTIVE  IND  1 S  2 3 ero           X A
V     5 0 FUT   ACTIVE  IND  2 S  2 4 eris          X A
V     5 0 FUT   ACTIVE  IND  3 S  2 4 erit          X A
V     5 0 FUT   ACTIVE  IND  1 P  2 6 erimus        X A
V     5 0 FUT   ACTIVE  IND  2 P  2 6 eritis        X A
V     5 0 FUT   ACTIVE  IND  3 P  2 5 erunt         X A
V     5 0 FUT   ACTIVE  IND  3 P  2 5 erint         E D
V     5 0 PRES  ACTIVE  SUB  1 S  1 2 im            X A
V     5 0 PRES  ACTIVE  SUB  2 S  1 2 is            X A
V     5 0 PRES  ACTIVE  SUB  3 S  1 2 it            X A
V     5 0 PRES  ACTIVE  SUB  1 P  1 4 imus          X A
V     5 0 PRES  ACTIVE  SUB  2 P  1 4 itis          X A
V     5 0 PRES  ACTIVE  SUB  3 P  1 3 int           X A
V     5 1 IMPF  ACTIVE  SUB  1 S  2 5 essem         X A
V     5 1 IMPF  ACTIVE  SUB  2 S  2 5 esses         X A
V     5 1 IMPF  ACTIVE  SUB  3 S  2 5 esset         X A
V     5 1 IMPF  ACTIVE  SUB  1 P  2 7 essemus       X A
V     5 1 IMPF  ACTIVE  SUB  2 P  2 7 essetis       X A
V     5 1 IMPF  ACTIVE  SUB  3 P  2 6 essent        X A
V     5 1 IMPF  ACTIVE  SUB  1 S  2 5 forem         X A
V     5 1 IMPF  ACTIVE  SUB  2 S  2 5 fores         X A
V     5 1 IMPF  ACTIVE  SUB  3 S  2 5 foret         X A
V     5 1 IMPF  ACTIVE  SUB  1 P  2 7 foremus       X A
V     5 1 IMPF  ACTIVE  SUB  2 P  2 7 foretis       X A
V     5 1 IMPF  ACTIVE  SUB  3 P  2 6 forent        X A
V     5 1 PRES  ACTIVE  IMP  2 S  2 2 es            X A
```

```
V     5 1 PRES  ACTIVE  IMP  2 P  2 4 este          X A
V     5 1 FUT   ACTIVE  IMP  2 S  2 4 esto          X A
V     5 1 FUT   ACTIVE  IMP  3 S  2 4 esto          X A
V     5 1 FUT   ACTIVE  IMP  2 P  2 6 estote        X A
V     5 1 FUT   ACTIVE  IMP  3 P  1 4 unto          X A
V     5 1 PRES  ACTIVE  INF  0 X  2 4 esse          X A
V     5 0 PERF  ACTIVE  INF  0 X  3 4 isse          X A
V     5 0 FUT   ACTIVE  INF  0 X  2 4 fore          X A
--  Ex: possum posse potui -  =>  poss pot potu -
V     5 2 IMPF  ACTIVE  SUB  1 S  1 2 em            X A
V     5 2 IMPF  ACTIVE  SUB  2 S  1 2 es            X A
V     5 2 IMPF  ACTIVE  SUB  3 S  1 2 et            X A
V     5 2 IMPF  ACTIVE  SUB  1 P  1 4 emus          X A
V     5 2 IMPF  ACTIVE  SUB  2 P  1 4 etis          X A
V     5 2 IMPF  ACTIVE  SUB  3 P  1 3 ent           X A
V     5 2 FUT   ACTIVE  IND  3 P  2 5 erint         X E
V     5 2 PRES  ACTIVE  INF  0 X  1 1 e             X A
V     5 2 PERF  ACTIVE  INF  0 X  3 4 isse          X A
--  Ex: eo ire ivi itus  =>  e i iv (i) it
V     6 1 PRES  ACTIVE  IND  1 S  1 1 o             X A
V     6 1 PRES  ACTIVE  IND  2 S  2 1 s             X A
V     6 1 PRES  ACTIVE  IND  3 S  2 1 t             X A
V     6 1 PRES  ACTIVE  IND  1 P  2 3 mus           X A
V     6 1 PRES  ACTIVE  IND  2 P  2 3 tis           X A
V     6 1 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     6 1 IMPF  ACTIVE  IND  1 S  2 3 bam           X A
V     6 1 IMPF  ACTIVE  IND  2 S  2 3 bas           X A
V     6 1 IMPF  ACTIVE  IND  3 S  2 3 bat           X A
V     6 1 IMPF  ACTIVE  IND  1 P  2 5 bamus         X A
V     6 1 IMPF  ACTIVE  IND  2 P  2 5 batis         X A
V     6 1 IMPF  ACTIVE  IND  3 P  2 4 bant          X A
V     6 1 IMPF  ACTIVE  IND  1 S  2 4 ebam          D B
V     6 1 IMPF  ACTIVE  IND  2 S  2 4 ebas          D B
V     6 1 IMPF  ACTIVE  IND  3 S  2 4 ebat          D B
V     6 1 IMPF  ACTIVE  IND  1 P  2 6 ebamus        D B
V     6 1 IMPF  ACTIVE  IND  2 P  2 6 ebatis        D B
V     6 1 IMPF  ACTIVE  IND  3 P  2 5 ebant         D B
V     6 1 FUT   ACTIVE  IND  1 S  2 2 bo            X A
V     6 1 FUT   ACTIVE  IND  2 S  2 3 bis           X A
V     6 1 FUT   ACTIVE  IND  3 S  2 3 bit           X A
V     6 1 FUT   ACTIVE  IND  1 P  2 5 bimus         X A
V     6 1 FUT   ACTIVE  IND  2 P  2 5 bitis         X A
V     6 1 FUT   ACTIVE  IND  3 P  2 4 bunt          X A
V     6 1 FUT   ACTIVE  IND  1 S  2 2 am            D B
V     6 1 FUT   ACTIVE  IND  2 S  2 2 es            D B
V     6 1 FUT   ACTIVE  IND  3 S  2 2 et            D B
V     6 1 FUT   ACTIVE  IND  1 P  2 4 emus          D B
V     6 1 FUT   ACTIVE  IND  2 P  2 4 etis          D B
V     6 1 FUT   ACTIVE  IND  3 P  2 3 ent           D B
V     6 1 PRES  PASSIVE IND  1 S  1 2 or            X A
V     6 1 PRES  PASSIVE IND  2 S  2 3 ris           X A
```

```
V     6 1 PRES  PASSIVE IND  2 S  2 2 re           X A
V     6 1 PRES  PASSIVE IND  3 S  2 3 tur          X A
V     6 1 PRES  PASSIVE IND  1 P  2 3 mur          X A
V     6 1 PRES  PASSIVE IND  2 P  2 4 mini         X A
V     6 1 PRES  PASSIVE IND  3 P  1 5 untur        X A
V     6 1 IMPF  PASSIVE IND  1 S  2 3 bar         X A
V     6 1 IMPF  PASSIVE IND  2 S  2 5 baris       X A
V     6 1 IMPF  PASSIVE IND  2 S  2 4 bare        X A
V     6 1 IMPF  PASSIVE IND  3 S  2 5 batur       X A
V     6 1 IMPF  PASSIVE IND  1 P  2 5 bamur       X A
V     6 1 IMPF  PASSIVE IND  2 P  2 6 bamini      X A
V     6 1 IMPF  PASSIVE IND  3 P  2 6 bantur      X A
V     6 1 FUT   PASSIVE IND  1 S  2 3 bor           X A
V     6 1 FUT   PASSIVE IND  2 S  2 5 beris         X A
V     6 1 FUT   PASSIVE IND  2 S  2 4 bere          X A
V     6 1 FUT   PASSIVE IND  3 S  2 5 bitur         X A
V     6 1 FUT   PASSIVE IND  1 P  2 5 bimur         X A
V     6 1 FUT   PASSIVE IND  2 P  2 6 bimini        X A
V     6 1 FUT   PASSIVE IND  3 P  2 6 buntur        X A
V     6 1 FUT   PASSIVE IND  1 S  2 2 ar          E C
V     6 1 FUT   PASSIVE IND  2 S  2 4 eris        E C
V     6 1 FUT   PASSIVE IND  2 S  2 3 ere         E C
V     6 1 FUT   PASSIVE IND  3 S  2 4 etur        E C
V     6 1 FUT   PASSIVE IND  1 P  2 4 emur        E C
V     6 1 FUT   PASSIVE IND  2 P  2 5 emini       E C
V     6 1 FUT   PASSIVE IND  3 P  2 5 entur       E C
V     6 1 PRES  ACTIVE  SUB  1 S  1 2 am          X A
V     6 1 PRES  ACTIVE  SUB  2 S  1 2 as          X A
V     6 1 PRES  ACTIVE  SUB  3 S  1 2 at          X A
V     6 1 PRES  ACTIVE  SUB  1 P  1 4 amus        X A
V     6 1 PRES  ACTIVE  SUB  2 P  1 4 atis        X A
V     6 1 PRES  ACTIVE  SUB  3 P  1 3 ant         X A
V     6 1 IMPF  ACTIVE  SUB  1 S  2 3 rem         X A
V     6 1 IMPF  ACTIVE  SUB  2 S  2 3 res         X A
V     6 1 IMPF  ACTIVE  SUB  3 S  2 3 ret         X A
V     6 1 IMPF  ACTIVE  SUB  1 P  2 5 remus       X A
V     6 1 IMPF  ACTIVE  SUB  2 P  2 5 retis       X A
V     6 1 IMPF  ACTIVE  SUB  3 P  2 4 rent        X A
V     6 1 PRES  PASSIVE SUB  1 S  1 2 ar          X A
V     6 1 PRES  PASSIVE SUB  2 S  2 4 aris        X A
V     6 1 PRES  PASSIVE SUB  2 S  2 3 are         X A
V     6 1 PRES  PASSIVE SUB  3 S  2 4 atur        X A
V     6 1 PRES  PASSIVE SUB  1 P  2 4 amur        X A
V     6 1 PRES  PASSIVE SUB  2 P  2 5 amini       X A
V     6 1 PRES  PASSIVE SUB  3 P  1 5 antur       X A
V     6 1 IMPF  PASSIVE SUB  1 S  2 3 rer         X A
V     6 1 IMPF  PASSIVE SUB  2 S  2 5 reris       X A
V     6 1 IMPF  PASSIVE SUB  2 S  2 4 rere        X A
V     6 1 IMPF  PASSIVE SUB  3 S  2 5 retur       X A
V     6 1 IMPF  PASSIVE SUB  1 P  2 5 remur       X A
V     6 1 IMPF  PASSIVE SUB  2 P  2 6 remini      X A
```

