CIRCE

Comprehensive Istat R Coding Environment

Versione 1.0

Luglio 2016

A cura di: Laura Capparucci, Massimiliano Degortes, Loredana Mazza

INDICE[1]

[1] Il lavoro è frutto dell'attività di ricerca e sviluppo congiunta degli autori. In ogni caso, ai soli fini dell'attribuzione, i capitoli 1 e 5 sono da attribuirsi a Loredana Mazza , i capitoli 2 e 3 a Laura Capparucci, il capitolo 4 e le appendici A e B a Massimiliano Degortes. Si ringraziano inoltre Maria Teresa Buglielli e Diego Zardetto per il supporto fornito allo sviluppo del software.

1. Introduzione

Il presente documento è una guida per gli utilizzatori di CIRCE che descrive le caratteristiche e le funzionalità del prodotto per poi fornire informazioni di dettaglio sul come utilizzarle attraverso l'interfaccia grafica.

CIRCE sostituisce ACTR v3, sviluppato da Statistics Canada, che non essendo più supportato non era compatibile con i nuovi sistemi Windows 7 e Windows Server 2008 verso cui l'Istat ha migrato.

CIRCE ricalca ACTR v3 ampiamente descritto nel volume della collana Istat "Tecniche e strumenti n.4 -2007" cui si rimanda per ulteriori dettagli sull'algoritmo di *matching* e la creazione delle basi informative. Questa scelta è stata dettata dall'esigenza di garantire agli utenti gli stessi livelli di qualità della codifica raggiunti con il precedente sistema ampiamente utilizzato in Istituto.

Il presente manuale utente si divide in due parti:

- la prima parte (fino al paragrafo 3.6) descrive l'algoritmo di *matching* e le funzioni di standardizzazione che saranno denominate *parsing*;
- la seconda parte (dal capitolo 4) descrive come utilizzare il software attraverso l'interfaccia grafica.

Nota per l'utilizzatore: L'unità MSS/C è responsabile del software e continua a svolgere la funzione di supporto alla codifica per le variabili testuali, afferenti alle classificazioni ufficiali, rilevate in corso di indagine.

2. Installazione di CIRCE

I requisiti minimi per il funzionamento di CIRCE in ambiente pc sono:

- Sistema operativo Windows 7 o superiori
- Microsoft Framework .NET 4
- Software R installato

Momentaneamente il pacchetto funziona solo per la lingua italiana, che deve essere impostata nella sezione "Impostazione lingua" del pannello di controllo, ed è stato testato su Windows7 e Windows10.

Per l'installazione, dopo aver scaricato e spacchettato la cartella CIRCE.zip dal sito web istat[2], eseguire il file setup_circe.exe[3].

Il setup chiederà di accettare la licenza. Successivamente richiede il path dove installare tutti i componenti del software CIRCE: il default è C:\Circe, ma è modificabile con il bottone "browse". A questo punto viene proposto il path in cui è installato R, il default è C:\Program Files\R\R-3.1.1, ma è modificabile.

Seguirà l'impostazione del nome del pacchetto da mettere nel menu Start e la richiesta di creazione di un'icona su desktop. Infine verrà visualizzata la finestra dell'installazione. Cliccando su Install il pacchetto verrà installato secondo i parametri specificati in precedenza.

[2] http://www.istat.it/it/strumenti/metodi-e-strumenti-it/strumenti-di-elaborazione/circe
[3] Il nome dei file CIRCE.zip e setup_circe.exe è seguito dal numero della versione.

3. Il nuovo pacchetto di codifica CIRCE

Il nuovo pacchetto di codifica CIRCE si basa sul software R ed ha come scopo l'attribuzione automatica di un codice a partire da un testo, indipendentemente dalla lingua e dalla classificazione adottata.

Il software è stato sviluppato in modo tale da garantire almeno gli stessi risultati qualitativi e quantitativi raggiunti da ACTR v3, prendendo come elemento di confronto la performance di ACTR v3 per la codifica della variabile Ateco (Gruppo di lavoro: D08 97 DGEN 14).

CIRCE riproduce il comportamento di ACTR v3, ma, essendo sviluppato dall'Istat, offre l'opportunità di modifiche e/o aggiunte di nuove funzionalità.

La sviluppo in R ha reso CIRCE portabile su diversi ambienti senza necessità di compilazione. Questo ha permesso di realizzare un unico pacchetto di codifica funzionante sia in ambiente Windows che Linux. Quindi, a differenza di ACTR v3, CIRCE è utilizzabile sia in ambiente pc, attraverso un'interfaccia grafica utente, che in ambiente web, attraverso la "chiamata" ad un web service. In quest'ultimo caso è attualmente disponibile un web service dedicato alla codifica dell'Ateco[4]. Su richiesta, il servizio può essere replicato per la codifica delle altre variabili testuali afferenti alle classificazioni ufficiali.

Di contro, essendo basato su R, non garantisce la stessa velocità di esecuzione propria dei pacchetti scritti utilizzando linguaggi compilati. Tale criticità riguarda la funzione di codifica batch, i cui tempi sono superiori rispetto al precedente software. È in fase di studio la possibilità di sviluppare un contesto che lavori in modalità multi-processore per rendere più veloce l'applicazione.

3.1 Funzionamento del software

CIRCE prevede un confronto tra la stringa di testo da codificare e le voci contenute nel dizionario della classificazione.

Sia il dizionario che il testo da codificare vengono preliminarmente sottoposti ad un processo di standardizzazione o *parsing* che ha lo scopo di eliminare dal testo la variabilità grammaticale o sintattica che non incide sull'aspetto semantico, ma soltanto sulla forma, e che pertanto è irrilevante ai fini dell'abbinamento con le voci del dizionario.

La prima operazione da eseguire è il caricamento della base informativa relativa allo specifico dizionario della classificazione. In questa fase vengono anche calcolati i pesi delle singole parole che fanno parte del dizionario.

Nella fase di codifica vengono selezionate le voci del dizionario che risultano essere uguali o simili al testo da codificare in base alle parole che hanno in comune. La selezione avviene sulla base del punteggio di *matching* che viene calcolato considerando sia i pesi delle parole che alcuni parametri che stabiliscono le soglie di precisione. Nel caso in cui la corrispondenza sia esatta, oppure venga selezionata una sola voce che supera un certo valore di soglia, viene associato al testo da codificare un codice univoco o "unico". Nel caso in cui la corrispondenza non sia esatta, o non superi una certa soglia, la procedura fornisce un output composto dalle voci del dizionario che più si avvicinano al testo ("multipli" e "possibili"). Se non c'è alcuna voce del dizionario che supera la soglia minima, il testo da codificare viene definito "fallito" (per la descrizione dell'algoritmo di *matching* si rimanda al paragrafo 3.5).

[4] Il web service per la codifica dell'Ateco è accessibile attraverso al pagina http://www.istat.it/it/strumenti/definizioni-e-classificazioni/ateco-2007

La “chiamata” a CIRCE, da qualsiasi ambiente provenga (web o pc), deve necessariamente contenere 4 parametri di tipo stringa, anche quando non sono tutti utilizzati dalla funzione richiesta. In questo ultimo caso, per il parametro non necessario, è sufficiente indicare una stringa vuota. I parametri citati sono i seguenti:

1. path**:** path della cartella di contesto;
2. funzione**:** "C" = caricamento dizionario, "O" = codifica online/interattiva, "B" = codifica batch, "W" = codifica web;
3. stringa**:** testo da codificare, per la codifica online/interattiva e web;
4. progetto**:** nome del file .prg (contenuto nella cartella di contesto) da utilizzare per la codifica.

3.2 Il contesto

Sia sulla piattaforma Linux che Windows, CIRCE prevede una cartella in cui è definito il “contesto”, cioè l’ambiente utilizzato per eseguire la codifica.

Il “contesto” rappresenta l’insieme di tutti i file che concorrono a definire la base informativa necessaria per la codifica, in relazione alla classificazione adottata (Ateco, Professione, ecc.). Pertanto il contesto è costituito da: il dizionario della classificazione di riferimento, la strategia ed i file di *parsing*, il file di input (ovvero i testi da codificare), i file di output contenenti i risultati della codifica ed il file di progetto.

La struttura della cartella di contesto deve prevedere al suo interno le sotto-cartelle rappresentate nella Figura 1 che dovranno essere denominate esattamente come mostrato, mentre il nome della cartella di contesto può essere personalizzato dall’utente.

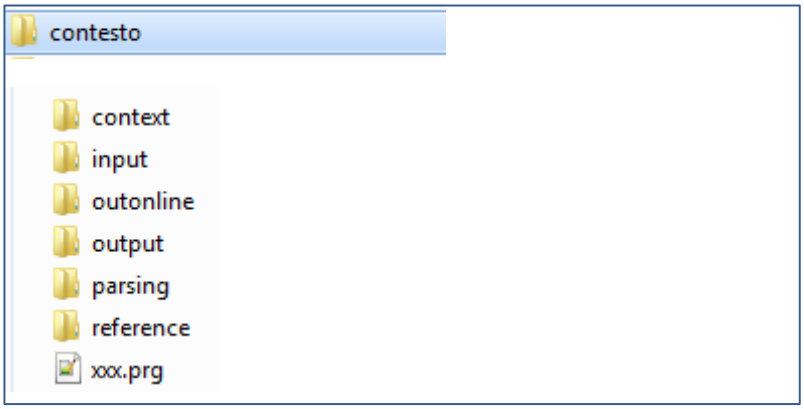

Figura 1: Struttura della cartella contesto

Le singole sotto-cartelle contengono:

- context: il dizionario standardizzato della classificazione di riferimento (in formato .txt), i file relativi ai pesi delle parole ed il file degli eventuali duplicati (in formato .csv), l’ambiente R contenente il database (in formato .Rdata);
- input: il file di input per la codifica batch (in formato .csv);
- outonline: i file di appoggio per la visualizzazione della codifica interattiva effettuata attraverso la maschera utente;
- output: i file di output della codifica batch ed il report (in formato .csv) sull’esito della codifica;
- parsing: i file predisposti per la standardizzazione dei testi ed il file della strategia di *parsing* (in formato .txt);
- reference: il dizionario della classificazione di riferimento (in formato .txt), ed il relativo tracciato (Config.txt).

Il file di progetto, genericamente indicato con "xxx .prg"**,** contiene la descrizione della struttura del file di input per la codifica batch ed i parametri della strategia di codifica, così come sarà descritto nel paragrafo 4.3. Inoltre permette di indicare, in fase di creazione del database, il percorso (path e nome) della cartella del contesto di riferimento.

3.3 Standardizzazione del testo (*parsing*)

La fase di standardizzazione di CIRCE ricalca quella di ACTR v3 *(Tecniche e strumenti n.4 -2007)* ed utilizzata come base di partenza.

È stato tuttavia necessario mettere a punto una fase di sperimentazione empirica in quanto non si disponeva né del codice sorgente di ACTR v3 né di una descrizione esaustiva e tecnicamente dettagliata del suo funzionamento. Procedendo empiricamente è stato possibile avere informazioni fondamentali sulle procedure di trasformazione dei testi e di calcolo del punteggio di *matching*.

In particolare, questa fase sperimentale ha permesso di capire che la sequenza con cui eseguire le trasformazioni dei testi doveva essere dettata dall'ordine in cui le singole parole apparivano nella frase di input e non dall'ordine in cui erano scritte le trasformazioni stesse all'interno dei file di *parsing*. Inoltre, ha consentito di determinare correttamente tutti gli elementi che costituiscono la formula del peso delle parole, [P(Wi) = 1 – log(NWi)/log(N)], per la quale la documentazione disponibile forniva una descrizione non esaustiva per la sua implementazione software.

Nella fase di caricamento del dizionario e nella fase di codifica, tutti i testi vengono standardizzati attraverso una serie di trasformazioni realizzate in sequenza. Il *parsing* fornisce diverse funzioni di trasformazione, quali, ad esempio, la rimozione dei caratteri ininfluenti, delle parole inutili, di suffissi/prefissi, l'individuazione di sinonimi, ecc. La finalità del *parsing* è quindi quella di rimuovere tutte le varianti grammaticali e sintattiche in modo da rendere uguali due descrizioni diverse ma dallo stesso contenuto semantico.

