razione, e sono poi i nomi che verranno successivamente utilizzati per le ricerche all'interno del corpus. Come esempio, la query

[2066a] `CT> [lemma = "avere"];` *comando CQP da shell Linux,*

digitata (in locale) nella riga di comando (*prompt*) del Corpus Query Processor, mostrerà (nella visualizzazione di default sul terminale, comunque modificabile) tutte le occorrenze di qualsiasi forma di *avere* presente nel CT, racchiusa tra parentesi uncinate ed inserita in un piccolo contesto testuale (definito per numero di caratteri):

```
[2066b] 2:    D' amore <abiendo> gioia interamente ,
        9:    ente , lasso , nonn¬ <aio> in altro intendimen
        27:   l primo loco là onde <avea> abento ; ma feci co
        46:   talento : traduto m' <àn> li sguardi che sove
        ...   ...                          risultato in locale della query 2066a.
```

Come mostrato nelle prime quattro righe del risultato, una particolare forma di *avere* è già presente nella riga 2[20] del corpus alla terza posizione assoluta, l'occorrenza successiva si trova in riga 9 e così via. A questo punto il corpus è pronto per essere interrogato, da terminale (come in 2066a) o tramite web.

20.1.6 *ENCODING*: NOTE FINALI E SCHEMA RIASSUNTIVO. [MT] La descrizione della procedura da seguire per la costituzione di un corpus interrogabile mediante la piattaforma CWB, ha preso finora in considerazione unicamente i passaggi più significativi, strettamente legati all'utilizzo di specifici programmi di elaborazione. Tuttavia, è importante precisare che, oltre a quanto finora descritto, risulta necessario svolgere alcune operazioni supplementari che, sebbene possano apparire per certi versi meno significative e sostanzialmente ovvie, sono assolutamente necessarie al fine di ottenere un corpus realmente utilizzabile, e pertanto vanno comunque precisate. Le operazioni in questione si riducono sostanzialmente a due, che all'interno della sequenza di passaggi, trovano il loro inserimento nel momento immediatamente successivo alla procedura di *encoding*. Scendendo più concretamente nei dettagli, la prima operazione consiste nel copiare tutti i file generati dall'*encoder* nella cartella che ospiterà il corpus, area solitamente già definita in fase di installazione del CWB[21]. Ciò fatto, il passo seguente, conclusivo, consisterà nella creazione dei file di lessico ed indice, mediante l'applicazione del comando `cwb-makeall` alla cartella di destinazione.

```
1.  recode -v cp850..lat1 testiDOSdaconvertire.txt
2.  ./Aufbereitung_new.perl CT_disambiguato.txt > tmp_1.txt
3.  gawk -f Fix_uncinate.awk tmp_1.txt > tmp_2.txt
4.  ../mwl-hinzu.perl -m MultiWord.txt tmp_2.txt > CT_encode.txt
5.  mkdir AreaTemporanea
6.  cd AreaTemporanea
7.  cwb-encode -t ../CT_encode.txt -P lemma -P pos -P kat -P typ -P corr -P
    genre -P msform -P philform -P mwlword -P mwlkat -P mwlnum -V author -V
    title -V line -V page -V par -V type -V chapter -V s -V genr -V mwl
8.  cp -f *.* PercorsoAreaDefinitiva (es. /usr/etc/corpora/ant2009
9.  cwb-makeall AreaDefinitiva (es. ant2009)
```

Tav. 239: Sinossi della procedura della preparazione del CT in CWB.

[20] Nel CWB il conteggio, si badi però, parte da 0.

[21] Trattandosi di un argomento esclusivamente informatico, riservato ai tecnici del settore, la procedura di installazione del CWB e dei relativi corpora non verrà trattata in questa sede. Per eventuali informazioni al riguardo, si faccia riferimento alla documentazione tecnica fornita insieme al programma.

Nella Tav. 239 è quindi riportato lo schema riassuntivo dei passaggi necessari per la trasformazione di un corpus da un semplice formato testuale ad un formato-CQP.