```
V     6 1 IMPF  PASSIVE SUB  3 P  2 6 rentur        X A
V     6 1 PRES  ACTIVE  IMP  2 S  2 0               X A
V     6 1 PRES  ACTIVE  IMP  2 P  2 2 te            X A
V     6 1 FUT   ACTIVE  IMP  2 S  2 2 to            X A
V     6 1 FUT   ACTIVE  IMP  3 S  2 2 to            X A
V     6 1 FUT   ACTIVE  IMP  2 P  2 4 tote          X A
V     6 1 FUT   ACTIVE  IMP  3 P  1 4 unto          X A
V     6 1 PRES  PASSIVE IMP  2 S  2 3 ere           X A
V     6 1 PRES  PASSIVE IMP  2 P  2 4 mini          X A
V     6 1 FUT   PASSIVE IMP  2 S  2 3 tor           X A
V     6 1 FUT   PASSIVE IMP  3 S  2 3 tor           X A
V     6 1 FUT   PASSIVE IMP  3 P  1 5 untor         X A
V     6 1 PRES  ACTIVE  INF  0 X  2 2 re            X A
V     6 1 PERF  ACTIVE  INF  0 X  3 4 isse          X A
V     6 1 PRES  PASSIVE INF  0 X  2 2 ri            X A
V     6 1 PRES  PASSIVE INF  0 X  2 4 rier          B B
--  Ex: volo velle volui -  =>  vol vel volu -
--  Ex: nolo nolle nolui -  =>  nol nol nolu -
--  Ex: malo malle malui -  =>  mal mal malu -
V     6 2 PRES  ACTIVE  IND  1 S  1 1 o             X A
V     6 2 PRES  ACTIVE  IND  1 P  1 4 umus          X A
V     6 2 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     6 2 IMPF  ACTIVE  IND  1 S  1 4 ebam          X A
V     6 2 IMPF  ACTIVE  IND  2 S  1 4 ebas          X A
V     6 2 IMPF  ACTIVE  IND  3 S  1 4 ebat          X A
V     6 2 IMPF  ACTIVE  IND  1 P  1 6 ebamus        X A
V     6 2 IMPF  ACTIVE  IND  2 P  1 6 ebatis        X A
V     6 2 IMPF  ACTIVE  IND  3 P  1 5 ebant         X A
V     6 2 FUT   ACTIVE  IND  1 S  1 2 am            X A
V     6 2 FUT   ACTIVE  IND  2 S  1 2 es            X A
V     6 2 FUT   ACTIVE  IND  3 S  1 2 et            X A
V     6 2 FUT   ACTIVE  IND  1 P  1 4 emus          X A
V     6 2 FUT   ACTIVE  IND  2 P  1 4 etis          X A
V     6 2 FUT   ACTIVE  IND  3 P  1 3 ent           X A
V     6 2 PRES  ACTIVE  SUB  1 S  2 2 im            X A
V     6 2 PRES  ACTIVE  SUB  2 S  2 2 is            X A
V     6 2 PRES  ACTIVE  SUB  3 S  2 2 it            X A
V     6 2 PRES  ACTIVE  SUB  1 P  2 4 imus          X A
V     6 2 PRES  ACTIVE  SUB  2 P  2 4 itis          X A
V     6 2 PRES  ACTIVE  SUB  3 P  2 3 int           X A
V     6 2 IMPF  ACTIVE  SUB  1 S  2 3 lem           X A
V     6 2 IMPF  ACTIVE  SUB  2 S  2 3 les           X A
V     6 2 IMPF  ACTIVE  SUB  3 S  2 3 let           X A
V     6 2 IMPF  ACTIVE  SUB  1 P  2 5 lemus         X A
V     6 2 IMPF  ACTIVE  SUB  2 P  2 5 letis         X A
V     6 2 IMPF  ACTIVE  SUB  3 P  2 4 lent          X A
V     6 2 PRES  ACTIVE  IMP  2 S  1 1 i             X A
V     6 2 PRES  ACTIVE  IMP  2 P  1 3 ite           X A
V     6 2 FUT   ACTIVE  IMP  2 S  1 3 ito           X A
V     6 2 FUT   ACTIVE  IMP  3 S  1 3 ito           X A
V     6 2 FUT   ACTIVE  IMP  2 P  1 5 itote         X A
```

```
V     6 2 FUT   ACTIVE  IMP  3 P  1 4 unto          X A
V     6 2 PRES  ACTIVE  INF  0 X  2 2 le            X A
V     6 2 PERF  ACTIVE  INF  0 X  3 4 isse          X A
--  Ex: aio  x   =>   ai  a  zzz  zzz
V     7 1 PRES  ACTIVE  IND  1 S  1 1 o             X A
V     7 1 PRES  ACTIVE  IND  2 S  2 2 is            X A
V     7 1 PRES  ACTIVE  IND  3 S  2 2 it            X A
V     7 1 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     7 1 IMPF  ACTIVE  IND  1 S  1 4 ebam          X A
V     7 1 IMPF  ACTIVE  IND  2 S  1 4 ebas          X A
V     7 1 IMPF  ACTIVE  IND  3 S  1 4 ebat          X A
V     7 1 IMPF  ACTIVE  IND  1 P  1 6 ebamus        X A
V     7 1 IMPF  ACTIVE  IND  2 P  1 6 ebatis        X A
V     7 1 IMPF  ACTIVE  IND  3 P  1 5 ebant         X A
V     7 1 IMPF  ACTIVE  IND  1 S  2 4 ibam          B B
V     7 1 IMPF  ACTIVE  IND  2 S  2 4 ibas          B B
V     7 1 IMPF  ACTIVE  IND  3 S  2 4 ibat          B B
V     7 1 IMPF  ACTIVE  IND  1 P  2 6 ibamus        B B
V     7 1 IMPF  ACTIVE  IND  2 P  2 6 ibatis        B B
V     7 1 IMPF  ACTIVE  IND  3 P  2 5 ibant         B B
V     7 1 PERF  ACTIVE  IND  3 S  2 2 it            X A
V     7 1 PRES  ACTIVE  SUB  2 S  2 3 ias           X A
V     7 1 PRES  ACTIVE  SUB  3 S  2 3 iat           X A
V     7 1 PRES  ACTIVE  SUB  3 P  2 4 iant          X A
V     7 1 PRES  ACTIVE  IMP  2 S  2 1 i             B D
V     7 2 PRES  ACTIVE  IND  1 S  2 2 am            X A
V     7 2 PRES  ACTIVE  IND  2 S  2 2 is            X A
V     7 2 PRES  ACTIVE  IND  3 S  2 2 it            X A
V     7 2 PRES  ACTIVE  IND  1 P  2 4 imus          X A
V     7 2 PRES  ACTIVE  IND  2 P  2 4 itis          X A
V     7 2 PRES  ACTIVE  IND  3 P  1 3 unt           X A
V     7 2 IMPF  ACTIVE  IND  3 S  1 4 ebat          X A
V     7 2 FUT   ACTIVE  IND  2 S  1 2 es            X A
V     7 2 FUT   ACTIVE  IND  3 S  1 2 et            X A
V     7 2 PERF  ACTIVE  IND  1 S  1 1 i             X A
V     7 2 PERF  ACTIVE  IND  2 S  2 4 isti          X A
V     7 2 PERF  ACTIVE  IND  3 S  2 2 it            X A
V     7 2 PRES  ACTIVE  IMP  2 S  2 1 e             X A
V     7 2 FUT   ACTIVE  IMP  2 S  2 3 ito           X A
V     7 2 FUT   ACTIVE  IMP  3 S  2 3 ito           X A
--  Ex: edo edere (esse) edi esus    =>   ed ed ed es (+ ed  es  zzz  zzz)
V     7 3 PRES  ACTIVE  IND  2 S  2 0               B B
V     7 3 PRES  ACTIVE  IND  3 S  2 1 t             B B
V     7 3 PRES  ACTIVE  IND  2 P  2 3 tis           B B
V     7 3 PRES  ACTIVE  SUB  1 S  1 2 im            B B
V     7 3 PRES  ACTIVE  SUB  2 S  1 2 is            B B
V     7 3 PRES  ACTIVE  SUB  3 S  1 2 it            B B
V     7 3 PRES  ACTIVE  SUB  1 P  1 4 imus          B B
V     7 3 PRES  ACTIVE  SUB  2 P  1 4 itis          B B
V     7 3 PRES  ACTIVE  SUB  3 P  1 3 int           B B
V     7 3 PRES  ACTIVE  IND  3 S  2 3 tur           B B
```

```
V     7 3 IMPF  ACTIVE  SUB  1 S  2 3 sem          B B
V     7 3 IMPF  ACTIVE  SUB  2 S  2 3 ses          B B
V     7 3 IMPF  ACTIVE  SUB  3 S  2 3 set          B B
V     7 3 IMPF  ACTIVE  SUB  1 P  2 5 semus        B B
V     7 3 IMPF  ACTIVE  SUB  2 P  2 5 setis        B B
V     7 3 IMPF  ACTIVE  SUB  3 P  2 4 sent         B B
V     7 3 IMPF  ACTIVE  SUB  3 S  2 5 setur        B B
V     7 3 PRES  ACTIVE  IMP  2 S  2 0              B B
V     7 3 PRES  ACTIVE  IMP  2 P  2 2 te           B B
V     7 3 FUT   ACTIVE  IMP  2 S  2 2 to           B B
V     7 3 FUT   ACTIVE  IMP  3 S  2 2 to           B B
V     7 3 FUT   ACTIVE  IMP  2 P  2 4 tote         B B
V     7 3 PRES  ACTIVE  INF  0 X  2 2 se           B B
--  Ex: faxo FUTP IND of facere, faxim PERF SUB, faxem PLUP SUB - stem 3 fax
--  Ex: capso FUTP of capere
--  Ex: duxim FUTP of ducere
V     8 0 FUTP  ACTIVE  IND  1 S  3 1 o          B C
V     8 0 FUTP  ACTIVE  IND  2 S  3 2 is         B C
V     8 0 FUTP  ACTIVE  IND  3 S  3 2 it         B C
V     8 0 FUTP  ACTIVE  IND  1 P  3 4 imus       B C
V     8 0 FUTP  ACTIVE  IND  2 P  3 4 itis       B C
V     8 0 FUTP  ACTIVE  IND  3 P  3 3 int        B C
V     8 0 PERF  ACTIVE  SUB  1 S  3 2 im         B C
V     8 0 PERF  ACTIVE  SUB  2 S  3 2 is         B C
V     8 0 PERF  ACTIVE  SUB  3 S  3 2 it         B C
V     8 0 PERF  ACTIVE  SUB  1 P  3 4 imus       B D
V     8 0 PERF  ACTIVE  SUB  2 P  3 4 itis       B D
V     8 0 PERF  ACTIVE  SUB  3 P  3 3 int        B D
V     8 0 PLUP  ACTIVE  SUB  1 S  3 2 em         B E
V     8 0 PLUP  ACTIVE  SUB  2 S  3 2 es         B E
V     8 0 PLUP  ACTIVE  SUB  3 S  3 2 et         B E
V     8 0 PLUP  ACTIVE  SUB  1 P  3 4 emus       B E
V     8 0 PLUP  ACTIVE  SUB  2 P  3 4 etis       B E
V     8 0 PLUP  ACTIVE  SUB  3 P  3 3 ent        B E
V     8 0 PRES  ACTIVE  INF  0 X  2 1 e          B C
V     9 8 X X X 0 X 1 0     X A
V     9 9 X X X 0 X  1 0           X A
-- PARTICIPI
VPAR 1 0 NOM S X PRES ACTIVE  PPL 1 3 ans         X A
VPAR 1 0 GEN S X PRES ACTIVE  PPL 1 5 antis       X A
VPAR 1 0 DAT S X PRES ACTIVE  PPL 1 4 anti        X A
VPAR 1 0 ACC S C PRES ACTIVE  PPL 1 5 antem       X A
VPAR 1 0 ABL S X PRES ACTIVE  PPL 1 4 anti        X A
VPAR 1 0 ABL S X PRES ACTIVE  PPL 1 4 ante        X A
VPAR 1 0 VOC S X PRES ACTIVE  PPL 1 3 ans         X A
VPAR 1 0 NOM P C PRES ACTIVE  PPL 1 5 antes       X A
VPAR 1 0 GEN P X PRES ACTIVE  PPL 1 6 antium      X A
VPAR 1 0 GEN P X PRES ACTIVE  PPL 1 5 antum       X C
VPAR 1 0 DAT P X PRES ACTIVE  PPL 1 7 antibus     X A
VPAR 1 0 ACC P C PRES ACTIVE  PPL 1 5 antes       X A
VPAR 1 0 ABL P X PRES ACTIVE  PPL 1 7 antibus     X A
```