Il *parsing* è suddiviso in quattro *fasi* principali:

1. Pre-trattamento
2. Trattamento delle stringhe
3. Trattamento delle parole
4. Post-elaborazione

Il pre-trattamento è orientato ad eliminare e/o sostituire tutti i caratteri ininfluenti o che potrebbero creare ambiguità nei successivi passaggi (es. spazi iniziali e finali, vocali accentate, ecc.).

Nel trattamento delle stringhe il testo è elaborato come una stringa continua di caratteri. Il principale obiettivo di questa fase è facilitare il riconoscimento di particolari sequenze di caratteri, così come appaiono nel testo da codificare. Risulta particolarmente utile, ad esempio, per la gestione delle abbreviazioni standard (es. T.V. = TELEVISIONE) in quanto i punti vengono sostituiti da spazi nella successiva fase di separazione delle parole e senza questa trasformazione rimarrebbero due parole: T e V che non avrebbero alcun senso ai fini della codifica.

Dopo il trattamento delle stringhe, il testo viene suddiviso in parole, che costituiranno quindi l'input delle fasi successive. Il trattamento delle parole consiste in sostituzioni e/o eliminazioni delle stesse in base alle specifiche indicate nelle trasformazioni previste e nella gestione di prefissi, suffissi e caratteri doppi.

Infine, nella post-elaborazione, avviene l'eliminazione delle parole duplicate e l'ordinamento alfabetico delle stesse.

Tutti i file relativi al processo di standardizzazione si trovano nella sottocartella *parsing* del contesto: ad ogni file è associato un tipo di trasformazione. L'ordine con cui saranno effettuate le trasformazioni è indicato nel file *strategy.txt*. A differenza di ACTR v3, CIRCE non prevede controlli né sulla sequenza delle trasformazioni, né sul numero di ripetizioni delle stesse. Questa scelta è stata fatta per superare le limitazioni di ACTR v3, offrendo così un maggiore grado di flessibilità e, quindi, di personalizzazione della strategia di codifica, anche in funzione di eventuali sviluppi futuri. E' importante precisare comunque che la personalizzazione della strategia deve avvenire rispettando la sequenza delle quattro fasi sopra descritte.

Di seguito è riportata una tabella sintetica delle possibili trasformazioni dei testi:

Trasformazione	Fase/Descrizione	File dei dati
Fase: Pre-trattamento		
Autotrim	caratteri iniziali e finali: toglie tutti i blank, tabulatori e newline	-
trimStrimming	toglie tutti i blank, tabulatori e newline rispettivamente a sinistra e a destra dei caratteri indicati nei rispettivi file	trim_left.txt trim_right.txt
CCHR	trasforma i caratteri indicati nella I col. in quelli della II col.	Cchr.txt
WCHR	trasforma i caratteri indicati nella I col. in quelli della II col. I caratteri in questione sono diversi da quelli della precedente trasformazione	WChr.txt
Fase: Trattamento delle stringhe		
DCLS	cancella tutto ciò che è compreso tra incisi (tra il carattere/stringa della I col. e quello della II col.)	DCls.txt
DSTR	cancella le stringhe ritenute inutili	DStr.txt
RSTR	sostituzione di stringhe indicate nella I col. in quelle della II col.	rstr.txt
SepWord	trasforma in blank tutti i caratteri che non fanno parte della II colonna del file. Toglie tutti i blank all'inizio e alla fine e i doppi blank fra le parole, in modo che un solo blank separi ogni parola	WChr.txt
Fase: Trattamento delle parole		
RWRD	consente la gestione dei sinonimi e delle parole ininfluenti. Sostituisce le singole parole indicate nella I col. con una o due parole (anche blank) indicate rispettivamente nelle col. II e III (se esistono)	rwrd.txt
DWRD	consente la gestione di sinonimi a livello di coppie di parole. Sostituisce le coppie di parole indicate nelle col. I e II con quelle della III e della IV col. (se esistono)	dwrd.txt
HWRD	sostituisce le coppie di parole contenenti il trattino indicate nella I col. con quelle della II, della III e della IV col. (se esistono)	HWrd.txt
IWRD	cancella tutte le parole che contengono i caratteri indicati nel file	IWrd.txt
EXCP	permette di definire le parole, elencate nel file, che non devono subire le successive trasformazioni (PRFX, SUFX e MCHR) e pertanto tale passaggio va sempre anteposto a queste trasformazioni	Excp.txt
PRFX	prefissi da eliminare	Prfx.txt
SUFX	suffissi da eliminare	Sufx.txt
MCHR	caratteri doppi o tripli da rimuovere	MChr.txt
Fase: Post-elaborazione		
RDUP	cancella tutte le parole duplicate	
SORT	ordina le parole secondo un ordine alfabetico ascendente	

Per alcune di queste trasformazioni è necessario aggiungere ulteriori dettagli:

WCHR effettua una trasformazione di singoli caratteri: converte principalmente tutte le lettere in maiuscolo e toglie le vocali accentate. La seconda colonna del file WChr.txt serve, inoltre, ad individuare i caratteri validi ai fini del riconoscimento delle singole parole nel passaggio successivo denominato SepWord. La funzione **WCHR è** obbligatoria.

CCHR effettua anch'essa una trasformazione di singoli caratteri, utilizzando però un altro file (Cchr.txt), che non serve per il riconoscimento delle parole. Questa funzione non era presente in ACTR v3 ed è stata inserita per gestire alcuni particolari caratteri (es. apice ' che diventa ').

DSTR cancella le stringhe presenti nel file DStr.txt. Le stringhe da cancellare vengono preventivamente ordinate (internamente al programma) per lunghezza decrescente per evitare che, nel caso una sia contenuta in un'altra, la cancellazione non avvenga correttamente. Se ad esempio si vogliono eliminare le stringhe ALL' e DALL', e la trasformazione venisse eseguita in questo ordine, la cancellazione della seconda stringa non sarebbe possibile in quanto non esisterebbe più (rimarrebbe infatti solo il carattere D).

RSTR sostituisce le stringhe. Come già detto può essere molto utile per standardizzare, ad esempio, abbreviazioni standard (T.V.=TELEVISIONE), ma, poiché un suo erroneo utilizzo può comportare gravi errori nella fase di standardizzazione, se ne consiglia un uso limitato, preferendo, quando possibile, il ricorso ai processi orientati alle parole.

Se ad esempio fosse prevista nel file rstr.txt la trasformazione: USO MEDICINALE -> USOMEDICO e la stringa da codificare fosse 'ARTICOLI AD USO MEDICINALE', il risultato del *parsing* sarebbe 'ARTICOLI AD USOMEDICO', ossia una trasformazione corretta. Se invece la stringa da codificare fosse 'INFUSO MEDICINALE', essendo la stringa "USO" parte della parola "INFUSO", il risultato sarebbe 'INF USOMEDICO', perdendo così ogni riferimento alla parola INFUSO e trasformando la frase in una priva di senso compiuto .

SepWord è il passaggio finale della fase di trattamento delle stringhe ed è obbligatorio in quanto permette di identificare le singole parole che compongono il testo, che rappresentano l'input per le fasi successive relative al trattamento delle parole. Queste ultime vengono identificate attraverso l'individuazione dei caratteri validi e la sostituzione di tutti gli altri con il *blank*, che verrà così a rappresentare il carattere separatore tra parole.

DWRD effettua una scansione del testo per coppie di parole, sostituendole con una o due parole o con il *blank*, in base al contenuto delle colonne 3 e 4 del corrispondente file di *parsing* (Dwrd.txt).

L'algoritmo funziona nel seguente modo; ponendo n=1:

1) prende in esame la prima coppia di parole, n ed n+1, della stringa di input;

2) se la coppia di parole è presente nelle prime due colonne del file di *parsing*, procede alla trasformazione e passa ad analizzare la successiva coppia di parole, n+2 ed n+3, della stringa di input (incrementa n di 2);

3) se la coppia di parole non è presente nelle prime due colonne del file di *parsing,* salta la prima parola e passa ad analizzare la coppia di parole successiva, n+1 ed n+2 (incrementa n di 1);

4) reitera il procedimento dal punto 1) finchè ci sono parole nella stringa di input.

HWRD permette la gestione di parole separate dal trattino, che possono essere sostituite da una, due o tre parole, in base al contenuto del corrispondente file di *parsing* (HWrd.txt). Per poter utilizzare questa funzione è necessario però che il trattino rientri nell'elenco dei caratteri validi, individuati nella seconda colonna del file WChr.txt.

La tabella seguente descrive il "tracciato" dei file di *parsing.*

File	Descrizione/Tracciato
trim_left.txt	1 carattere per riga
trim_right.txt	1 carattere per riga
Cchr.txt	2 caratteri separati da blank
WChr.txt	2 caratteri separati da blank
DCls.txt	2 colonne di stringhe separate da blank
DStr.txt	1 stringa per riga
rstr.txt	2 colonne di stringhe. La prima colonna deve essere lunga 50 caratteri (la lunghezza fissa è necessaria perché le stringhe possono contenere blank al loro interno)
rwrd.txt	2 o 3 colonne di stringhe separate da blank
dwrd.txt	2, 3 o 4 colonne di stringhe separate da blank
HWrd.txt	2, 3 o 4 colonne di stringhe separate da blank
IWrd.txt	1 carattere per riga
Excp.txt	1 stringa per riga
Sufx.txt	1 stringa per riga
Prfx.txt	1 stringa per riga
MChr.txt	1 carattere per riga

Per verificare la coerenza dei file di *parsing*, relativi alle funzioni RSTR, RWRD e DWRD, è stata approntata una procedura, descritta nell'Appendice B, che è stata di supporto per la codifica di variabili complesse, come l'Ateco, per le quali i file di *parsing* contengono non solo i sinonimi ma anche i criteri classificatori. Tale procedura non fa parte del pacchetto CIRCE e quindi non è richiamabile dalla GUI perché il suo utilizzo richiede un buon livello di competenze tecnico-contenutistiche.

3.4 Calcolo dei pesi

Anche per questa fase è stato implementato un algoritmo analogo a quello di ACTR v3 descritto nel volume "Tecniche e strumenti n.4 -2007". Tuttavia alcune formule e le relative descrizioni si discostano leggermente da quanto scritto nel testo citato, in quanto sono il risultato delle prove empiriche effettuate per comprendere dettagliatamente il comportamento di ACTR v3.

Nella fase di caricamento del dizionario viene creata una tabella che associa ad ogni parola presente nel dizionario un peso, secondo la formula:

$$P(W_i) = 1 - \log(NW_i)/\log(N) \qquad (1)$$

dove NW_i è il numero di codici contenenti la parola i-esima (W_i), mentre N è il numero totale di codici contenuti nel dizionario.

Nel caso in cui nel dizionario siano presenti più voci con lo stesso codice, al fine del calcolo dei pesi, queste vengono preventivamente accorpate in una sola voce contenente tutte le parole univoche riferite allo stesso codice. Questo perché, attraverso la sperimentazione, è stato possibile capire che il termine "numero dei codici" con cui sono descritti gli argomenti dei logaritmi della (1) doveva far riferimento ai soli codici univoci contenuti nel dizionario, tenendo però in considerazione tutte le parole diverse afferenti allo stesso codice.

I pesi così calcolati sono utilizzati nella fase di codifica per il calcolo del punteggio di *matching*.

3.5 Calcolo del punteggio di *matching* e risultati della codifica

Nella fase di codifica, il testo da codificare è confrontato con il dizionario alla ricerca di un abbinamento esatto (*direct match*), ossia di quella singola voce del dizionario con cui ha tutte le parole in comune. L'esito positivo della ricerca dà luogo all'assegnazione di un codice unico. Se, invece, il tentativo fallisce viene ricercato un abbinamento parziale (*indirect match*). In questo caso il *software* individua, tramite una misura della similarità tra testi (S) di tipo empirico la voce/le voci del dizionario con descrizione più simile al testo da codificare. A tal fine, vengono selezionate tutte le voci che hanno almeno una parola in comune con il testo da codificare alle quali viene assegnato un punteggio di *matching* che è funzione del numero di parole in comune e del loro grado d'informatività (peso) all'interno del dizionario di riferimento. Le voci selezionate vengono successivamente ordinate in ordine decrescente di punteggio.