Una prima osservazione è che al passo 3. è menzionato l'uso di uno script non precedentemente descritto né menzionato: sarà appunto l'argomento del § 20.2 e sottoparagrafi.

Come nota conclusiva, infine, ci sembra corretto segnalare che le operazioni facenti capo ai punti 5. e 6., pure non descritte in precedenza, esulano dalla procedura di *encoding* vera e propria: la loro funzione, di fatto, si limita unicamente alla creazione di un'area temporanea di deposito dei file generati dal processo di codifica. Tale azione risulta di particolare comodità al fine di separare i testi di origine da quelli costituenti la versione CQP, in modo da rendere questi ultimi già pronti per affrontare le successive elaborazioni.

20.2 UN'APPENDICE: IL TRATTAMENTO DELLE CASSATURE. Il processo di trasformazione del corpus dal formato-CT al formato-CQP descritto nel precedente blocco di paragrafi da Arne Fitschen, così come gli script da lui preparati, sono stati messi a punto tra il 1999 ed il 2000, ed in séguito sostanzialmente mantenuti come tali.

Fa eccezione a ciò un'unica questione, di cui ci siamo accorti solo molto tardi (addirittura in Ver. 1.7!): il trattamento delle cassature ed espunzioni (cfr. § 8.5.2.7), nella versione CT affidato alla combinazione delle (tradizionali) uncinate col carattere di trasparenza, nella versione CQP non corrispondeva alle aspettative.

20.2.1 IL PROBLEMA E LA SUA SOLUZIONE. Stante il trattamento delineato poco sopra (§ 20.1.3) per la trasformazione di quel tipo di markup filologico (cioè tonde, quadre e corsivi filologici), il risultato in formato-CQP fino alla versione 1.7 era del tipo presentato nell'esempio in tavola seguente[22]:

<s 2484>											
In	in	|adp.pre|	|56,0,0,0,0,0|	P	n	Did	In	In	-	-	0
questo	questo	|pd.dem.s|	|30,0,4,6,0,0|	P	n	Did	questo	questo	-	-	0
mezzo	mezzo	|n.c|	|20,0,4,6,0,0|	P	n	Did	mezzo	mezzo	-	-	0
</line>											
<line 13>											
genti	gente	|n.c|	|20,0,5,7,0,0|	P	n	Did	genti	genti	-	-	0
che	che	|pd.rel|	|36,0,4;5,6;7,0,0|	P	n	Did	che	che	-	-	0
passavano	passare	|v.m.f.ind.ipf|	|112,3,0,7,0,0|	P	n	Did	passavano	passavano	-	-	0
<per>	per	|adp.pre|	|56,0,0,0,0,0|	P	n	Did	<per>	<per>	-	-	0
<la>	la	|art.d|	|60,0,5,6,0,0|	P	n	Did	<la>	<la>	-	-	0
<via>	via	|n.c|	|20,0,5,6,0,0|	P	n	Did	<via>	<via>	-	-	0
per	per	|adp.pre|	|56,0,0,0,0,0|	P	n	Did	per	per	-	-	0
lo	lo	|art.d|	|60,0,4,6,0,0|	P	n	Did	lo	lo	-	-	0
camino	cammino	|n.c|	|20,0,4,6,0,0|	P	n	Did	camino	camino	-	-	0
trovaro	trovare/-si/	|v.m.f.ind.pt|	|113,3,0,7,0,0|	P	n	Did	trovaro	trovaro	-	-	0
</line>											

Tav. 240: Un esempio con cassature dalla Ver. 1.7.

[22] L'ordine degli attributi posizionali, colonna per colonna, è 1 `lemma`, 2 `pos`, 3 `kat`, 4 `typ`, 5 `corr`, 6 `genre`, 7 `msform`, 8 `philform`, 9 `mwlword`, 10 `mwlkat` ed 11 `mwlnum`; ci si scusa che questa tavola e la seguente siano state stampate in Times anziché in Courier (come avrebbero dovuto) per mere ragioni tipografiche di spazio; per le medesime ragioni, anziché il doppio trattino <--> è usato quello semplice <->.