```
VPAR 1 0 VOC P C PRES ACTIVE  PPL 1 5 antes       X A
VPAR 1 0 ACC S N PRES ACTIVE  PPL 1 3 ans         X A
VPAR 1 0 NOM P N PRES ACTIVE  PPL 1 5 antia       X A
VPAR 1 0 ACC P N PRES ACTIVE  PPL 1 5 antia       X A
VPAR 1 0 VOC P N PRES ACTIVE  PPL 1 5 antia       X A
VPAR 2 0 NOM S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 2 0 GEN S X PRES ACTIVE  PPL 1 5 entis       X A
VPAR 2 0 DAT S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 2 0 ACC S C PRES ACTIVE  PPL 1 5 entem       X A
VPAR 2 0 ABL S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 2 0 ABL S X PRES ACTIVE  PPL 1 4 ente        X A
VPAR 2 0 VOC S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 2 0 NOM P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 2 0 GEN P X PRES ACTIVE  PPL 1 6 entium      X A
VPAR 2 0 GEN P X PRES ACTIVE  PPL 1 5 entum       X C
VPAR 2 0 DAT P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 2 0 ACC P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 2 0 ABL P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 2 0 VOC P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 2 0 ACC S N PRES ACTIVE  PPL 1 3 ens         X A
VPAR 2 0 NOM P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 2 0 ACC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 2 0 VOC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 3 0 NOM S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 3 0 GEN S X PRES ACTIVE  PPL 1 5 entis       X A
VPAR 3 0 DAT S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 3 0 ACC S C PRES ACTIVE  PPL 1 5 entem       X A
VPAR 3 0 ABL S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 3 0 ABL S X PRES ACTIVE  PPL 1 4 ente        X A
VPAR 3 0 VOC S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 3 0 NOM P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 3 0 GEN P X PRES ACTIVE  PPL 1 6 entium      X A
VPAR 3 0 GEN P X PRES ACTIVE  PPL 1 5 entum       X C
VPAR 3 0 DAT P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 3 0 ACC P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 3 0 ABL P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 3 0 VOC P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 3 0 ACC S N PRES ACTIVE  PPL 1 3 ens         X A
VPAR 3 0 NOM P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 3 0 ACC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 3 0 VOC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 5 1 NOM S X PRES ACTIVE  PPL 2 3 ens         X A
VPAR 5 1 GEN S X PRES ACTIVE  PPL 2 5 entis       X A
VPAR 5 1 DAT S X PRES ACTIVE  PPL 2 4 enti        X A
VPAR 5 1 ACC S C PRES ACTIVE  PPL 2 5 entem       X A
VPAR 5 1 ABL S X PRES ACTIVE  PPL 2 4 enti        X A
VPAR 5 1 ABL S X PRES ACTIVE  PPL 2 4 ente        X A
VPAR 5 1 VOC S X PRES ACTIVE  PPL 2 3 ens         X A
VPAR 5 1 NOM P C PRES ACTIVE  PPL 2 5 entes       X A
VPAR 5 1 GEN P X PRES ACTIVE  PPL 2 6 entium      X A
VPAR 5 1 DAT P X PRES ACTIVE  PPL 2 7 entibus     X A
```

```
VPAR 5 1 ACC P C PRES ACTIVE  PPL 2 5 entes       X A
VPAR 5 1 ABL P X PRES ACTIVE  PPL 2 7 entibus     X A
VPAR 5 1 VOC P C PRES ACTIVE  PPL 2 5 entes       X A
VPAR 5 1 ACC S N PRES ACTIVE  PPL 2 3 ens         X A
VPAR 5 1 NOM P N PRES ACTIVE  PPL 2 5 entia       X A
VPAR 5 1 ACC P N PRES ACTIVE  PPL 2 5 entia       X A
VPAR 5 1 VOC P N PRES ACTIVE  PPL 2 5 entia       X A
VPAR 6 1 NOM S X PRES ACTIVE  PPL 2 3 ens         X A
VPAR 6 1 GEN S X PRES ACTIVE  PPL 1 5 untis       X A
VPAR 6 1 DAT S X PRES ACTIVE  PPL 1 4 unti        X A
VPAR 6 1 ACC S C PRES ACTIVE  PPL 1 5 untem       X A
VPAR 6 1 ABL S X PRES ACTIVE  PPL 1 4 unti        X A
VPAR 6 1 ABL S X PRES ACTIVE  PPL 1 4 unte        X A
VPAR 6 1 VOC S X PRES ACTIVE  PPL 2 3 ens         X A
VPAR 6 1 NOM P C PRES ACTIVE  PPL 1 5 untes       X A
VPAR 6 1 GEN P X PRES ACTIVE  PPL 1 6 untium      X A
VPAR 6 1 DAT P X PRES ACTIVE  PPL 1 7 untibus     X A
VPAR 6 1 ACC P C PRES ACTIVE  PPL 1 5 untes       X A
VPAR 6 1 ABL P X PRES ACTIVE  PPL 1 7 untibus     X A
VPAR 6 1 VOC P C PRES ACTIVE  PPL 1 5 untes       X A
VPAR 6 1 ACC S N PRES ACTIVE  PPL 2 3 ens         X A
VPAR 6 1 NOM P N PRES ACTIVE  PPL 1 5 untia       X A
VPAR 6 1 ACC P N PRES ACTIVE  PPL 1 5 untia       X A
VPAR 6 1 VOC P N PRES ACTIVE  PPL 1 5 untia       X A
VPAR 6 1 GEN S X PRES ACTIVE  PPL 2 5 entis       E D
VPAR 6 1 DAT S X PRES ACTIVE  PPL 2 4 enti        E D
VPAR 6 1 ACC S C PRES ACTIVE  PPL 2 5 entem       E D
VPAR 6 1 ABL S X PRES ACTIVE  PPL 2 4 enti        E D
VPAR 6 1 ABL S X PRES ACTIVE  PPL 2 4 ente        E D
VPAR 6 1 NOM P C PRES ACTIVE  PPL 2 5 entes       E D
VPAR 6 1 GEN P X PRES ACTIVE  PPL 2 6 entium      E D
VPAR 6 1 DAT P X PRES ACTIVE  PPL 2 7 entibus     E D
VPAR 6 1 ACC P C PRES ACTIVE  PPL 2 5 entes       E D
VPAR 6 1 ABL P X PRES ACTIVE  PPL 2 7 entibus     E D
VPAR 6 1 VOC P C PRES ACTIVE  PPL 2 5 entes       E D
VPAR 6 2 NOM P N PRES ACTIVE  PPL 1 5 entia       E D
VPAR 6 2 ACC P N PRES ACTIVE  PPL 1 5 entia       E D
VPAR 6 2 VOC P N PRES ACTIVE  PPL 1 5 entia       E D
VPAR 6 2 NOM S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 6 2 GEN S X PRES ACTIVE  PPL 1 5 entis       X A
VPAR 6 2 DAT S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 6 2 ACC S C PRES ACTIVE  PPL 1 5 entem       X A
VPAR 6 2 ABL S X PRES ACTIVE  PPL 1 4 enti        X A
VPAR 6 2 ABL S X PRES ACTIVE  PPL 1 4 ente        X A
VPAR 6 2 VOC S X PRES ACTIVE  PPL 1 3 ens         X A
VPAR 6 2 NOM P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 6 2 GEN P X PRES ACTIVE  PPL 1 6 entium      X A
VPAR 6 2 DAT P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 6 2 ACC P C PRES ACTIVE  PPL 1 5 entes       X A
VPAR 6 2 ABL P X PRES ACTIVE  PPL 1 7 entibus     X A
VPAR 6 2 VOC P C PRES ACTIVE  PPL 1 5 entes       X A
```