La formula di calcolo del punteggio (S) è la seguente:

$$S = 10\,(a + 2b)/3 \qquad (2)$$

$$a = 2\,N_C/(N_R + N_D)$$

$$b = 2\sum_i P(W_i^C)/\,\{\sum_j P(W_J^R) + \sum_i P(W_i^D)\}$$

dove N_C è il numero di parole in comune tra il testo da codificare e la singola voce del dizionario, N_R e N_D rappresentano il numero totale di parole contenute rispettivamente nel testo da codificare e nella singola voce del dizionario, mentre $P(W_i^C)$, $P(W_J^R)$ e $P(W_i^D)$ rappresentano i pesi definiti nella (1) rispettivamente della i-esima parola in comune (W_i^C), della j-esima parola contenuta nel testo da codificare (W_J^R) e della i-esima parola della singola voce del dizionario (W_i^D) .

La misura di similarità espressa nella (2) assume valori compresi nell'intervallo [0,10] i cui estremi corrispondono ad un abbinamento testuale nullo (S=0) o ad un abbinamento esatto (S=10). La regione di accettazione per la misura di similarità è data dalle relazioni (3) ed è costruita utilizzando tre parametri soglia, Smin, Smax e ΔS, che rappresentano rispettivamente le soglie minima e massima di accettazione e la distanza minima tra il punteggio massimo (S_1) ed il successivo (S_2) attributi alle voci del dizionario con cui è stato realizzato il *match*. A seconda del dove si "collocherà" il punteggio di *matching* rispetto alla regione di accettazione, si avranno diversi risultati della codifica, suddivisibili in i codici Unici, Multipli, Possibili o Falliti. In maggior dettaglio:

Codice UNICO:	$S_1 > Smax$ e $\{[(S_1 - S_2) > \Delta S]$ oppure [non esiste S_2] oppure $[S_2 < Smax]\}$	(3a)[5]
CodicI MULTIPLI:	$S_1 > Smax$ e $(S_1 - S_2) \leq \Delta S$	(3b)
Codici POSSIBILI:	$Smin < S_1 \leq Smax$	(3c)
Casi FALLITI:	($S_1 \leq Smin$) oppure (non esiste S_1)	(3d)

Se è soddisfatta la condizione (3a) la voce del dizionario a punteggio massimo (S_1) è dichiarata "vincente", il codice che le è associato è unico e viene assegnato in modo completamente automatico al testo di input. I casi descritti nelle (3b) e (3c) necessitano, invece, di un'analisi manuale finalizzata a valutare la presenza di un codice corretto tra quelli proposti dal sistema da assegnare al testo di input.

I valori dei parametri soglia, Smin, Smax e ΔS, sono fissati dall'utente in funzione degli obiettivi di codifica. Valori alti aumentano la precisione della codifica, ossia la percentuale dei codici unici corretti, a scapito però del tasso di codifica, ossia della percentuale di codici unici assegnati. Il viceversa accade per valori

[5] Si precisa che la formalizzazione della (3a) è diversa rispetto a quella contenuta in "Tecniche e strumenti n.4 -2007" in quanto esplicita un numero maggiore di casistiche possibili comprendendo anche i casi in cui S_2 non esiste oppure ha un punteggio inferiore alla soglia minima.

bassi dei parametri di soglia. Quindi la scelta dei valori "ottimali" da attribuire ai parametri sarà il frutto di prove di codifica finalizzate ad individuare quei valori che permettano un bilanciamento tra quantità di testi codificati univocamente (*recall rate*) e qualità degli stessi (*precision rate*).

3.6 Output della codifica

La struttura ed il contenuto dell'output di una procedura di codifica dipendono, ovviamente, dal tipo di codifica effettuata:

- per la codifica interattiva il risultato viene visualizzato nella maschera utente come descritto nel paragrafo 4.3.1;
- per la codifica web viene prodotto un output in formato JSON che viene passato alla pagina chiamante e visualizzato a video (http://www.istat.it/it/strumenti/definizioni-e-classificazioni/ateco-2007)
- per la codifica batch i risultati vengono scritti nella sotto-cartella "output" del contesto e sono costituiti da un file "report.csv", che contiene una sintesi sull'esito della codifica e, se esistono, dai file "unici.csv", "multipli.csv", "possibili.csv" e "falliti.csv" che contengono le relative tipologie di match (il contenuto di detti file è descritto nel paragrafo 4.3.2).

In tutti i casi, attraverso l'impostazione dei parametri della strategia di codifica, è possibile definire quante righe visualizzare o scrivere in caso di risultati "multipli" o "possibili".

4. L'interfaccia Utente di CIRCE

Per l'utilizzo di CIRCE in ambiente pc (Windows) è stata realizzata una Graphical User Interface (GUI) che facilita notevolmente l'utente nell'utilizzo del software di codifica in quanto evita che l'esecuzione dei comandi avvenga da riga di comando. Inoltre, semplifica notevolmente l'approccio all'utilizzo di CIRCE rendendo non necessaria la conoscenza preventiva di specifici comandi R per eseguire determinate operazioni.

La GUI permette di eseguire un progetto di codifica batch, consentendo anche di modificarne uno già esistente, e di gestire il *matching* interattivo di singole stringhe da codificare rispetto ad un determinato contesto. Consente inoltre la visualizzazione, in formato pdf, del manuale utente e delle istruzioni relative alla Demo.

In particolare la GUI permette di:

1. selezionare un progetto di codifica;
2. modificare un progetto già esistente;
3. creare il contesto di codifica;
4. gestire la codifica interattiva e batch;
5. scegliere una delle strategie di codifica definite all'interno del file di progetto.

4.1 Scelta del progetto di codifica

Selezionando il pulsante "Apri" dalla prima finestra che appare quando si lancia CIRCE, (Figura 2) è possibile selezionare il file di progetto che si intende utilizzare (per dettagli sulla struttura del file consultare l'Appendice A). Ricordiamo che il progetto contiene la descrizione della struttura del file di input per la codifica batch ed i parametri delle strategie di codifica, inoltre serve ad individuare la cartella del contesto di riferimento.

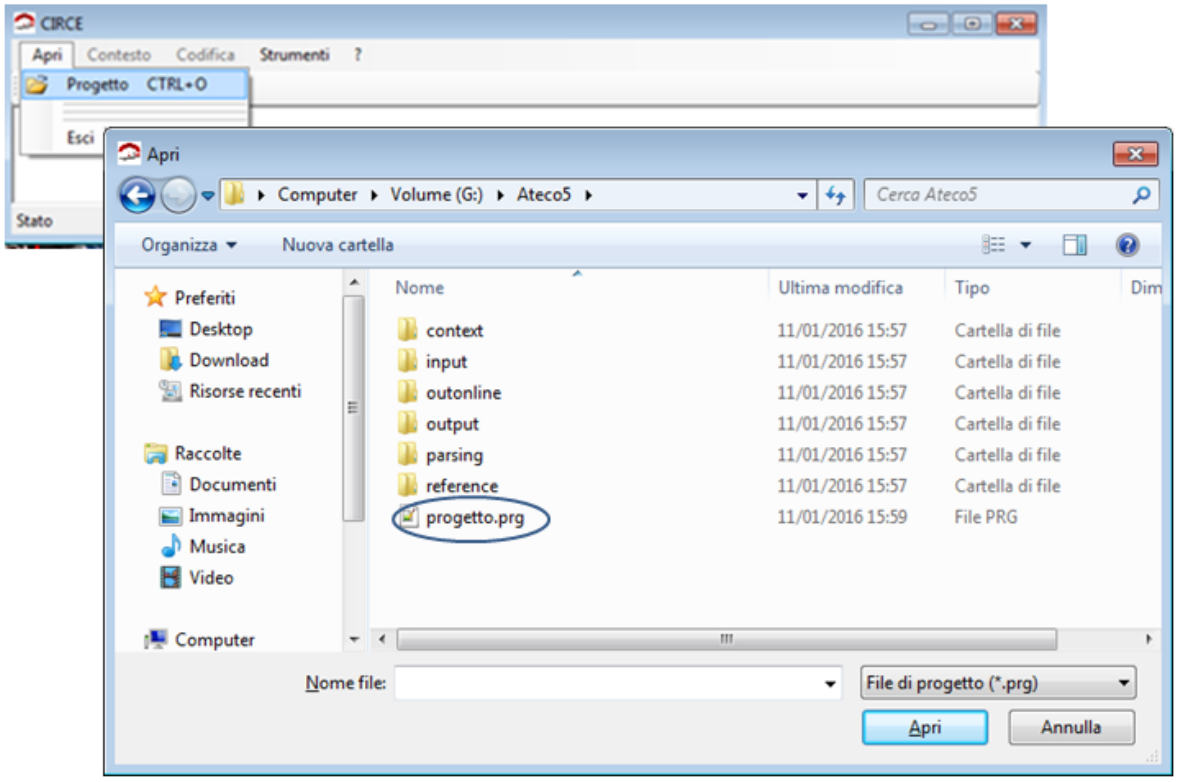

Figura 2: Scelta del progetto

Il progetto selezionato verrà visualizzato nella schermata principale della GUI come mostra la schermata successiva.

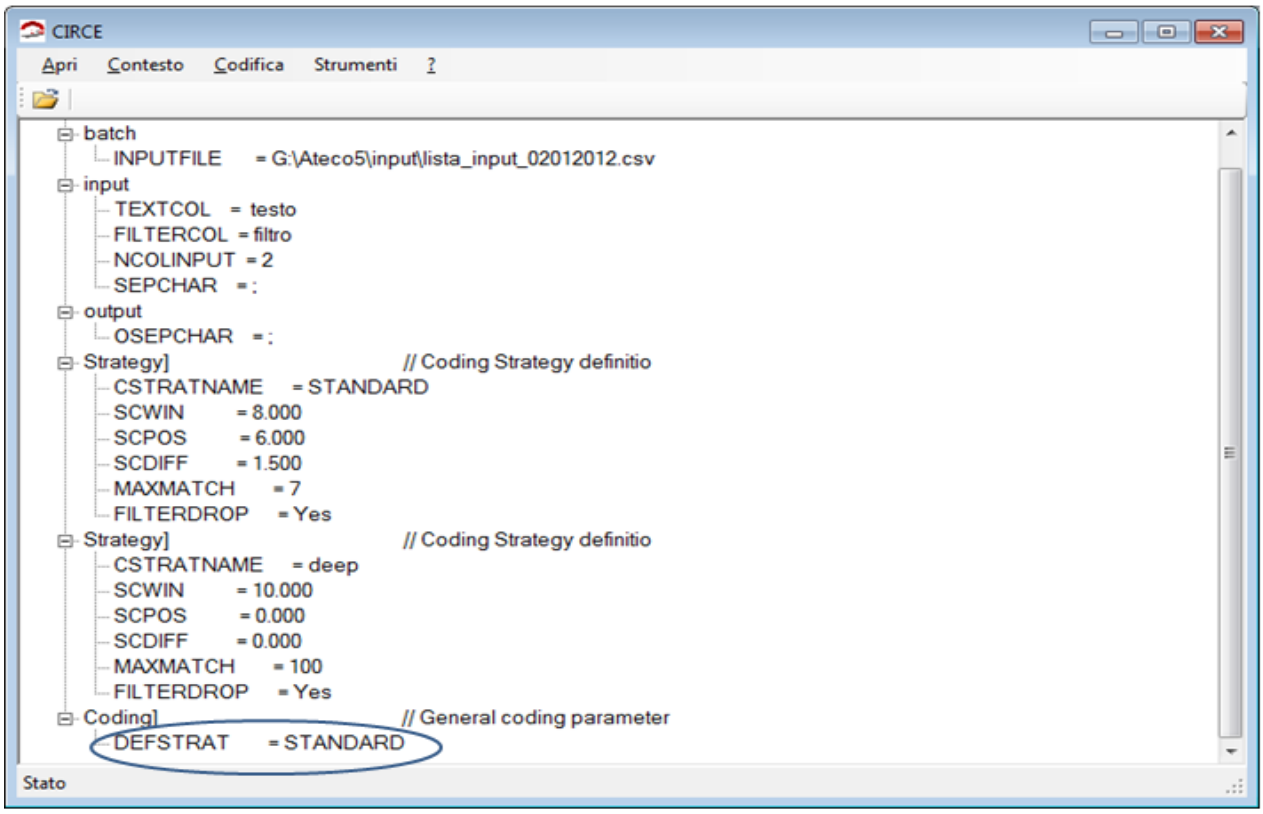

Figura 3: Contenuto del file di progetto

Come si può osservare, il progetto specifica dove si trova il file di input da codificare (INPUTFILE), la sua struttura (input) e la strategia di codifica da utilizzare (DEFSTRAT=STANDARD).

Attraverso la GUI è possibile modificare un progetto di codifica creato precedentemente. Per far questo basta “cliccare” due volte sul path del progetto selezionato, come indicato nelle figure seguenti.

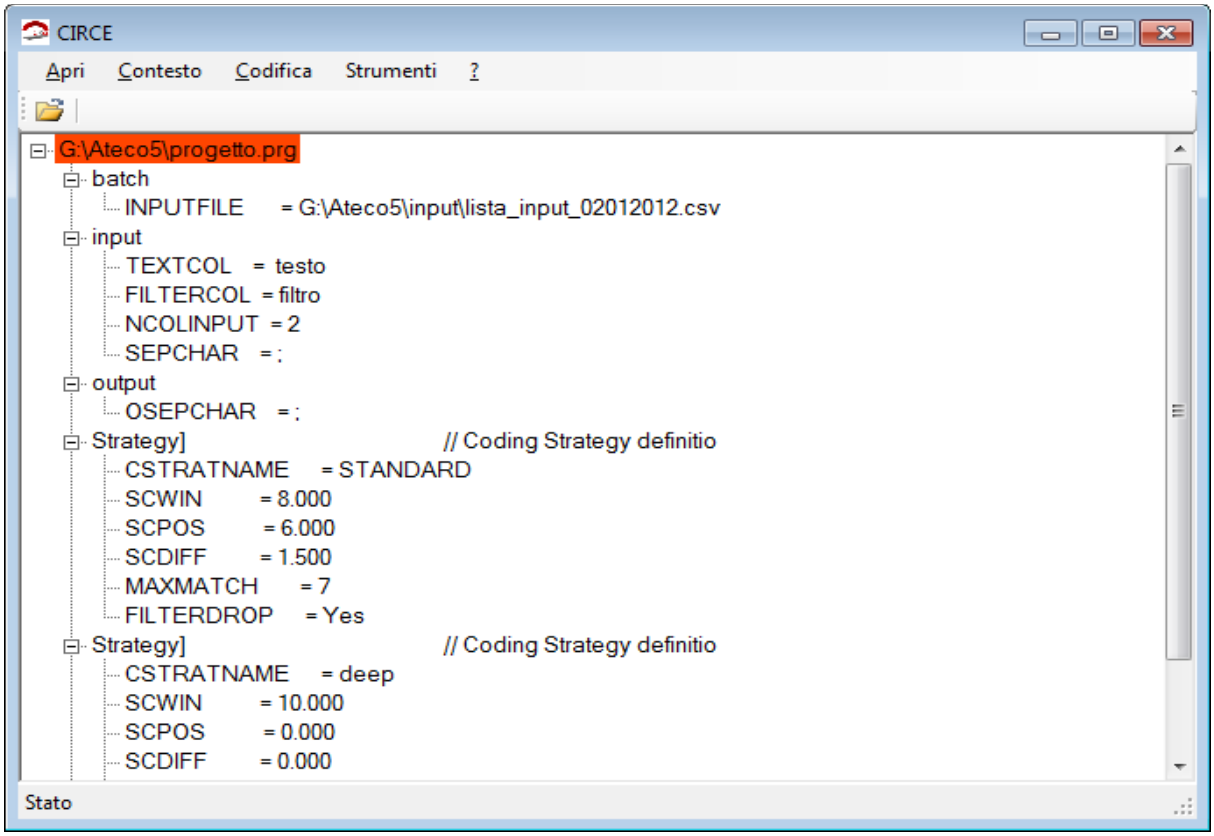

Figura 4: Progetto selezionato e da modificare.

Cliccando due volte sul path (evidenziato in rosso) si apre il file di testo contenente il progetto, mostrato di seguito.

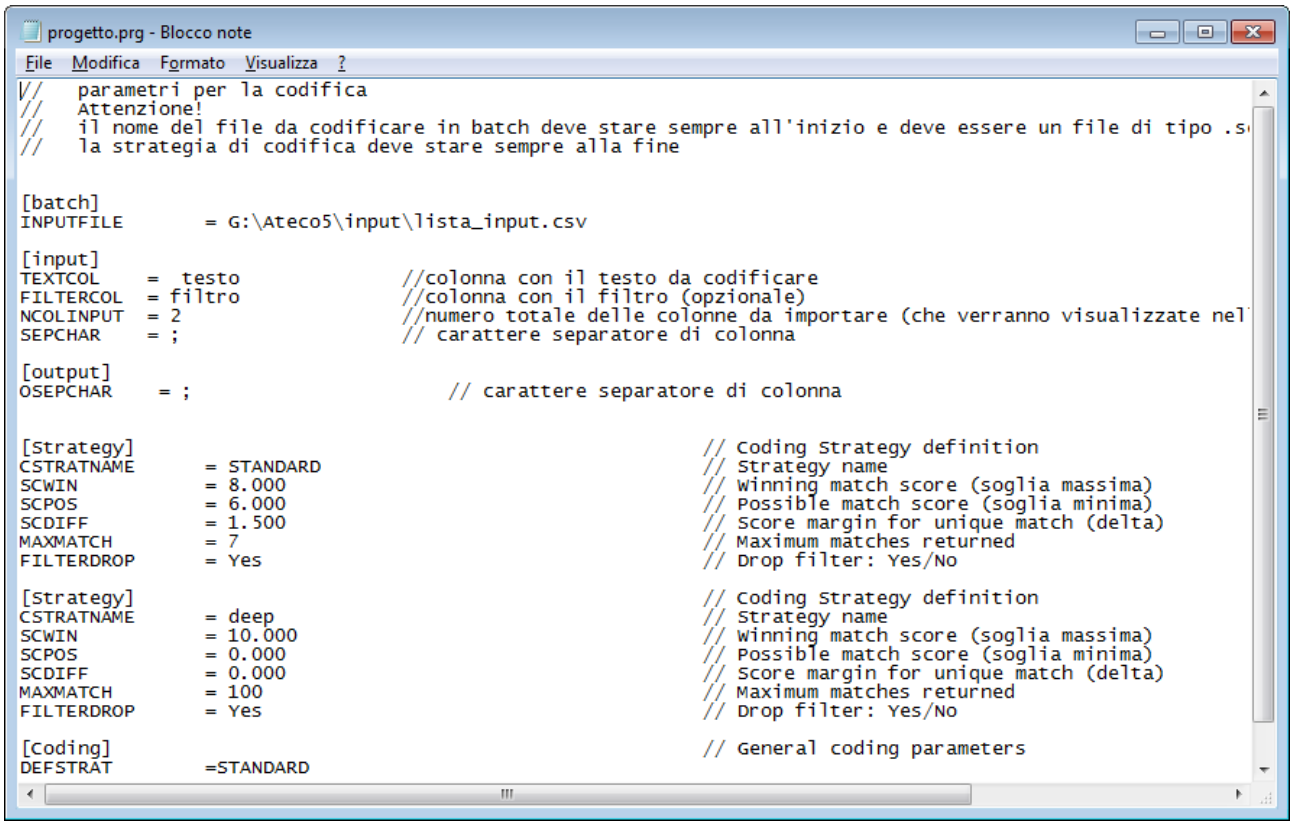

Figura 5: File di testo del progetto

4.2 Creazione di un contesto di codifica

Creare un contesto di codifica vuol dire predisporre il dizionario, contenente la classificazione di riferimento, per la fase di match. Il file di testo relativo al dizionario viene caricato nel sistema e sottoposto alla fase di standardizzazione dei testi secondo le regole definite dalla strategia di *parsing*. Questa operazione deve essere eseguita ogni volta che si vuole creare un nuovo contesto di codifica, oppure modificarne uno già esistente in caso di variazioni al dizionario oppure al *parsing*. Non è invece necessario ripetere il caricamento per effettuare la codifica di diversi file di input relativi allo stesso contesto.

Selezionando il pulsante “Contesto” e quindi “Crea” verrà eseguita una *shell dos* che permetterà di creare un nuovo contesto di codifica. I risultati di questa fase saranno racchiusi nella cartella “context”, che oltre al dizionario standardizzato ed al relativo “database” conterrà anche i file, direttamente consultabili dall’utente, dei pesi associati alle singole parole e degli eventuali duplicati (vedi paragrafo 3.2). Questi ultimi rappresentano record che possono essere duplicati effettivi (stesso codice e stessa standardizzazione) oppure incompatibilità (stessa standardizzazione ma codice diverso) che debbono essere analizzate. Qualora ci fossero duplicati effettivi e/o incompatibilità CIRCE non consente la creazione del contesto che potrà avvenire solo dopo la rimozione e/o correzione degli stessi.

Il file dei pesi delle parole (non presente in ACTR v3) permette ai responsabili del software e della classificazione di capire “l’influenza” delle singole parole sulla fase di match. Rappresenta quindi un ausilio nell’ottimizzazione delle regole di *parsing* e quindi dei risultati della codifica.

Le figure seguenti mostrano quanto appena descritto.