```
VPAR 6 2 ACC S N PRES ACTIVE  PPL 1 3 ens         X A
VPAR 6 2 NOM P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 6 2 ACC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 6 2 VOC P N PRES ACTIVE  PPL 1 5 entia       X A
VPAR 7 2 NOM S X PRES ACTIVE  PPL 1 3 ens         E A
VPAR 7 2 NOM P C PRES ACTIVE  PPL 1 5 entes       E A
VPAR 0 0 NOM S M PERF PASSIVE PPL 4 2 us          X A
VPAR 0 0 GEN S M PERF PASSIVE PPL 4 1 i           X A
VPAR 0 0 DAT S M PERF PASSIVE PPL 4 1 o           X A
VPAR 0 0 ACC S M PERF PASSIVE PPL 4 2 um          X A
VPAR 0 0 ABL S M PERF PASSIVE PPL 4 1 o           X A
VPAR 0 0 VOC S M PERF PASSIVE PPL 4 1 e           X A
VPAR 0 0 NOM P M PERF PASSIVE PPL 4 1 i           X A
VPAR 0 0 GEN P M PERF PASSIVE PPL 4 4 orum        X A
VPAR 0 0 DAT P X PERF PASSIVE PPL 4 2 is          X A
VPAR 0 0 ACC P M PERF PASSIVE PPL 4 2 os          X A
VPAR 0 0 ABL P X PERF PASSIVE PPL 4 2 is          X A
VPAR 0 0 VOC P M PERF PASSIVE PPL 4 1 i           X A
VPAR 0 0 NOM S F PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 GEN S F PERF PASSIVE PPL 4 2 ae          X A
VPAR 0 0 DAT S F PERF PASSIVE PPL 4 2 ae          X A
VPAR 0 0 ACC S F PERF PASSIVE PPL 4 2 am          X A
VPAR 0 0 ABL S F PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 VOC S F PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 NOM P F PERF PASSIVE PPL 4 2 ae          X A
VPAR 0 0 GEN P F PERF PASSIVE PPL 4 4 arum        X A
VPAR 0 0 ACC P F PERF PASSIVE PPL 4 2 as          X A
VPAR 0 0 VOC P F PERF PASSIVE PPL 4 2 ae          X A
VPAR 0 0 NOM S N PERF PASSIVE PPL 4 2 um          X A
VPAR 0 0 GEN S N PERF PASSIVE PPL 4 1 i           X A
VPAR 0 0 DAT S N PERF PASSIVE PPL 4 1 o           X A
VPAR 0 0 ACC S N PERF PASSIVE PPL 4 2 um          X A
VPAR 0 0 ABL S N PERF PASSIVE PPL 4 1 o           X A
VPAR 0 0 VOC S N PERF PASSIVE PPL 4 2 um          X A
VPAR 0 0 NOM P N PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 GEN P N PERF PASSIVE PPL 4 4 orum        X A
VPAR 0 0 ACC P N PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 VOC P N PERF PASSIVE PPL 4 1 a           X A
VPAR 0 0 NOM S M FUT  ACTIVE  PPL 4 4 urus        X A
VPAR 0 0 GEN S M FUT  ACTIVE  PPL 4 3 uri         X A
VPAR 0 0 DAT S M FUT  ACTIVE  PPL 4 3 uro         X A
VPAR 0 0 ACC S M FUT  ACTIVE  PPL 4 4 urum        X A
VPAR 0 0 ABL S M FUT  ACTIVE  PPL 4 3 uro         X A
VPAR 0 0 VOC S M FUT  ACTIVE  PPL 4 3 ure         X A
VPAR 0 0 NOM P M FUT  ACTIVE  PPL 4 3 uri         X A
VPAR 0 0 GEN P M FUT  ACTIVE  PPL 4 6 urorum      X A
VPAR 0 0 DAT P X FUT  ACTIVE  PPL 4 4 uris        X A
VPAR 0 0 ACC P M FUT  ACTIVE  PPL 4 4 uros        X A
VPAR 0 0 ABL P X FUT  ACTIVE  PPL 4 4 uris        X A
VPAR 0 0 VOC P M FUT  ACTIVE  PPL 4 3 uri         X A
VPAR 0 0 NOM S F FUT  ACTIVE  PPL 4 3 ura         X A
```

```
VPAR 0 0 GEN S F FUT  ACTIVE  PPL 4 4 urae         X A
VPAR 0 0 DAT S F FUT  ACTIVE  PPL 4 4 urae         X A
VPAR 0 0 ACC S F FUT  ACTIVE  PPL 4 4 uram         X A
VPAR 0 0 ABL S F FUT  ACTIVE  PPL 4 3 ura          X A
VPAR 0 0 VOC S F FUT  ACTIVE  PPL 4 3 ura          X A
VPAR 0 0 NOM P F FUT  ACTIVE  PPL 4 4 urae         X A
VPAR 0 0 GEN P F FUT  ACTIVE  PPL 4 6 urarum       X A
VPAR 0 0 ACC P F FUT  ACTIVE  PPL 4 4 uras         X A
VPAR 0 0 VOC P F FUT  ACTIVE  PPL 4 4 urae         X A
VPAR 0 0 NOM S N FUT  ACTIVE  PPL 4 4 urum         X A
VPAR 0 0 GEN S N FUT  ACTIVE  PPL 4 3 uri          X A
VPAR 0 0 DAT S N FUT  ACTIVE  PPL 4 3 uro          X A
VPAR 0 0 ACC S N FUT  ACTIVE  PPL 4 4 urum         X A
VPAR 0 0 ABL S N FUT  ACTIVE  PPL 4 3 uro          X A
VPAR 0 0 VOC S N FUT  ACTIVE  PPL 4 4 urum         X A
VPAR 0 0 NOM P N FUT  ACTIVE  PPL 4 3 ura          X A
VPAR 0 0 GEN P N FUT  ACTIVE  PPL 4 6 urorum       X A
VPAR 0 0 ACC P N FUT  ACTIVE  PPL 4 3 ura          X A
VPAR 0 0 VOC P N FUT  ACTIVE  PPL 4 3 ura          X A
VPAR 1 0 NOM S M FUT  PASSIVE PPL 1 5 andus        X A
VPAR 1 0 GEN S M FUT  PASSIVE PPL 1 4 andi         X A
VPAR 1 0 DAT S M FUT  PASSIVE PPL 1 4 ando         X A
VPAR 1 0 ACC S M FUT  PASSIVE PPL 1 5 andum        X A
VPAR 1 0 ABL S M FUT  PASSIVE PPL 1 4 ando         X A
VPAR 1 0 VOC S M FUT  PASSIVE PPL 1 4 ande         X A
VPAR 1 0 NOM P M FUT  PASSIVE PPL 1 4 andi         X A
VPAR 1 0 GEN P M FUT  PASSIVE PPL 1 7 andorum      X A
VPAR 1 0 DAT P X FUT  PASSIVE PPL 1 5 andis        X A
VPAR 1 0 ACC P M FUT  PASSIVE PPL 1 5 andos        X A
VPAR 1 0 ABL P X FUT  PASSIVE PPL 1 5 andis        X A
VPAR 1 0 VOC P M FUT  PASSIVE PPL 1 4 andi         X A
VPAR 1 0 NOM S F FUT  PASSIVE PPL 1 4 anda         X A
VPAR 1 0 GEN S F FUT  PASSIVE PPL 1 5 andae        X A
VPAR 1 0 DAT S F FUT  PASSIVE PPL 1 5 andae        X A
VPAR 1 0 ACC S F FUT  PASSIVE PPL 1 5 andam        X A
VPAR 1 0 ABL S F FUT  PASSIVE PPL 1 4 anda         X A
VPAR 1 0 VOC S F FUT  PASSIVE PPL 1 4 anda         X A
VPAR 1 0 NOM P F FUT  PASSIVE PPL 1 5 andae        X A
VPAR 1 0 GEN P F FUT  PASSIVE PPL 1 7 andarum      X A
VPAR 1 0 ACC P F FUT  PASSIVE PPL 1 5 andas        X A
VPAR 1 0 VOC P F FUT  PASSIVE PPL 1 5 andae        X A
VPAR 1 0 NOM S N FUT  PASSIVE PPL 1 5 andum        X A
VPAR 1 0 GEN S N FUT  PASSIVE PPL 1 4 andi         X A
VPAR 1 0 DAT S N FUT  PASSIVE PPL 1 4 ando         X A
VPAR 1 0 ACC S N FUT  PASSIVE PPL 1 5 andum        X A
VPAR 1 0 ABL S N FUT  PASSIVE PPL 1 4 ando         X A
VPAR 1 0 VOC S N FUT  PASSIVE PPL 1 5 andum        X A
VPAR 1 0 NOM P N FUT  PASSIVE PPL 1 4 anda         X A
VPAR 1 0 GEN P N FUT  PASSIVE PPL 1 7 andorum      X A
VPAR 1 0 ACC P N FUT  PASSIVE PPL 1 4 anda         X A
VPAR 1 0 VOC P N FUT  PASSIVE PPL 1 4 anda         X A
```

```
VPAR 2 0 NOM S M FUT  PASSIVE PPL 1 5 endus        X A
VPAR 2 0 GEN S M FUT  PASSIVE PPL 1 4 endi         X A
VPAR 2 0 DAT S M FUT  PASSIVE PPL 1 4 endo         X A
VPAR 2 0 ACC S M FUT  PASSIVE PPL 1 5 endum        X A
VPAR 2 0 ABL S M FUT  PASSIVE PPL 1 4 endo         X A
VPAR 2 0 VOC S M FUT  PASSIVE PPL 1 4 ende         X A
VPAR 2 0 NOM P M FUT  PASSIVE PPL 1 4 endi         X A
VPAR 2 0 GEN P M FUT  PASSIVE PPL 1 7 endorum      X A
VPAR 2 0 DAT P X FUT  PASSIVE PPL 1 5 endis        X A
VPAR 2 0 ACC P M FUT  PASSIVE PPL 1 5 endos        X A
VPAR 2 0 ABL P X FUT  PASSIVE PPL 1 5 endis        X A
VPAR 2 0 VOC P M FUT  PASSIVE PPL 1 4 endi         X A
VPAR 2 0 NOM S F FUT  PASSIVE PPL 1 4 enda         X A
VPAR 2 0 GEN S F FUT  PASSIVE PPL 1 5 endae        X A
VPAR 2 0 DAT S F FUT  PASSIVE PPL 1 5 endae        X A
VPAR 2 0 ACC S F FUT  PASSIVE PPL 1 5 endam        X A
VPAR 2 0 ABL S F FUT  PASSIVE PPL 1 4 enda         X A
VPAR 2 0 VOC S F FUT  PASSIVE PPL 1 4 enda         X A
VPAR 2 0 NOM P F FUT  PASSIVE PPL 1 5 endae        X A
VPAR 2 0 GEN P F FUT  PASSIVE PPL 1 7 endarum      X A
VPAR 2 0 ACC P F FUT  PASSIVE PPL 1 5 endas        X A
VPAR 2 0 VOC P F FUT  PASSIVE PPL 1 5 endae        X A
VPAR 2 0 NOM S N FUT  PASSIVE PPL 1 5 endum        X A
VPAR 2 0 GEN S N FUT  PASSIVE PPL 1 4 endi         X A
VPAR 2 0 DAT S N FUT  PASSIVE PPL 1 4 endo         X A
VPAR 2 0 ACC S N FUT  PASSIVE PPL 1 5 endum        X A
VPAR 2 0 ABL S N FUT  PASSIVE PPL 1 4 endo         X A
VPAR 2 0 VOC S N FUT  PASSIVE PPL 1 5 endum        X A
VPAR 2 0 NOM P N FUT  PASSIVE PPL 1 4 enda         X A
VPAR 2 0 GEN P N FUT  PASSIVE PPL 1 7 endorum      X A
VPAR 2 0 ACC P N FUT  PASSIVE PPL 1 4 enda         X A
VPAR 2 0 VOC P N FUT  PASSIVE PPL 1 4 enda         X A
VPAR 3 0 NOM S M FUT  PASSIVE PPL 1 5 endus        D A
VPAR 3 0 GEN S M FUT  PASSIVE PPL 1 4 endi         D A
VPAR 3 0 DAT S M FUT  PASSIVE PPL 1 4 endo         D A
VPAR 3 0 ACC S M FUT  PASSIVE PPL 1 5 endum        D A
VPAR 3 0 ABL S M FUT  PASSIVE PPL 1 4 endo         D A
VPAR 3 0 VOC S M FUT  PASSIVE PPL 1 4 ende         D A
VPAR 3 0 NOM P M FUT  PASSIVE PPL 1 4 endi         D A
VPAR 3 0 GEN P M FUT  PASSIVE PPL 1 7 endorum      D A
VPAR 3 0 DAT P X FUT  PASSIVE PPL 1 5 endis        D A
VPAR 3 0 ACC P M FUT  PASSIVE PPL 1 5 endos        D A
VPAR 3 0 ABL P X FUT  PASSIVE PPL 1 5 endis        D A
VPAR 3 0 VOC P M FUT  PASSIVE PPL 1 4 endi         D A
VPAR 3 0 NOM S F FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 GEN S F FUT  PASSIVE PPL 1 5 endae        D A
VPAR 3 0 DAT S F FUT  PASSIVE PPL 1 5 endae        D A
VPAR 3 0 ACC S F FUT  PASSIVE PPL 1 5 endam        D A
VPAR 3 0 ABL S F FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 VOC S F FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 NOM P F FUT  PASSIVE PPL 1 5 endae        D A
```