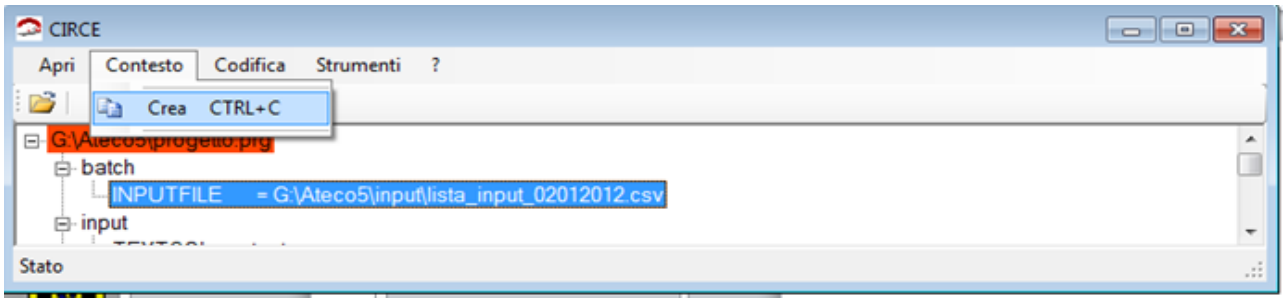

Figura 6: Creazione del contesto

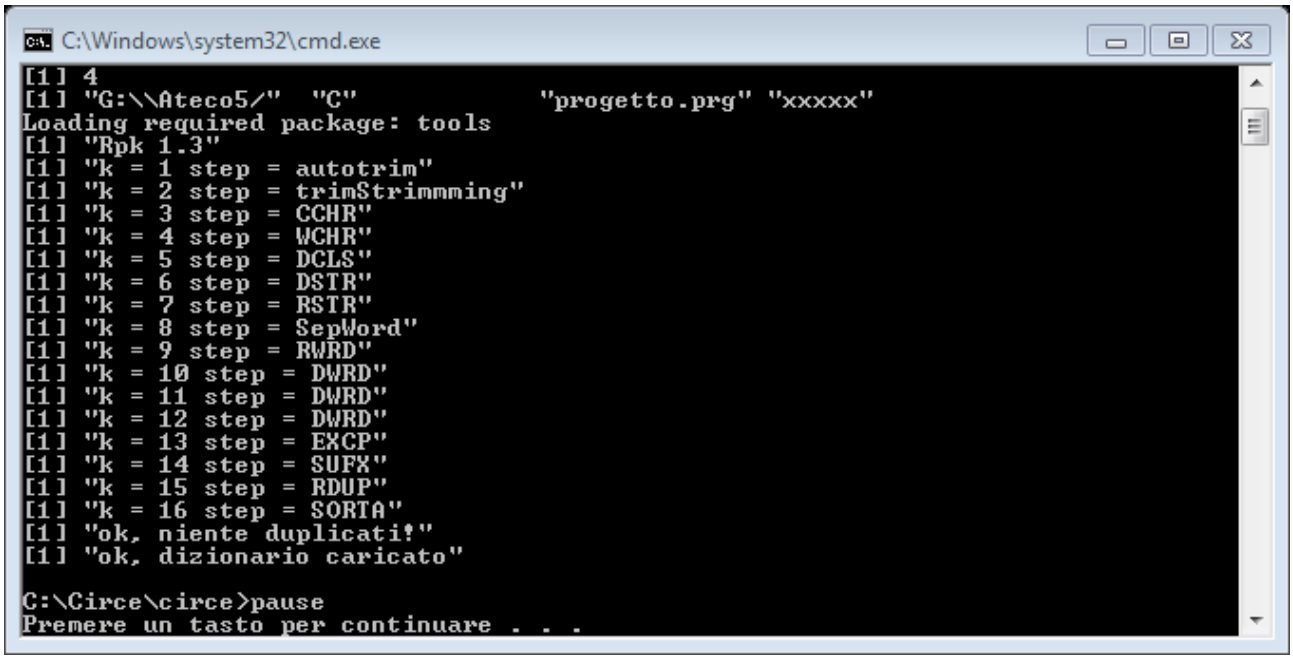

Figura 7: Shell dos di esecuzione

4.3 Codifica e definizione della strategia

Dopo aver selezionato il progetto di codifica ed eventualmente creato il contesto, è possibile, attraverso l'interfaccia, codificare testi di input secondo due modalità: codifica interattiva e codifica batch. Le maschere della GUI mostrano il tipo di strategia di codifica che si intende utilizzare ed i relativi parametri (Smax, Smin e ΔS) che l'utente può cambiare in fase di codifica interattiva per vedere come si modifica il risultato del match. Nel seguito sono descritti i passi da seguire per la codifica interattiva e per quella batch.

4.3.1 Codifica interattiva

Selezionando il pulsante "Codifica" verrà proposta la schermata (Figura 8) da cui sarà possibile eseguire la codifica secondo la modalità desiderata.

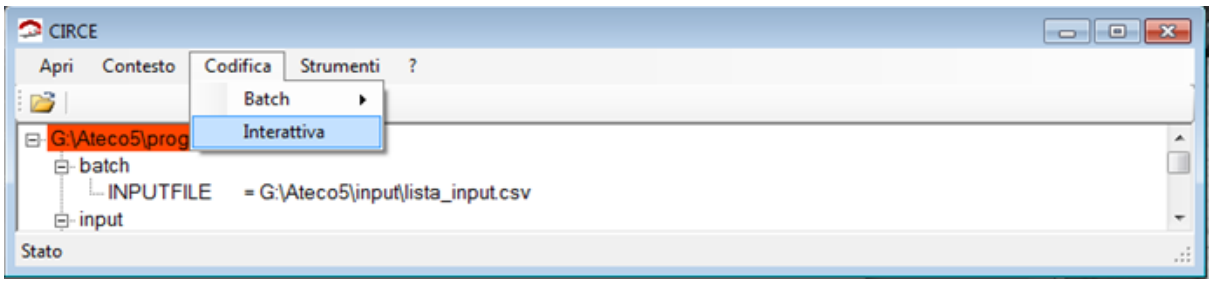

Figura 8: Selezione tipologia di codifica

Scegliendo “Interattiva” apparirà la seguente schermata, che può essere divisa in due parti, contraddistinte da A e B, come nella Figura 9.

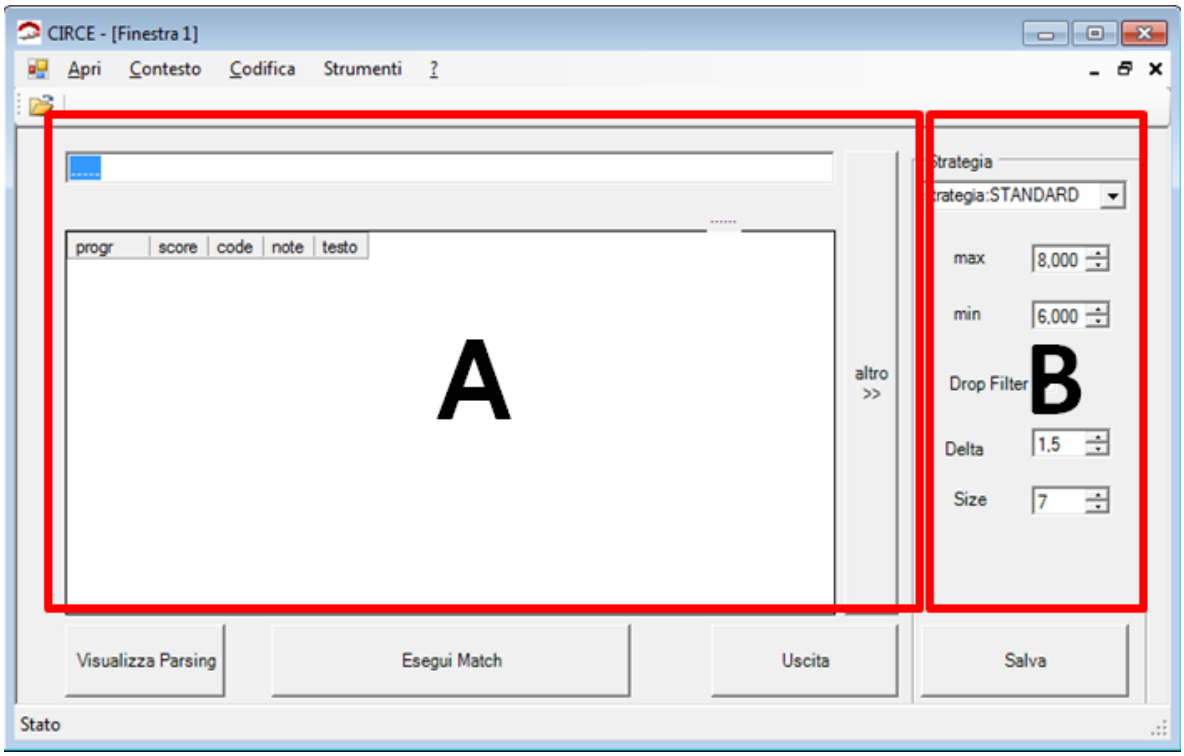

Figura 9: Schermata della codifica interattiva

Nella parte A si inserisce il testo da codificare, digitandolo nell’apposito box in alto. Nella parte B è possibile impostare i parametri relativi alla strategia di codifica, come descritto più avanti.

Selezionando il pulsante “Esegui Match”, dopo aver digitato il testo da codificare, verrà eseguita una *shell dos* che visualizzerà i diversi step previsti dalla procedura di codifica (Figura 10).

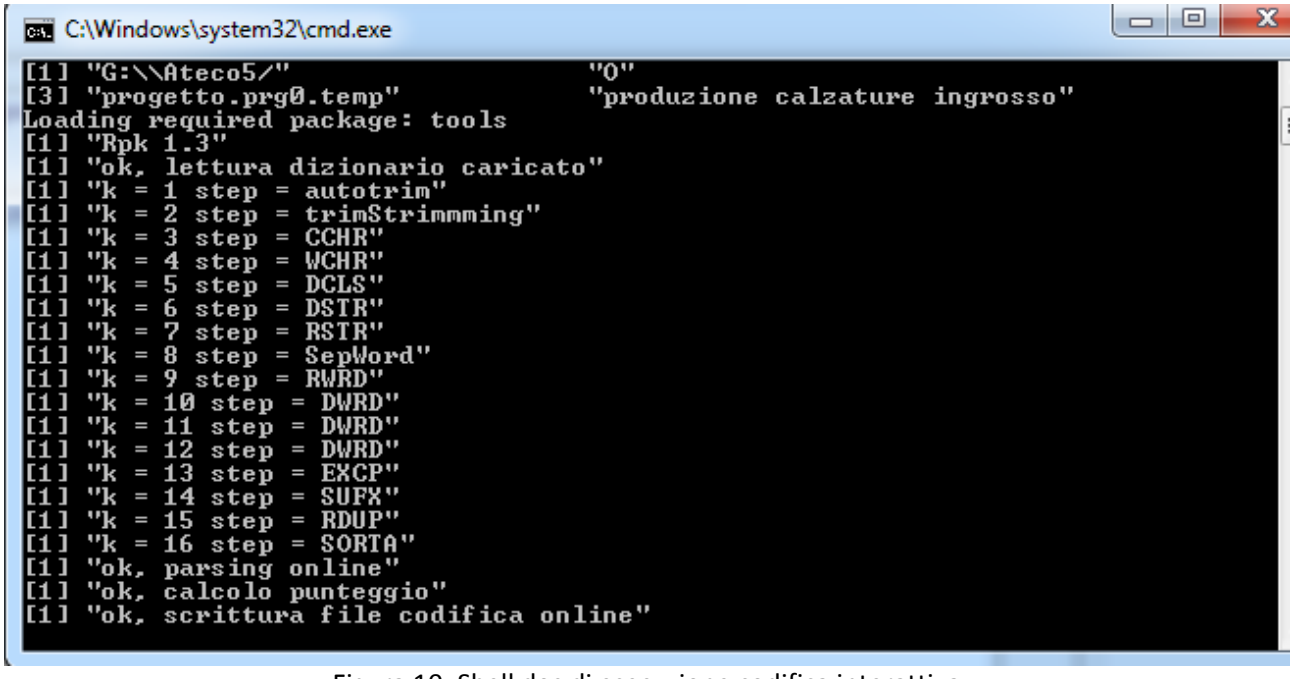

Figura 10: Shell dos di esecuzione codifica interattiva

Premendo un qualsiasi tasto verrà visualizzato il risultato del match. A titolo puramente esemplificativo, nella Figura 11 è stato riportato il risultato del match del testo «PRODUZIONE CALZATURE INGROSSO» che, come si può osservare, presenta un elevato livello di similarità con diversi testi presenti nel dizionario. Poiché i punteggi del match sono poco distanziati tra loro (la distanza fra i primi tre casi è inferiore al Delta) e compresi tra il punteggio massimo ed il minimo il risultato della codifica rientra nella casistica dei *"Multipli"* (vedi paragrafo. 3.5).