```
VPAR 3 0 GEN P F FUT  PASSIVE PPL 1 7 endarum      D A
VPAR 3 0 ACC P F FUT  PASSIVE PPL 1 5 endas        D A
VPAR 3 0 VOC P F FUT  PASSIVE PPL 1 5 endae        D A
VPAR 3 0 NOM S N FUT  PASSIVE PPL 1 5 endum        D A
VPAR 3 0 GEN S N FUT  PASSIVE PPL 1 4 endi         D A
VPAR 3 0 DAT S N FUT  PASSIVE PPL 1 4 endo         D A
VPAR 3 0 ACC S N FUT  PASSIVE PPL 1 5 endum        D A
VPAR 3 0 ABL S N FUT  PASSIVE PPL 1 4 endo         D A
VPAR 3 0 VOC S N FUT  PASSIVE PPL 1 5 endum        D A
VPAR 3 0 NOM P N FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 GEN P N FUT  PASSIVE PPL 1 7 endorum      D A
VPAR 3 0 ACC P N FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 VOC P N FUT  PASSIVE PPL 1 4 enda         D A
VPAR 3 0 NOM S M FUT  PASSIVE PPL 1 5 undus        B A
VPAR 3 0 GEN S M FUT  PASSIVE PPL 1 4 undi         B A
VPAR 3 0 DAT S M FUT  PASSIVE PPL 1 4 undo         B A
VPAR 3 0 ACC S M FUT  PASSIVE PPL 1 5 undum        B A
VPAR 3 0 ABL S M FUT  PASSIVE PPL 1 4 undo         B A
VPAR 3 0 VOC S M FUT  PASSIVE PPL 1 4 unde         B A
VPAR 3 0 NOM P M FUT  PASSIVE PPL 1 4 undi         B A
VPAR 3 0 GEN P M FUT  PASSIVE PPL 1 7 undorum      B A
VPAR 3 0 DAT P X FUT  PASSIVE PPL 1 5 undis        B A
VPAR 3 0 ACC P M FUT  PASSIVE PPL 1 5 undos        B A
VPAR 3 0 ABL P X FUT  PASSIVE PPL 1 5 undis        B A
VPAR 3 0 VOC P M FUT  PASSIVE PPL 1 4 undi         B A
VPAR 3 0 NOM S F FUT  PASSIVE PPL 1 4 unda         B A
VPAR 3 0 GEN S F FUT  PASSIVE PPL 1 5 undae        B A
VPAR 3 0 DAT S F FUT  PASSIVE PPL 1 5 undae        B A
VPAR 3 0 ACC S F FUT  PASSIVE PPL 1 5 undam        B A
VPAR 3 0 ABL S F FUT  PASSIVE PPL 1 4 unda         B A
VPAR 3 0 VOC S F FUT  PASSIVE PPL 1 4 unda         B A
VPAR 3 0 NOM P F FUT  PASSIVE PPL 1 5 undae        B A
VPAR 3 0 GEN P F FUT  PASSIVE PPL 1 7 undarum      B A
VPAR 3 0 ACC P F FUT  PASSIVE PPL 1 5 undas        B A
VPAR 3 0 VOC P F FUT  PASSIVE PPL 1 5 undae        B A
VPAR 3 0 NOM S N FUT  PASSIVE PPL 1 5 undum        B A
VPAR 3 0 GEN S N FUT  PASSIVE PPL 1 4 undi         B A
VPAR 3 0 DAT S N FUT  PASSIVE PPL 1 4 undo         B A
VPAR 3 0 ACC S N FUT  PASSIVE PPL 1 5 undum        B A
VPAR 3 0 ABL S N FUT  PASSIVE PPL 1 4 undo         B A
VPAR 3 0 VOC S N FUT  PASSIVE PPL 1 5 undum        B A
VPAR 3 0 NOM P N FUT  PASSIVE PPL 1 4 unda         B A
VPAR 3 0 GEN P N FUT  PASSIVE PPL 1 7 undorum      B A
VPAR 3 0 ACC P N FUT  PASSIVE PPL 1 4 unda         B A
VPAR 3 0 VOC P N FUT  PASSIVE PPL 1 4 unda         B A
VPAR 6 1 NOM S M FUT  PASSIVE PPL 1 5 undus        X A
VPAR 6 1 GEN S M FUT  PASSIVE PPL 1 4 undi         X A
VPAR 6 1 DAT S M FUT  PASSIVE PPL 1 4 undo         X A
VPAR 6 1 ACC S M FUT  PASSIVE PPL 1 5 undum        X A
VPAR 6 1 ABL S M FUT  PASSIVE PPL 1 4 undo         X A
VPAR 6 1 VOC S M FUT  PASSIVE PPL 1 4 unde         X A
```

```
VPAR 6 1 NOM P M FUT  PASSIVE PPL 1 4 undi         X A
VPAR 6 1 GEN P M FUT  PASSIVE PPL 1 7 undorum      X A
VPAR 6 1 DAT P X FUT  PASSIVE PPL 1 5 undis        X A
VPAR 6 1 ACC P M FUT  PASSIVE PPL 1 5 undos        X A
VPAR 6 1 ABL P X FUT  PASSIVE PPL 1 5 undis        X A
VPAR 6 1 VOC P M FUT  PASSIVE PPL 1 4 undi         X A
VPAR 6 1 NOM S F FUT  PASSIVE PPL 1 4 unda         X A
VPAR 6 1 GEN S F FUT  PASSIVE PPL 1 5 undae        X A
VPAR 6 1 DAT S F FUT  PASSIVE PPL 1 5 undae        X A
VPAR 6 1 ACC S F FUT  PASSIVE PPL 1 5 undam        X A
VPAR 6 1 ABL S F FUT  PASSIVE PPL 1 4 unda         X A
VPAR 6 1 VOC S F FUT  PASSIVE PPL 1 4 unda         X A
VPAR 6 1 NOM P F FUT  PASSIVE PPL 1 5 undae        X A
VPAR 6 1 GEN P F FUT  PASSIVE PPL 1 7 undarum      X A
VPAR 6 1 ACC P F FUT  PASSIVE PPL 1 5 undas        X A
VPAR 6 1 VOC P F FUT  PASSIVE PPL 1 5 undae        X A
VPAR 6 1 NOM S N FUT  PASSIVE PPL 1 5 undum        X A
VPAR 6 1 GEN S N FUT  PASSIVE PPL 1 4 undi         X A
VPAR 6 1 DAT S N FUT  PASSIVE PPL 1 4 undo         X A
VPAR 6 1 ACC S N FUT  PASSIVE PPL 1 5 undum        X A
VPAR 6 1 ABL S N FUT  PASSIVE PPL 1 4 undo         X A
VPAR 6 1 VOC S N FUT  PASSIVE PPL 1 5 undum        X A
VPAR 6 1 NOM P N FUT  PASSIVE PPL 1 4 unda         X A
VPAR 6 1 GEN P N FUT  PASSIVE PPL 1 7 undorum      X A
VPAR 6 1 ACC P N FUT  PASSIVE PPL 1 4 unda         X A
VPAR 6 1 VOC P N FUT  PASSIVE PPL 1 4 unda         X A
--  Supino
SUPINE 0 0 ACC S N  4 2 um                          X A
SUPINE 0 0 ABL S N  4 1 u                           X A
--  Pronomi
--      1 1   qui         in NOM S M
--      1 2   quis
--
--      1 3         qua   in NOM S F
--      1 4         quae
--      1 5         quis
--
--      1 6               id  in NON/ACC S N
--      1 7               od
--
--      1 8               qua   in NOM P N
--      1 9               quae
--
--
--aliquis, aliqua (rare), aliquid (aliquod rare/late) (aliqua P N)    INDEF
--‘‘
--aliqui, qua, quod  (quid rare/late)  (qua P N)    ADJECT
--
--quis, qua/quae, quid (qua/quae P N)            INDEF
--
--qui, qua (quae), quod (qua /-quae P N)   ADJECT
```

```
--
--qui, quae, quod (quae P N)          REL
--
--quis, quis, quid (quae P N)          INTERR
--
--qui, quae, quod (quae P N)          ADJECT
--
PRON  1 0 GEN S X   2 3 jus                       X A
PRON  1 0 DAT S X   2 1 i                         X A
PRON  1 0 ACC S M   1 2 em                        X A
PRON  1 0 ABL S M   1 1 o                         X A
PRON  1 0 ACC S F   1 2 am                        X A
PRON  1 0 ABL S N   1 1 o                         X A
PRON  1 0 NOM P M   1 1 i                         X A
PRON  1 0 GEN P M   1 4 orum                      X A
PRON  1 0 DAT P X   1 4 ibus                      X A
PRON  1 0 DAT P X   1 2 is                        X A
PRON  1 0 ACC P M   1 2 os                        X A
PRON  1 0 ABL P X   1 4 ibus                      X A
PRON  1 0 ABL P X   1 2 is                        X A
PRON  1 0 NOM P F   1 2 ae                        X A
PRON  1 0 GEN P F   1 4 arum                      X A
PRON  1 0 ACC P F   1 2 as                        X A
PRON  1 0 GEN P N   1 4 orum                      X A
PRON  1 1 NOM S M   1 1 i                         X A
PRON  1 2 NOM S C   1 2 is                        X A
PRON  1 5 ABL S F   1 1 o                         X A
PRON  1 3 NOM S F   1 1 a                         X A
PRON  1 3 ABL S F   1 1 a                         X A
--  quae
PRON  1 4 NOM S F   1 2 ae                        X A
PRON  1 4 ABL S F   1 1 a                         X A
--  quis
--PRON  1 5 NOM S F   1 2 is                        X A
--PRON  1 5 ABL S F   1 1 o                         X A
--  -id or -od NOM/ACC N S;
--  id
PRON  1 6 NOM S N   1 2 id                        X A
PRON  1 6 ACC S N   1 2 id                        X A
--  od
PRON  1 7 NOM S N   1 2 od                        X A
PRON  1 7 ACC S N   1 2 od                        X A
--  -ae or -a NOM/ACC N P
--  -ae
PRON  1 8 NOM P N   1 2 ae                        X A
PRON  1 8 ACC P N   1 2 ae                        X A
--  -a
PRON  1 9 NOM P N   1 1 a                         X A
PRON  1 9 ACC P N   1 1 a                         X A
--  Ex: hic (huius) haec hoc  => h hu
PRON  3 0 NOM S M   1 2 ic                        X A
```