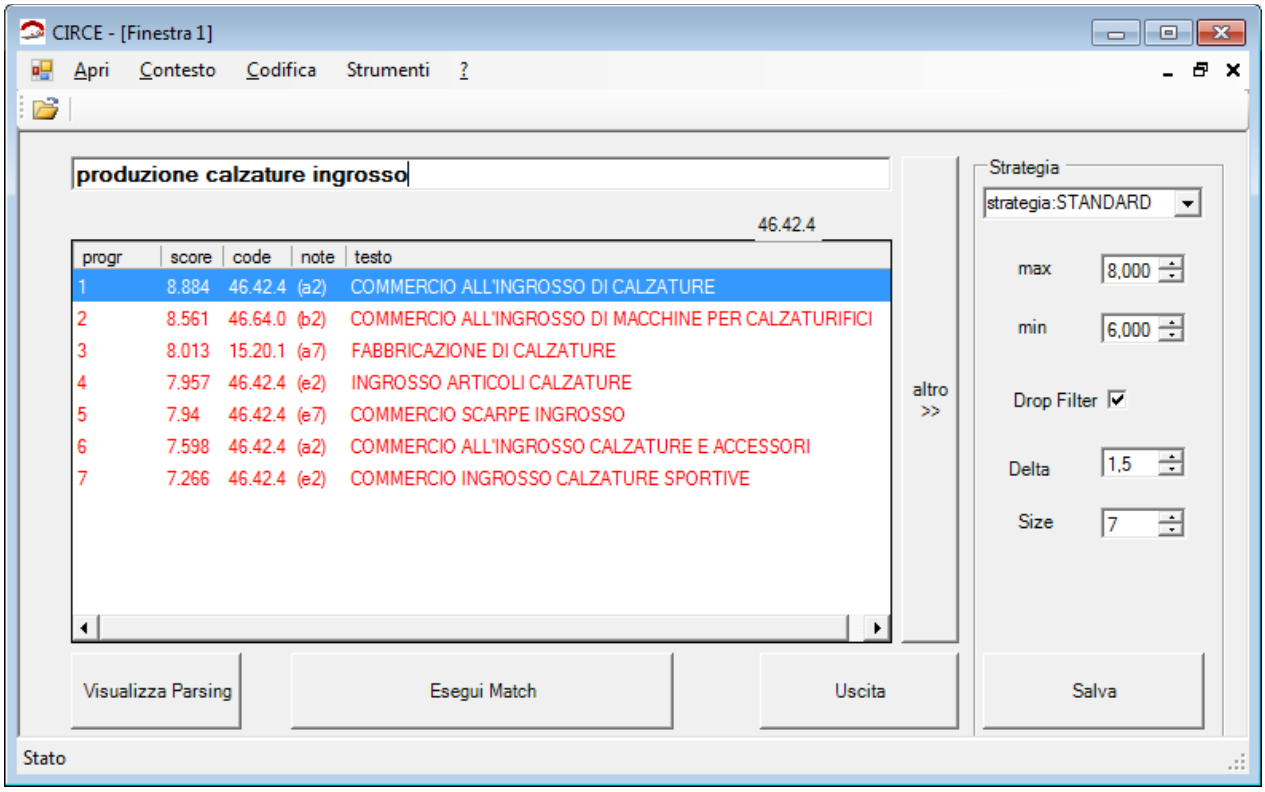

Figura 11: Risultato della codifica interattiva

È possibile verificare anche come il *parsing* abbia elaborato il testo di input (Figura 12), "cliccando" sul pulsante "Visualizza Parsing".

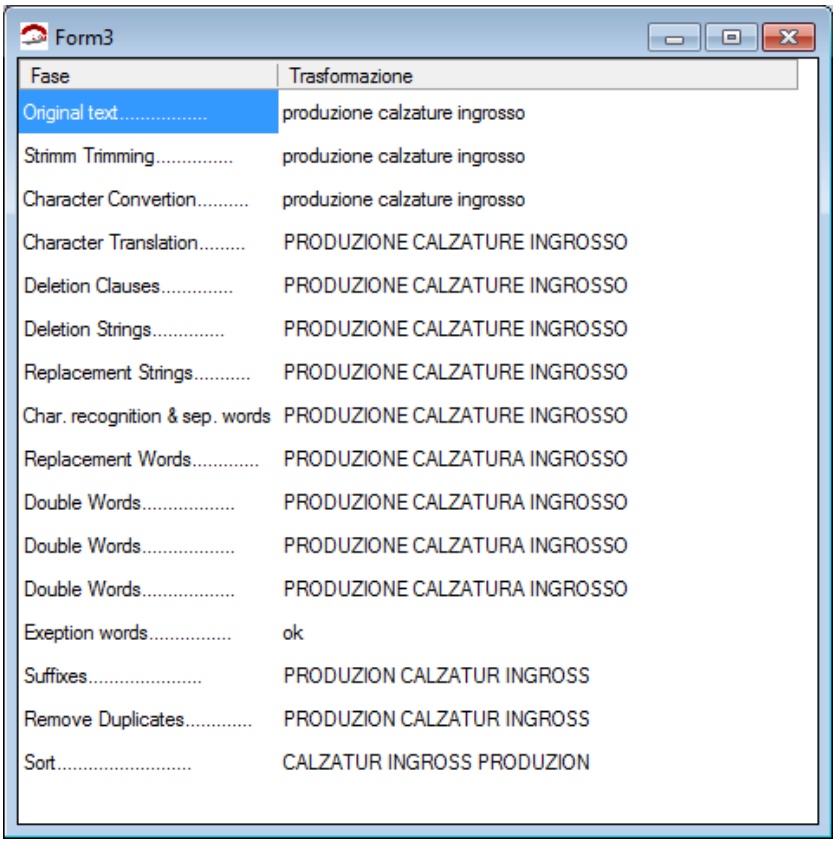

Figura 12: Risultati dei vari passi del *parsing*

Il poter visualizzare le trasformazioni che subisce il testo di input ad ogni passaggio della strategia di *parsing* è fondamentale nella fase di addestramento dell'ambiente di codifica.

Nella parte B della schermata mostrata in Figura 9 è possibile impostare una serie di parametri. Innanzi tutto si può selezionare la strategia tra quelle definite nel file di progetto; ad esempio, nel file di progetto riportato nella Figura 3, sono state definite due strategie "STANDARD" e "deep" che saranno quindi le strategie selezionabili attraverso la maschera.

Inoltre si possono modificare i valori dei parametri di soglia (Max, Min e Delta) nonché quelli relativi all'uso del filtro (Drop Filter)[6] ed al numero di risultati che si vuole siano mostrati a video (Size).

Tutti i valori proposti di default sono quelli definiti nel file di progetto. Il poter variare dinamicamente questi valori permetterà di sperimentare differenti combinazioni di parametri e strategie di codifica consentendo di verificarne subito gli effetti.

4.3.2 Codifica batch

Selezionando il pulsante "Codifica" e quindi "Batch", dopo aver selezionato la strategia, sarà possibile eseguire la codifica secondo questa modalità.

[6] La funzione che gestisce l'uso del filtro sarà implementata al più presto

Figura 13: Codifica Batch

Il file da codificare è specificato all'interno del file di progetto alla voce "INPUTFILE". Prima di eseguire la codifica batch occorre controllare che il nome del file, e il relativo percorso, siano scritti correttamente nel file di progetto.

Relativamente al file di input bisogna tenere presente che:

- Il file da codificare deve essere sempre in formato .csv**;**
- la colonna dove sono presenti le stringhe da codificare (parametro TEXTCOL del file di progetto) si deve chiamare "testo";
- è necessario definire il carattere separatore di colonna, nel parametro SEPCHAR del file di progetto. Per default nelle maschere verrà proposto il "**;**" che può eventualmente essere modificato;
- sempre nel file del progetto è necessario indicare il numero di colonne del file di input che si vogliono visualizzare nei file di output (parametro NCOLINPUT). A tal fine è importante sapere che tali colonne si contano nell'ordine in cui sono scritte, per cui quando si predispone il file da codificare bisogna fare attenzione a dove si mettono le informazioni che si vogliono visualizzare nell'output, in particolare la stringa da codificare. Le colonne così individuate occuperanno le prime posizioni nei file di output.

I file di output verranno scritti nella cartella "output" del contesto. Non è possibile personalizzarne il nome, mentre, se necessario, si può modificare il carattere separatore di colonna (parametro OSEPCHAR del file di progetto).

Prima di procedere alla codifica viene eseguito un passaggio di "pulizia" del file in cui vengono eliminati dei caratteri diversi da valori alfanumerici (o da quanto specificato tramite le funzioni speciali come descritto nel seguito). Nella funzione di "pulizia" non rientrerà il carattere separatore delle colonne (ricordiamo che il file di input è un file di tipo csv) per cui occorre specificarlo nella maschera che verrà proposta.

E' importante tenere presente che questa operazione viene eseguita sul file di input, che risulta quindi modificato nel suo contenuto. Per questo motivo prima della "pulizia", l'input viene preventivamente copiato in un file denominato con lo stesso nome, con l'estensione *.originale.* Se si eseguono più passaggi di codifica sullo stesso file, la versione "originale" sarà modificata ogni volta; è quindi consigliabile fare una copia del file di input prima di eseguire la codifica batch, per evitare una eventuale perdita di informazioni.
La "pulizia" del file può essere effettuata anche separatamente, senza eseguire la successiva codifica batch (per dettagli vedi paragrafo 4.4).

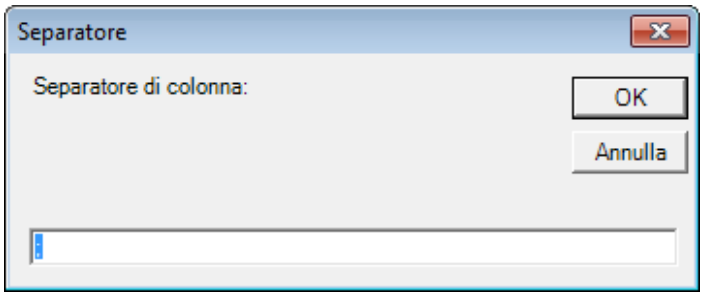

Figura 14: Scelta del separatore

Dopo questa maschera verrà proposta una *shell dos* che visualizzerà i diversi step di codifica stabiliti nella procedura di codifica batch, come mostra la figura seguente.

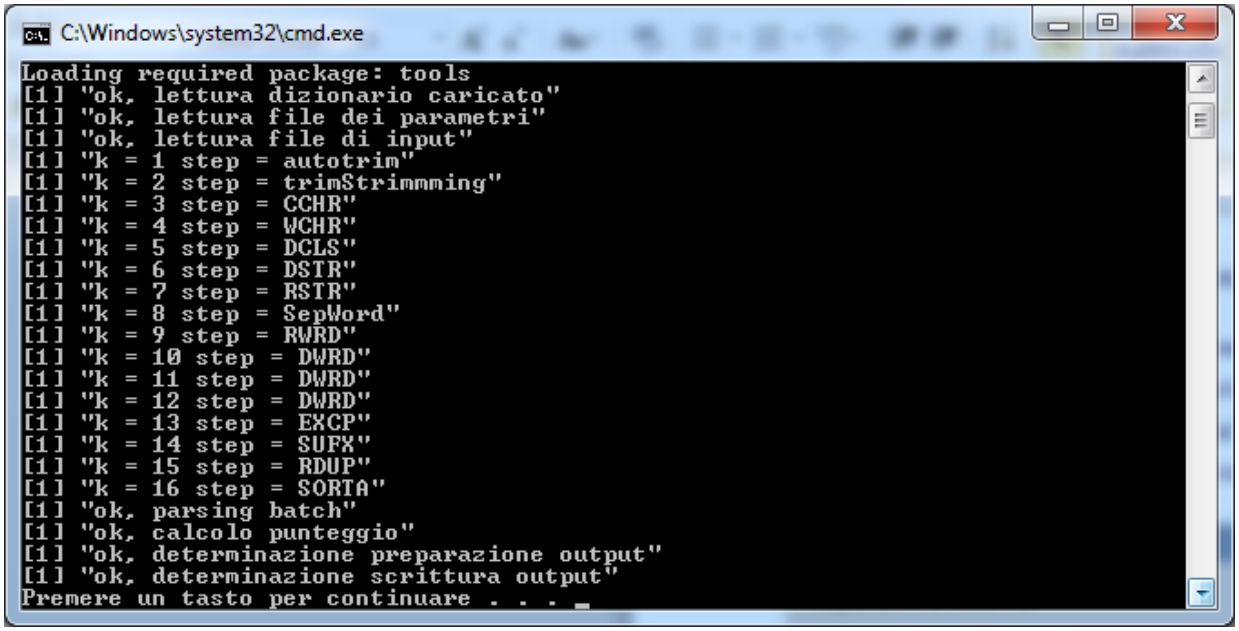

Figura 15: Passaggi codifica batch

I file di output saranno scritti nella cartella "output" del contesto, in formato csv: Unici, Multipli, Possibili e Falliti. Insieme a questi, viene prodotto un report che sintetizza i principali risultati della codifica, tra i quali citiamo il numero dei match unici diretti, ossia dei match esatti con punteggio uguale a 10, che rappresenta un indicatore di qualità della codifica (Figura 16)

Tipo	**Numero**	**%**
Unici	6690	44,49
Multipli	932	6,2
Possibili	5794	38,54
Falliti	1619	10,77
Totale	15035	100
Num. di Direct Match	5249	

Figura 16: Esempio di report di codifica

I file di output si possono personalizzare indicando nel file di progetto le prime *n* colonne del file di input che si vogliono di visualizzare nell'output (parametro NCOLINPUT). Come già detto, queste colonne verranno scritte per prime, seguite da una colonna di asterischi che separa i dati provenienti dall'input da quelli della codifica.