```
PRON  3 0 GEN S X   2 3 ius                       X A
PRON  3 0 DAT S X   2 2 ic                        X A
PRON  3 0 ACC S M   1 3 unc                       X A
PRON  3 0 ABL S M   1 2 oc                        X A
PRON  3 0 NOM P M   1 1 i                         X A
PRON  3 0 NOM P M   1 2 ii                        E C
PRON  3 0 GEN P M   1 4 orum                      X A
PRON  3 0 DAT P X   1 2 is                        X A
PRON  3 0 DAT P X   1 3 iis                       E C
PRON  3 0 ACC P M   1 2 os                        X A
PRON  3 0 ABL P X   1 2 is                        X A
PRON  3 0 ABL P X   1 3 iis                       E C
PRON  3 0 NOM S F   1 3 aec                       X A
PRON  3 0 ACC S F   1 3 anc                       X A
PRON  3 0 ABL S F   1 2 ac                        X A
PRON  3 0 NOM P F   1 2 ae                        X A
PRON  3 0 GEN P F   1 4 arum                      X A
PRON  3 0 ACC P F   1 2 as                        X A
PRON  3 1 NOM S N   1 2 oc                        X A
PRON  3 1 ACC S N   1 2 oc                        X A
PRON  3 0 ABL S N   1 2 oc                        X A
PRON  3 0 NOM P N   1 3 aec                       X A
PRON  3 0 GEN P N   1 4 orum                      X A
PRON  3 0 ACC P N   1 3 aec                       X A
--  Ex: istic (istuius) istaec istuc  => ist istu
PRON  3 2 NOM S N   1 2 uc                        X A
PRON  3 2 ACC S N   1 2 uc                        X A
--  Ex: is ea id  =>  i e i
PRON  4 1 NOM S M   1 1 s                         X A
PRON  4 0 GEN S X   2 3 ius                       X A
PRON  4 0 DAT S X   2 1 i                         X A
PRON  4 1 ACC S M   2 2 um                        X A
PRON  4 0 ABL S M   2 1 o                         X A
PRON  4 0 NOM P M   2 1 i                         X A
PRON  4 0 NOM P M   1 1 i                         X B
PRON  4 1 GEN P M   2 4 orum                      X A
PRON  4 0 DAT P X   2 2 is                        X A
PRON  4 0 DAT P X   1 2 is                        X A
PRON  4 0 DAT P X   1 1 s                         X C
PRON  4 0 ACC P M   2 2 os                        X A
PRON  4 0 ABL P X   2 2 is                        X A
PRON  4 0 ABL P X   1 2 is                        X A
PRON  4 0 ABL P X   1 1 s                         X C
PRON  4 0 NOM S F   2 1 a                         X A
PRON  4 1 ACC S F   2 2 am                        X A
PRON  4 0 ABL S F   2 1 a                         X A
PRON  4 0 NOM P F   2 2 ae                        X A
PRON  4 1 GEN P F   2 4 arum                      X A
PRON  4 0 ACC P F   2 2 as                        X A
PRON  4 1 NOM S N   1 1 d                         X A
PRON  4 1 ACC S N   1 1 d                         X A
```

```
PRON  4 0 ABL S N   1 1 o                          X A
PRON  4 0 NOM P N   2 1 a                          X A
PRON  4 1 GEN P N   2 4 orum                       X A
PRON  4 0 ACC P N   2 1 a                          X A
--  Ex: idem eadem idem  =>  i e i  -dem
PRON  4 2 NOM S M   1 0                            X A
PRON  4 2 ACC S M   2 2 un                         X A
PRON  4 2 GEN P M   2 4 orun                       X A
PRON  4 2 ACC S F   2 2 an                         X A
PRON  4 2 GEN P F   2 4 arun                       X A
PRON  4 2 NOM S N   1 0                            X A
PRON  4 2 ACC S N   1 0                            X A
PRON  4 2 NOM P N   2 1 a                          X A
PRON  4 2 GEN P N   2 4 orun                       X A
PRON  4 2 ACC P N   2 1 a                          X A
--  Ex: ego mei    =>       ego   m
--  Ex:     mei    =>       zzz   m
PRON  5 1 NOM S C   1 0                            X A
PRON  5 1 GEN S C   2 2 ei                         X A
PRON  5 1 GEN S C   2 2 is                         B C
PRON  5 1 DAT S C   2 3 ihi                        X A
PRON  5 1 DAT S C   2 1 i                          X C
PRON  5 1 ACC S C   2 1 e                          X A
PRON  5 1 ABL S C   2 1 e                          X A
PRON  5 1 ACC S C   2 2 ed                         A E
PRON  5 1 ABL S C   2 2 ed                         A E
PRON  5 1 ACC S C   2 3 eme                        X D
PRON  5 1 ABL S C   2 3 eme                        X D
--  Ex: tu tui    =>        tu   t
--  Ex: tui       =>        zzz  t      reflexive
PRON  5 2 NOM S C   1 0                            X A
PRON  5 2 GEN S C   2 2 ui                         X A
PRON  5 2 DAT S C   2 3 ibi                        X A
PRON  5 2 ACC S C   2 1 e                          X A
PRON  5 2 ABL S C   2 1 e                          X A
--  Ex: nos nostrum       =>  n nostr
--  Ex: vos vostrum       =>  v vostr
PRON  5 3 NOM P C   1 2 os                         X A
PRON  5 3 GEN P C   2 2 um                         X A
PRON  5 3 GEN P C   2 1 i                          X A
PRON  5 3 DAT P C   1 4 obis                       X A
PRON  5 3 ACC P C   1 2 os                         X A
PRON  5 3 ABL P C   1 4 obis                       X A
--  Ex: sui       =>        zzz  s      reflexive
PRON  5 4 GEN X C   2 2 ui                         X A
PRON  5 4 DAT X C   2 3 ibi                        X A
PRON  5 4 ACC X C   2 1 e                          X A
PRON  5 4 ACC X C   2 3 ese                        X A
PRON  5 4 ABL X C   2 1 e                          X A
PRON  5 4 ABL X C   2 3 ese                        X A
--  Ex: ille illa illud  => ill
```

```
--  Ex: iste ista istud  =>  ist
PRON  6 0 NOM S M   1 1 e                       X A
PRON  6 0 GEN S X   1 3 ius                     X A
PRON  6 0 DAT S X   1 1 i                       X A
PRON  6 0 ACC S M   1 2 um                      X A
PRON  6 0 ABL S M   1 1 o                       X A
PRON  6 0 NOM P M   1 1 i                       X A
PRON  6 0 GEN P M   1 4 orum                    X A
PRON  6 0 DAT P X   1 2 is                      X A
PRON  6 0 ACC P M   1 2 os                      X A
PRON  6 0 ABL P X   1 2 is                      X A
PRON  6 0 NOM S F   1 1 a                       X A
PRON  6 0 ACC S F   1 2 am                      X A
PRON  6 0 NOM P F   1 2 ae                      X A
PRON  6 0 GEN P F   1 4 arum                    X A
PRON  6 0 ACC P F   1 2 as                      X A
PRON  6 1 NOM S N   1 2 ud                      X A
PRON  6 1 ACC S N   1 2 ud                      X A
PRON  6 0 NOM P N   1 1 a                       X A
PRON  6 0 GEN P N   1 4 orum                    X A
PRON  6 0 ACC P N   1 1 a                       X A
--  Ex: ipse ipsa ipsum  =>  ips
PRON  6 2 NOM S M   1 2 us                      B C
PRON  6 2 NOM S M   1 2 os                      B E
PRON  6 2 NOM S N   1 2 um                      X A
PRON  6 2 ACC S N   1 2 um                      X A
PRON  6 2 NOM S N   1 2 ud                      E E
PRON  6 2 ACC S N   1 2 ud                      E E
--  Numerali
--  Cardinali
NUM    1 1 NOM S M CARD     1 2 us              X A
NUM    1 1 GEN S X CARD     1 3 ius             X A
NUM    1 1 DAT S X CARD     1 1 i               X A
NUM    1 1 ACC S M CARD     1 2 um              X A
NUM    1 1 ABL S M CARD     1 1 o               X A
NUM    1 1 VOC S M CARD     1 1 e               X D
NUM    1 1 NOM S F CARD     1 1 a               X A
NUM    1 1 ACC S F CARD     1 2 am              X A
NUM    1 1 ABL S F CARD     1 1 a               X A
NUM    1 1 VOC S F CARD     1 1 a               X A
NUM    1 1 NOM S N CARD     1 2 um              X A
NUM    1 1 ACC S N CARD     1 2 um              X A
NUM    1 1 ABL S N CARD     1 1 o               X A
NUM    1 1 VOC S N CARD     1 2 um              X A
NUM    1 2 NOM P M CARD     1 1 o               X A
NUM    1 2 GEN P M CARD     1 4 orum            X A
NUM    1 2 GEN P M CARD     1 2 um              X B
NUM    1 2 GEN P M CARD     1 2 om              B C
NUM    1 2 DAT P M CARD     1 4 obus            X A
NUM    1 2 ACC P M CARD     1 2 os              X A
NUM    1 2 ACC P M CARD     1 1 o               B B
```