Le colonne generate da CIRCE nella codifica batch sono (Figure 17 e 18) : il tipo di match (col. E); il punteggio del match (col. F); il codice/i con cui è stata abbinata la stringa di input (col. G); la relativa descrizione contenuta nel dizionario (col. I); un campo "note", che può servire a visualizzare dei contenuti aggiuntivi riferiti alla singola voce del dizionario (ad esempio per la classificazione dell'Ateco è stato inserito un flag per indicare se si tratta di una voce ufficiale della classificazione oppure una voce "empirica" derivante dal linguaggio dei rispondenti (col. H);

Di seguito sono riportati, a titolo di esempio, il contenuto di un file di "Unici" e di "Multipli"

A	B	C	D	E	F	G	H	I
0	1	SERVIZI DEI CENTRI PER IL BENESSERE FISICO	*	U	10	96.04.1	(b1)	SERVIZI DEI CENTRI PER IL BENESSERE FISICO
0	2	COLTIVAZIONE DI UVA	*	U	10	01.21.0	(ep)	AZIENDA AGRICOLA PRODUZIONE DI UVA
0	7	COMMERCIO AL DETTAGLIO AMBULANTE DI PRODOTTI ITTICI	*	U	10	47.81.0	(ei)	COMMERCIO AMBULANTE PRODOTTI ITTICI
0	64	ABBIGLIAMENTO PRODUZIONE	*	U	10	14.13.1	(e*)	PRODUZIONE ABBIGLIAMENTO UOMO E DONNA

Figura 17: Esempio di contenuto file Unici.csv

A	B	C	D	E	F	G	H	I
0	14	PRESTAZIONE DI SERVIZI AMMINISTRATIVI E CONTABILI	*	M	8,5	69.20.1	(ei)	PRESTAZIONE SERVIZI CONTABILI
0	14	PRESTAZIONE DI SERVIZI AMMINISTRATIVI E CONTABILI	*	M	8,5	69.20.1	(ei)	SERVIZI AMMINISTRATIVI CONTABILI
0	15	PRESTAZIONE DI SERVIZI AMMINISTRATIVI E CONTABILI	*	M	8,5	69.20.1	(ei)	PRESTAZIONE SERVIZI CONTABILI
0	15	PRESTAZIONE DI SERVIZI AMMINISTRATIVI E CONTABILI	*	M	8,5	69.20.1	(ei)	SERVIZI AMMINISTRATIVI CONTABILI
0	60	ABBIGLIAMENTO BAMBINI	*	M	9	14.13.1	(e*)	PRODUZIONE ABITI PER BAMBINI
0	60	ABBIGLIAMENTO BAMBINI	*	M	8,4	46.42.1	(e2)	COMMERCIO ALL'INGROSSO DI ABBIGLIAMENTO PER BAMBINO

Figura 18: Esempio di contenuto file Multipli.csv

Tra gli output della codifica batch si trova anche il file dei "falliti", ossia i casi in cui il punteggio di *matching* non ha superato la soglia minima. In ACTR v3 questo file conteneva solo le stringhe di input mentre CIRCE evidenzia anche il codice che ha ottenuto il punteggio di *matching* massimo (pur non superando, ovviamente, la soglia minima). Questo tipo di output non solo è di ausilio alla fase di addestramento del sistema, ma può essere usato anche per la valutazione dei risultati della codifica nei casi in cui si mettano a confronto gli esiti della procedura automatizzata con quelli della codifica manuale.

4.4 Strumenti di supporto alla codifica

In questa parte della GUI sono riportate delle funzioni di supporto al processo di codifica il cui utilizzo non è obbligatorio, ma consigliato, in quanto permettono di eliminare delle possibili fonti di errore per un corretto processo di codifica. In particolare le funzioni offerte sono:

- pulizia del file di input per la codifica batch;
- creazione file csv dalle *query* utente dell'applicazione web ATECO;
- pulizia file di input.

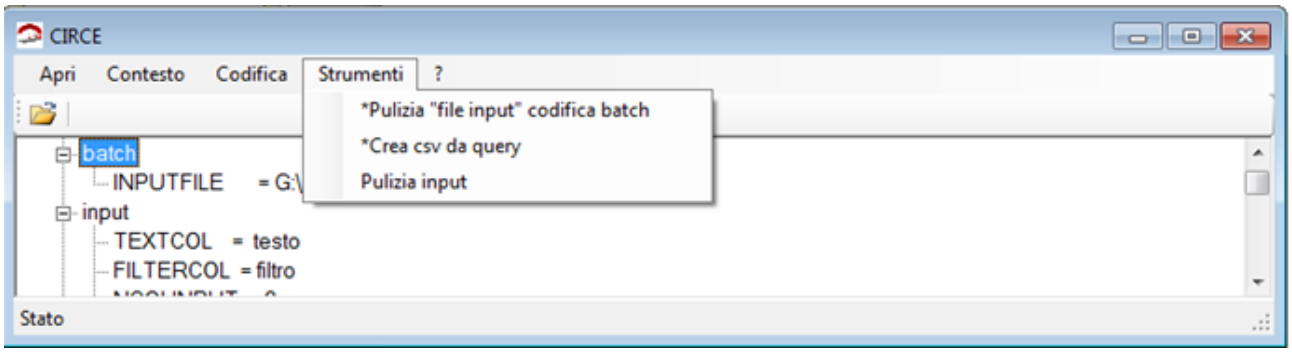

Figura 19: Strumenti di supporto

- **Funzione di pulizia del "file di input" per la codifica batch**

Questa funzione permette di poter eseguire una "pulizia" preventiva del file da sottoporre a codifica batch, ed è la stessa che viene eseguita automaticamente in fase di codifica (vedi paragrafo 4.3.2).
Dopo aver selezionato il file di input, di tipo csv, e specificato il delimitatore di colonna (la funzione proporrà il delimitatore di default di CIRCE), verranno sostituiti i caratteri accentati (definiti nelle funzioni speciali) con quelli definiti nell'espressione regolare delle funzioni speciali (paragrafo 4.5). Successivamente, tutti i caratteri alfabetici verranno trasformati in maiuscolo. La funzione riscriverà il file con le modifiche apportate, dopo aver salvato l'input originario in un nuovo file con lo stesso nome ma con suffisso ".originale".

Le figure seguenti mostrano quanto appena descritto.

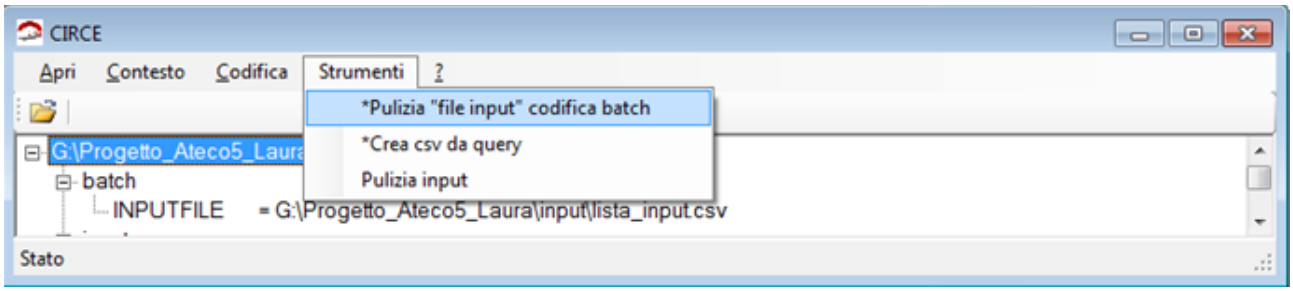

Figura 20: Pulizia "file di input" per la codifica batch

Si sceglierà il file da sottoporre a "pulizia" dalla cartella input.

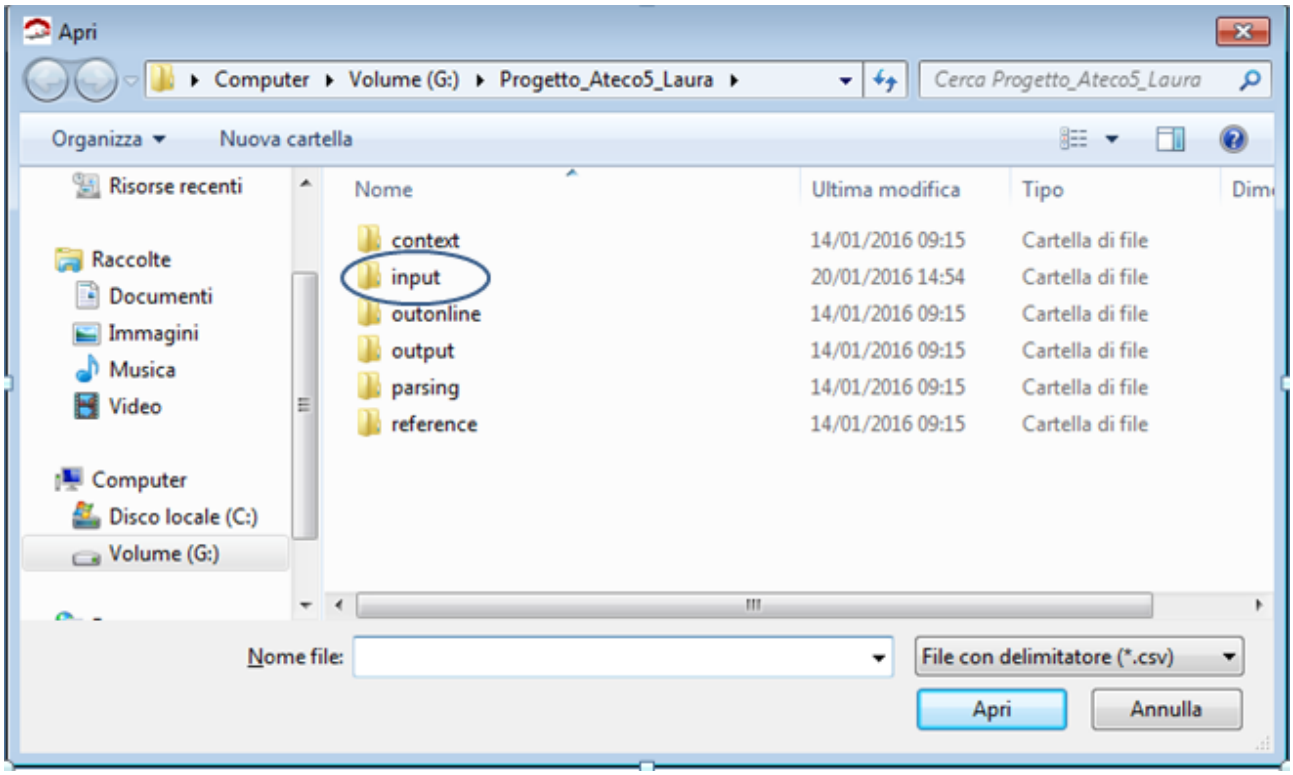

Figura 21: Selezione del file di input

Verrà proposta la schermata per la scelta del separatore.