```
NUM    1 2 ABL P M CARD     1 4 obus              X A
NUM    1 2 VOC P M CARD     1 1 o                 X A
NUM    1 2 NOM P F CARD     1 2 ae                X A
NUM    1 2 NOM P F CARD     1 1 o                 X D
NUM    1 2 NOM P F CARD     1 1 a                 X E
NUM    1 2 GEN P F CARD     1 4 arum              X A
NUM    1 2 DAT P F CARD     1 4 abus              X A
NUM    1 2 ACC P F CARD     1 2 as                X A
NUM    1 2 ABL P F CARD     1 4 abus              X A
NUM    1 2 VOC P F CARD     1 2 ae                X A
NUM    1 2 NOM P N CARD     1 1 o                 X A
NUM    1 2 GEN P N CARD     1 4 orum              X A
NUM    1 2 GEN P N CARD     1 2 um                X B
NUM    1 2 GEN P N CARD     1 2 om                B C
NUM    1 2 DAT P N CARD     1 4 obus              X A
NUM    1 2 ACC P N CARD     1 1 o                 X A
NUM    1 2 ABL P N CARD     1 4 obus              X A
NUM    1 2 VOC P N CARD     1 1 o                 X A
NUM    1 3 NOM P C CARD     1 2 es                X A
NUM    1 3 GEN P X CARD     1 3 ium               X A
NUM    1 3 DAT P X CARD     1 4 ibus              X A
NUM    1 3 ACC P C CARD     1 2 es                X A
NUM    1 3 ACC P C CARD     1 2 is        X A
NUM    1 3 ABL P X CARD     1 4 ibus              X A
NUM    1 3 VOC P C CARD     1 2 es                X A
NUM    1 3 NOM P N CARD     1 2 ia                X A
NUM    1 3 ACC P N CARD     1 2 ia                X A
NUM    1 3 VOC P N CARD     1 2 ia                X A
NUM    1 4 NOM P M  CARD     1 1 i                X A
NUM    1 4 GEN P M  CARD     1 2 um               X A
NUM    1 4 GEN P M  CARD     1 4 orum             X A
NUM    1 4 DAT P M  CARD     1 2 is               X A
NUM    1 4 ACC P M  CARD     1 2 os               X A
NUM    1 4 ABL P M  CARD     1 2 is               X A
NUM    1 4 VOC P M  CARD     1 1 i                X A
NUM    1 4 NOM P F  CARD     1 2 ae               X A
NUM    1 4 GEN P F  CARD     1 4 arum             X A
NUM    1 4 DAT P F  CARD     1 2 is               X A
NUM    1 4 ACC P F  CARD     1 2 as               X A
NUM    1 4 ABL P F  CARD     1 2 is               X A
NUM    1 4 VOC P F  CARD     1 2 ae               X A
NUM    1 4 NOM P N  CARD     1 1 a                X A
NUM    1 4 GEN P N  CARD     1 2 um               X A
NUM    1 4 DAT P N  CARD     1 2 is               X A
NUM    1 4 ACC P N  CARD     1 1 a                X A
NUM    1 4 ABL P N  CARD     1 2 is               X A
NUM    1 4 VOC P N  CARD     1 1 a                X A
--  indeclinabili cardinals
NUM    2 0 X X X CARD     1 0                     X A
--  Ordinali
NUM    0 0 NOM S M  ORD     2 2 us                X A
```

```
NUM    0 0 GEN S M  ORD     2 1 i                X A
NUM    0 0 DAT S M  ORD     2 1 o                X A
NUM    0 0 ACC S M  ORD     2 2 um               X A
NUM    0 0 ABL S M  ORD     2 1 o                X A
NUM    0 0 VOC S M  ORD     2 1 e                X A
NUM    0 0 NOM P M  ORD     2 1 i                X A
NUM    0 0 GEN P M  ORD     2 4 orum             X A
NUM    0 0 DAT P M  ORD     2 2 is               X A
NUM    0 0 ACC P M  ORD     2 2 os               X A
NUM    0 0 ABL P M  ORD     2 2 is               X A
NUM    0 0 VOC P M  ORD     2 1 i                X A
NUM    0 0 NOM S F  ORD     2 1 a                X A
NUM    0 0 GEN S F  ORD     2 2 ae               X A
NUM    0 0 DAT S F  ORD     2 2 ae               X A
NUM    0 0 ACC S F  ORD     2 2 am               X A
NUM    0 0 ABL S F  ORD     2 1 a                X A
NUM    0 0 VOC S F  ORD     2 1 a                X A
NUM    0 0 NOM P F  ORD     2 2 ae               X A
NUM    0 0 GEN P F  ORD     2 4 arum             X A
NUM    0 0 DAT P F  ORD     2 2 is               X A
NUM    0 0 ACC P F  ORD     2 2 as               X A
NUM    0 0 ABL P F  ORD     2 2 is               X A
NUM    0 0 VOC P F  ORD     2 2 ae               X A
NUM    0 0 NOM S N  ORD     2 2 um               X A
NUM    0 0 GEN S N  ORD     2 1 i                X A
NUM    0 0 DAT S N  ORD     2 1 o                X A
NUM    0 0 ACC S N  ORD     2 2 um               X A
NUM    0 0 ABL S N  ORD     2 1 o                X A
NUM    0 0 VOC S N  ORD     2 2 um               X A
NUM    0 0 NOM P N  ORD     2 1 a                X A
NUM    0 0 GEN P N  ORD     2 4 orum             X A
NUM    0 0 DAT P N  ORD     2 2 is               X A
NUM    0 0 ACC P N  ORD     2 1 a                X A
NUM    0 0 ABL P N  ORD     2 2 is               X A
NUM    0 0 VOC P N  ORD     2 1 a                X A
--  Distributivi
NUM    0 0 NOM P M  DIST    3 1 i                X A
NUM    0 0 GEN P C  DIST    3 2 um         X A
NUM    0 0 GEN P M  DIST    3 4 orum             X A
NUM    0 0 DAT P X  DIST    3 2 is               X A
NUM    0 0 ACC P M  DIST    3 2 os               X A
NUM    0 0 ABL P X  DIST    3 2 is               X A
NUM    0 0 VOC P M  DIST    3 1 i                X A
NUM    0 0 NOM P F  DIST    3 2 ae               X A
NUM    0 0 GEN P F  DIST    3 4 arum             X A
NUM    0 0 ACC P F  DIST    3 2 as               X A
NUM    0 0 VOC P F  DIST    3 2 ae               X A
NUM    0 0 NOM P N  DIST    3 1 a                X A
NUM    0 0 GEN P N  DIST    3 4 orum             X A
NUM    0 0 ACC P N  DIST    3 1 a                X A
NUM    0 0 VOC P N  DIST    3 1 a                X A
```

```
--  Avverbi numerali
NUM    1 1 X   X X  ADVERB      4 0                X A
NUM    1 2 X   X X  ADVERB      4 0                X A
NUM    1 3 X   X X  ADVERB      4 0                X A
NUM    1 4 X   X X  ADVERB      4 3 ies            X A
NUM    1 4 X   X X  ADVERB      4 4 iens           B A
NUM    2 0 X   X X  ADVERB      4 3 ies            X A
NUM    2 0 X   X X  ADVERB      4 4 iens           B A
```

Bibliografia

Alberto, P. F. (2002). *O projecto Olissipo: uma aplicação no âmbito do ensino do latim*. Euphrosyne, vol. 30:pp. 335–338.

Artale, A., Magnini, B. e Strapparava, C. (1997). *Proceedings of ACL/EACL-97 Workshop Lexical discrimination with the Italian version of WordNet*. In P. V. et al. (Cur.), *Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications (Madrid, Spain, July 1997)*. Association for Computational Linguistics, ACL.

Atserias, J., Climent, S., Farreres, X., Rigau, G. e Rodríguez, H. (1997). *Combining Multiple Methods for the Automatic Construction of Multilingual WordNets*. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Baeza-Yates, R. e Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

Ballesteros, L. e Croft, W. B. (1998). *Resolving ambiguity for cross-language retrieval*. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, pp. 64–71. doi:http://doi.acm.org/10.1145/290941.290958.

Bentivogli, L. e Pianta, E. (2000). *Looking for lexical gaps*. In *Proceedings of the ninth EURALEX International Congress*. Stuttgart, Germany.

Blair, D. C. e Maron, M. E. (1985). *An evaluation of retrieval effectiveness for a full-text document-retrieval system.* Commun. ACM, vol. 28(3):pp. 289–299. doi:http://doi.acm.org/10.1145/3166.3197.

Bozzi, A. e Cappelli, G. (1990). *A Project for Latin Lexicography: 2. A Latin Morphological Analyzer.* Computers and the Humanities, vol. 24:pp. 421–426.

Brown, P. F., Pietra, S. D., Pietra, V. J. D. e Mercer, R. L. (1991). *Word-Sense Disambiguation Using Statistical Methods.* In *Meeting of the Association for Computational Linguistics*. pp. 264–270.

Busa, R. (1968). *Un lexique latin électronique.* In J. Stindková e Z. Skoumalová (Cur.), *Les machines dans la linguistique.* Académie Tchécoslovaque des sciences, pp. 251–269.

Busa, R. (1987). *Fondamenti di informatica linguistica.* Vita e Pensiero, Milano.

Busa, R. (1994). *Inquisitiones lexicologicae in Indicem Thomisticum.* CAEL, Milano.

Busa, R. (2000). *Dal computer agli angeli.* Itacalibri e BVE.

Busa, R. (2004). *De Forcellini Lexici Totius Latinitatis ad haec nostra tempora utilitate.* Euphrosyne, vol. XXIII:pp. 24–28.

Buzzetti, D. (2004). *Markup e rappresentazione del testo. Una discussione con Allen Renear.* Griselda Online, vol. 2. doi:10.1473/gri2.

Cappelli, G. e Passarotti, M. (2003). *LEMLAT: uno strumento computazionale per l'analisi linguistica del latino - sviluppo e prospettive.* Euphrosyne, vol. 31:pp. 519–531.

Chakravarthy, A. S. (1994). *Toward semantic retrieval of pictures and video.* In *RIAO 94 Conference Proceedings: Intelligent multimedia information retrieval systems and management*, vol. 1. CID, Paris, FRANCE, pp. 676–686.

Chakravarthy, A. S. e Haase, K. B. (1995). *NetSerf: using semantic knowledge to find Internet information archives*. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, pp. 4–11. doi:http://doi.acm.org/10.1145/215206.215326.

Chen, H. e Lynch, K. J. (1992). *Automatic Construction of Networks of Concepts Characterizing Document Databases*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 22(5):pp. 885–902.

Chen, H., Lynch, K. J., Basu, K. e Ng, T. D. (1993). *Generating, integrating, and activating thesauri for concept-based document retrieval*. IEEE Expert: Intelligent Systems and Their Applications, vol. 8(2):pp. 25–34. doi:http://dx.doi.org/10.1109/64.207426.

Chen, H.-H., Lin, C.-C. e Lin, W.-C. (2002). *Building a Chinese-English wordnet for translingual applications*. ACM Transactions on Asian Language Information Processing (TALIP), vol. 1(2):pp. 103–122. doi:http://doi.acm.org/10.1145/568954.568955.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. e Stein, C. (2001). *Introduction to Algorithms, Second Edition*. The MIT Press.

Cowie, A. P. (1981). *The Treatment of Collocations and Idioms in Learners' Dictionaries*. Applied Linguistics, vol. II(3):pp. 223–235. doi:10.1093/applin/II.3.223.

Cremaschi, G. (1959). *Guida allo studio del latino medievale*. Liviana editrice, Padova.

De Prisco, A. (1991). *Il latino tardoantico e altomedievale*. Jouvance.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. e Harshman, R. A. (1990). *Indexing by Latent Semantic Analysis*. Journal of the American Society of Information Science, vol. 41(6):pp. 391–407.

Diederich, P. B. (1939). *The Frequency Of Latin Words And Their Endings*. Tesi di dottorato, UNIVERSITY OF CHICAGO.

ERNOUT, A. (1953). *Morphologie historique du latin*. Klincksieck, Paris.

FELLBAUM, C. (Cur.) (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

GARDNER, D. (1971). *Frequency dictionary of Classical Latin Words*. Tesi di dottorato, Stanford University.

GHOSE, A. e DHAWLE, A. (1977). *Problems of thesaurus construction*. Journal of the American Society for Information Science, vol. 28(4):pp. 211–217.

GIGER, H. P. (1988). *Concept based retrieval in classical IR systems*. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, pp. 275–289. doi:http://doi.acm.org/10.1145/62437.62461.

GINZBURG, C. (1979). *Crisi della ragione*, cap. Spie. Radici di un paradigma indiziario. Einaudi.

GORDON, M. D. (1997). *It's 10 a.m. Do you know where your documents are? The nature and scope of information retrieval problems in business*. Information Processing and Management, vol. 33(1):pp. 107–122. doi:http://dx.doi.org/10.1016/S0306-4573(96)00021-0.

GROSSMAN, D. A. e FRIEDER, O. (2004). *Information Retrieval: algorithms and heuristics*. Springer, Dordrecht, 2 ed.

GRUBER, T. R. (1993). *A translation approach to portable ontologies*. Knowledge Acquisition, vol. 5(2):pp. 199–220.

HINES, T. C. e HARRIS, J. L. (1971). *Columbia university school of library service system for thesaurus development and maintenance.* Information Storage and Retrieval, vol. 7(1):pp. 39–50.

HIRST, G. (1987). *Semantic interpretation and the resolution of ambiguity*. Cambridge University Press, New York, NY, USA.

HUTCHINS, J. e SOMERS, H. L. (1992). *An introduction to machine translation*. Academic Press, London.

KANTOR, P. (1994). *Information retrieval techniques*. Annual Review of Information Science and Technology, vol. 29:pp. 53–90.

KNUTH, D. E. (1998). *Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Professional, 2 ed.

KRISTENSEN, J. (1993). *Expanding end-users' query statements for free text searching with a search-aid thesaurus*. Inf. Process. Manage., vol. 29(6):pp. 733–744. doi:http://dx.doi.org/10.1016/0306-4573(93)90102-J.

KROVETZ, R. e CROFT, W. B. (1992). *Lexical Ambiguity and Information Retrieval*. Information Systems, vol. 10(2):pp. 115–141.

LAMARRA, A. (2004). *De Forcellini Lexici Totius Latinitatis ad haec nostra tempora utilitate*. Euphrosyne, vol. XXIII:pp. 89–97.

LEE, C., LEE, G. G. e SEO, J. (2004). *Multiple Heuristics and Their Combination for Automatic WordNet Mapping*. Computers and the Humanities, vol. 38(4):pp. 437–455.

LENAT, D. B. e GUHA, R. V. (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

LEVENSHTEIN, V. I. (1966). *Binary codes capable of correcting deletions, insertions, and reversals*. Soviet Physics Doklady, vol. 10(8):pp. 707–710.

LIU, S., LIU, F., YU, C. e MENG, W. (2004). *An effective approach to document retrieval via utilizing WordNet and recognizing phrases*. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, pp. 266–272. doi:http://doi.acm.org/10.1145/1008992.1009039.

MAGNINI, B. e CAVAGLIÀ, G. (2000). *Integrating subject field codes into WordNet*. In *Proceedings of LREC-2000, 2nd International Conference on Language Resources and Evaluation*. Atene, pp. 1413–1418.

MARINONE, N. (1990). *A Project for Latin Lexicography: 1. Automatic Lemmatization and Word-List*. Computers and the Humanities, vol. 24:pp. 417–420.

MILLER, G. A., BECKWITH, R., FELLBAUM, C., GROSS, D. e MILLER, K. J. (1990). *Introduction to WordNet: an on-line lexical database*. International Journal of Lexicography, vol. 3(4):pp. 235 – 244.

MINSKY, M. (1975). *A framework for Representing Knowledge*. In P. Winston (Cur.), *The Psychology of Computer Vision*. McGraw Hill, New York, pp. 211–277.

MURRAY-RUST, P. (2002). *Scientific Publications in XML - towards a global knowledge base*. Data Science, vol. 1:pp. 84–98.

NENCIONI, G. (1987). *Verso una nuova lessicografia*. In A. Cappelli, L. Cignoni e C. Peters (Cur.), *Studies in Honour of Roberto Busa S. J.*, Linguistica Computazionale. Giardini, Pisa, pp. 133–150.

NOVAK, V. (1992). *Fuzzy Sets in Natural Language Processing*. In R. R. Yager e L. A. Zadeh (Cur.), *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer, Boston, pp. 185–200.

OCHS, E. (1979). *Transcription as a theory*. In E. Ochs e B. Schieffelin (Cur.), *Developmental pragmatics*. Academic Press, New York.

ODELL, M. K. e RUSSELL, R. C. (1918/1922). *U.S. Patents 1261167 (1918), 1435663 (1922)*†. Brevetto non pubblicato. Citato in Knuth (1998).

ORLETTI, F. e TESTA, R. (1991). *La trascrizione di un corpus interlingua: aspetti teorici e metodologici*. Studi italiani di linguistica teorica e applicata, vol. 20:pp. 245–282.

PALMER, R. (1977). *La Lingua Latina (tr. italiana)*. Einaudi, Torino.

PASSAROTTI, M. e RUFFOLO, P. (2004). *L'utilizzo del lemmatizzatore LEMLAT per una sistemazione dell'omografia in latino*. Euphrosyne, vol. 32 Nova Série:pp. 99–110.

PEIRCE, C. S. (1980). *Semiotica*. Einaudi, Torino.

PIANTA, E., BENTIVOGLI, L. e GIRARDI, C. (2002). *MultiWordNet: Developing an aligned multingual database*. In *Proceedings of the 1st International WordNet Conference*. Mysore, India, pp. 293–302.

POCCETTI, P., POLI, D. e SANTINI, C. (1999). *Una storia della lingua latina*. Carocci, Roma.

PORTER, M. F. (1997). *An algorithm for suffix stripping*. Readings in information retrieval, vol. 14(3):pp. 313–316.

QUEMADA, B. (1990). *Les données lexicographiques et l'ordinateur*. Cahiers de Lexicologie, vol. LVI(I-II):pp. 170–189.

RADA, R., MILI, H., BICKNELL, E. e BLETTNER, M. (1989). *Development and application of a metric on semantic nets*. IEEE Transactions on Systems, Man and Cybernetics, vol. 19(1):pp. 17–30.

RAU, L. F. (1987). *Knowledge organization and access in a conceptual information system*. Inf. Process. Manage., vol. 23(4):pp. 269–283. doi:http://dx.doi.org/10.1016/0306-4573(87)90018-5.

RICHARDSON, R. (1994). *A semantic-based approach to information processing*. Tesi di dottorato, Doublin City University.

ROUSSEY, C., CALABRETTO, S. e PINON, J.-M. (1999). *Etat de l'art en indexation et recherche d'information*. Revue Document Numérique, vol. 3(3-4):pp. 121–150.

SACKS, H., SCHEGLOFF, E. A. e JEFFERSON, G. (1974). *A simplest systematic for the organization turn taking for conversation*. Language, vol. 50(4):pp. 696–735.

Salton, G. (1969). *A comparison between manual and automatic indexing methods*. Journal of American Documentation, vol. 20(1):pp. 61–71.

Salton, G. e Lesk, M. E. (1968). *Computer Evaluation of Indexing and Text Processing*. J. ACM, vol. 15(1):pp. 8–36. doi:http://doi.acm.org/10.1145/321439.321441.

Salton, G. e Lesk, M. E. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*, cap. Information Analysis and Dictionary Construction. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, pp. 115–142.

Salton, G., Wong, A. e Yang, C. S. (1975). *A Vector Space Model for Automatic Indexing*. Commun. ACM, vol. 18(11):pp. 613–620.

Schinke, R., Greengrass, M., Robertson, A. e Willett, P. (1997). *Retrieval of morphological variants in searches of Latin text databases*. Computing and the Humanities, vol. 5:pp. 409–432.

Schutze, H. e Pedersen, J. O. (1997). *A co-occurrence based thesaurus and two applications to information retrieval*. Information Processing and Management, vol. 33(3):pp. 307–318.

Stotz, P. (1996). *Handbuch zur lateinische Sprache des Mittelalters*. Beck.

Sussna, M. (1993). *Word sense disambiguation for free-text indexing using a massive semantic network*. In *CIKM '93: Proceedings of the second international conference on Information and knowledge management*. ACM Press, New York, NY, USA, pp. 67–74. doi:http://doi.acm.org/10.1145/170088.170106.

Traina, A. e Perini, G. B. (1972). *Propedeutica al latino universitario*. Patron, Bologna, 1 ed.

Tsujii, J. e Ananiadou, S. (2005). *Thesaurus or logical ontology, which one do we need for text mining?* Language Resources an Evaluation, vol. 39(1):pp. 77–90.

VOORHEES, E. M. (1993a). *On Expanding Query Vectors with Lexically Related Words*. In *Text REtrieval Conference*. pp. 223–232.

VOORHEES, E. M. (1993b). *Using WordNet to disambiguate word senses for text retrieval*. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, pp. 171–180. doi:http://doi.acm.org/10.1145/160688.160715.

VOORHEES, E. M. (1994). *Query expansion using lexical-semantic relations*. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., New York, NY, USA, pp. 61–69.

VOSSEN, P. (1996). *Right or wrong: combining lexical resources in the EuroWordNet project*. In M. Gellerstam, J. Jarborg, S. Malmgren, K. Noren, L. Rogstrom e C. Papmehl (Cur.), *Proceedings of Euralex-96*. Goetheborg, pp. 715–728.

WALLACH, H. M. (2006). *Topic modeling: beyond bag-of-words*. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*. ACM Press, New York, NY, USA, pp. 977–984. doi:http://doi.acm.org/10.1145/1143844.1143967.

WANG, Y.-C., VANDENDORPE, J. e EVENS, M. (1985). *Relational thesauri in information retrieval*. J. Am. Soc. Inf. Sci., vol. 36(1):pp. 15–27.

ZADEH, L. A. (1992). *Knowledge Representation in Fuzzy Logic*. In R. R. Yager e L. A. Zadeh (Cur.), *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer, Boston, pp. 1–25.

ZAMPOLLI, A. (1968). *L'elaboratore elettronico negli studi linguistici*. Rivista IBM, (2):pp. 14–19.

ZAMPOLLI, A. e DURO, A. (1968). *Analisi lessicali mediante elaboratori elettronici*. In *Atti del Convegno sul tema L'automazione elettronica e le sue implicazioni scientifiche, tecniche e sociali*, 110. Accademia Nazionale dei Lincei, pp. 119–139.

Indice analitico