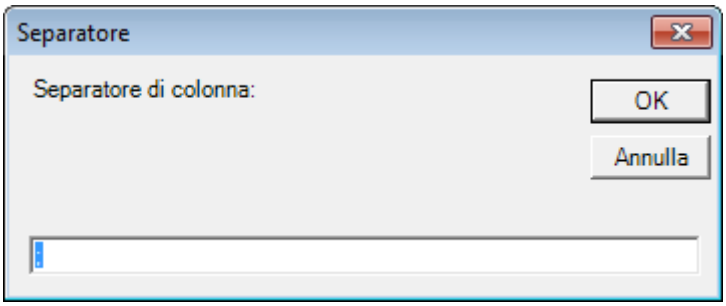

Figura 22: Indicazione del carattere separatore di colonna

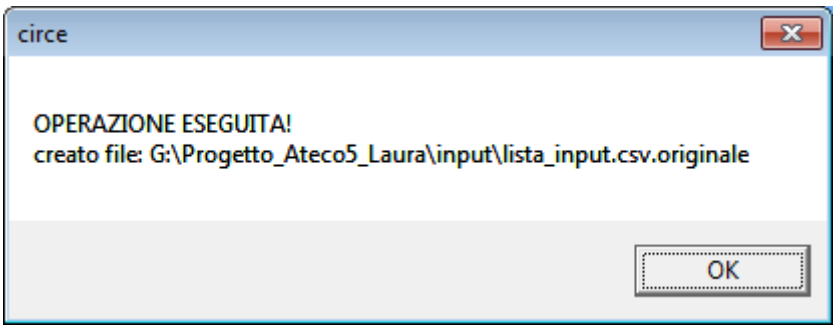

Figura 23: Risultato dell'operazione

Il file generato a seguito della funzione di pulizia è visibile nella figura sottostante

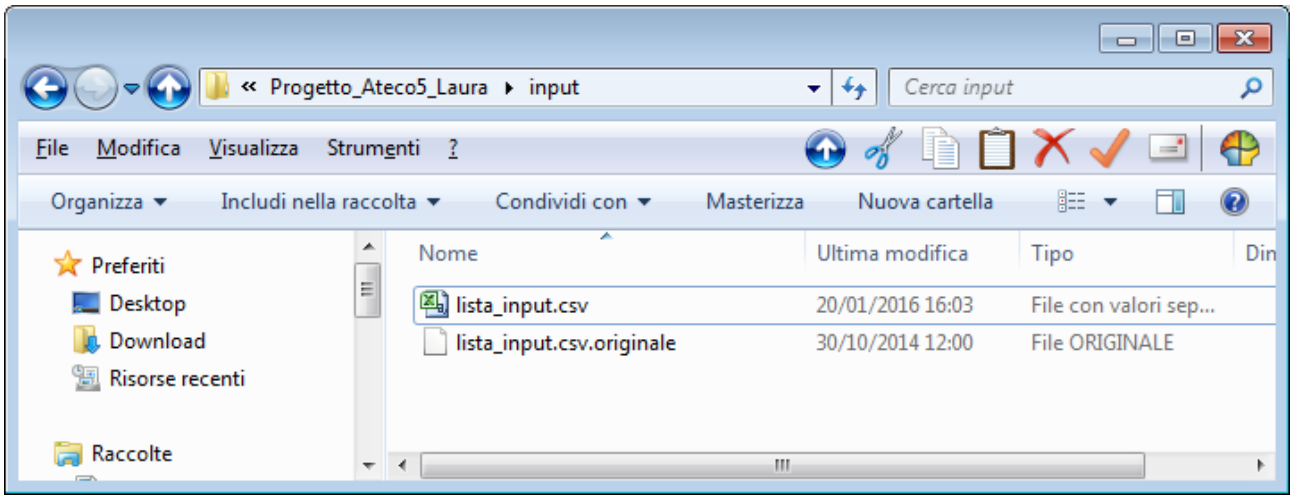

Figura 24: Visualizzazione del file generato

- **Funzione per la creazione del "csv" per i file formato testo**

La codifica batch è utilizzata in Istat per codificare le risposte testuali relative alle domande afferenti a classificazioni ufficiali rilevate nel corso delle indagini. In questi casi i file forniti dalle unità di produzione sono già in formato csv e possiedono un identificativo del record necessario all'aggancio con i dati del rispondente.

I file da sottoporre a codifica batch potrebbero però derivare da altre fonti e non disporre di una chiave identificativa. Un esempio è quello relativo alle *query* utente dell'applicazione di codifica web dell'Ateco che riporta tutte le stringhe digitate dagli utenti in un lasso di tempo settimanale. Per situazioni analoghe a questa viene messa a disposizione una funzione che trasforma il file in modo tale che sia elaborabile dalla procedura di codifica. In particolare questa funzione permette di selezionare un file di testo contenente delle semplici stringhe e di crearne uno in formato csv e con due colonne:

- la colonna del progressivo che fungerà da chiave univoca;
- la colonna testo contenente il testo da codificare.

Le figure seguenti mostrano quanto appena descritto.

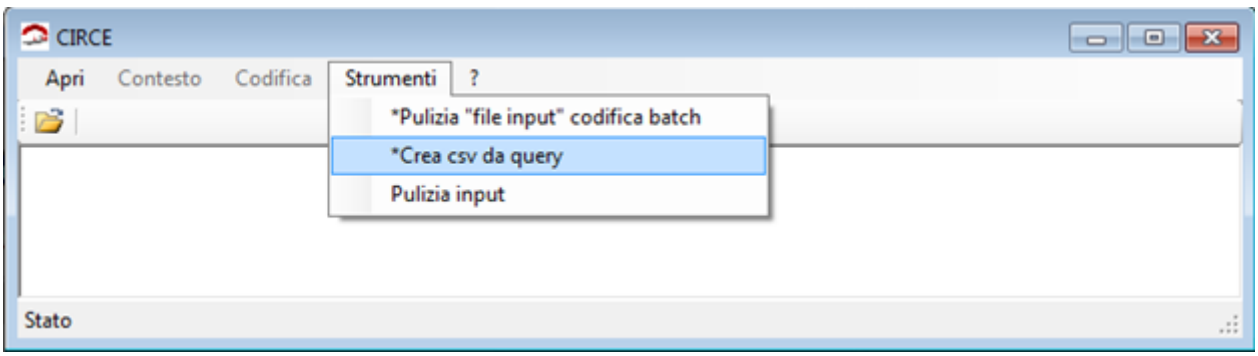

Figura 25: Crea file csv

Dopo aver selezionato il file dalla cartella di input verrà visualizzato il seguente messaggio

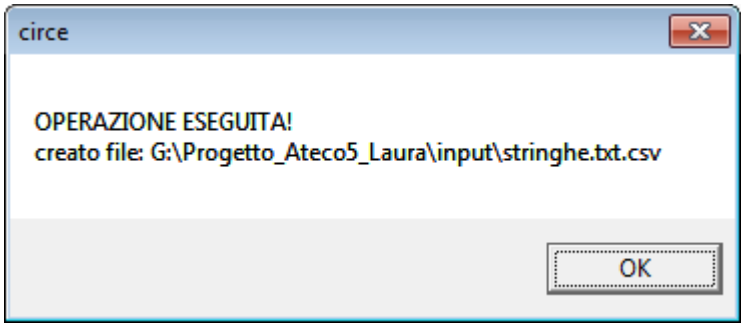

Figura 26: Risultato dell'operazione

l file generato a seguito della funzione di creazione csv è visibile nella figura sottostante

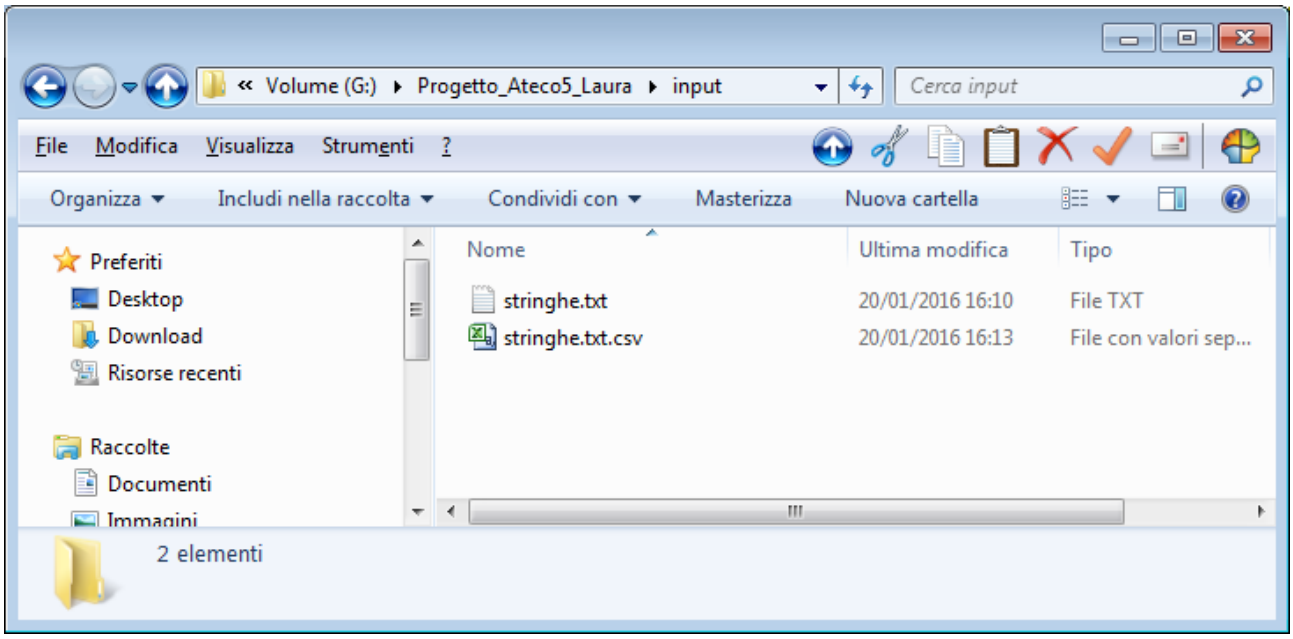

Figura 27: Visualizzazione del file generato

Attualmente il separatore di colonna per questa funzione è il carattere ";"

- **Funzione per la pulizia file di input**

Questa funzione è simile a quella denominata “pulizia file input codifica batch” con la differenza che non è possibile scegliere il separatore di colonna, che sarà obbligatoriamente il “;”, e che viene aggiunta una chiave identificativa ad ogni record. La funzione scriverà un nuovo file con lo stesso nome del file selezionato con suffisso “.csv”.

Il file selezionato verrà “pulito” utilizzando i parametri definiti nelle funzioni speciali.

Le figure seguenti mostrano quanto appena descritto.

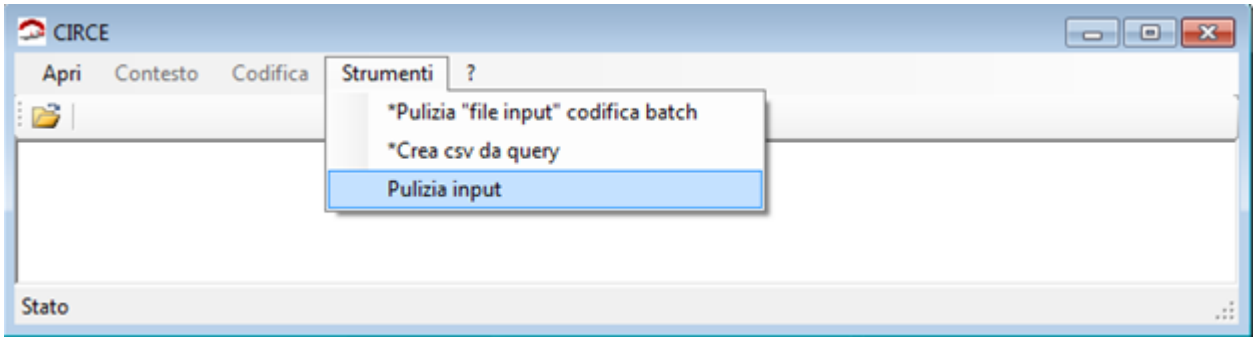

Figura 28: Pulizia file di input

Dopo aver selezionato il file dalla cartella di input verrà visualizzato il seguente messaggio

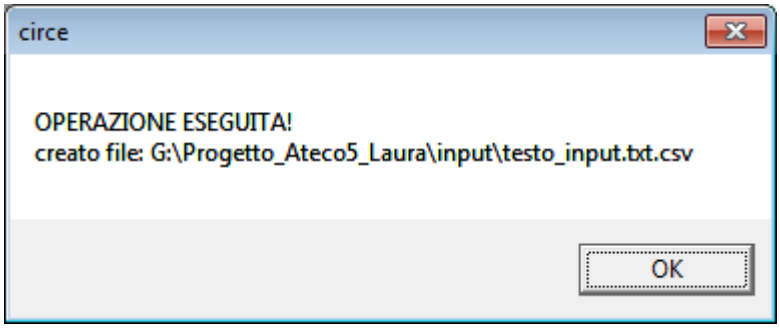

Figura 29: Risultato dell’operazione

l file generato a seguito della funzione di pulizia è visibile nella figura sottostante.

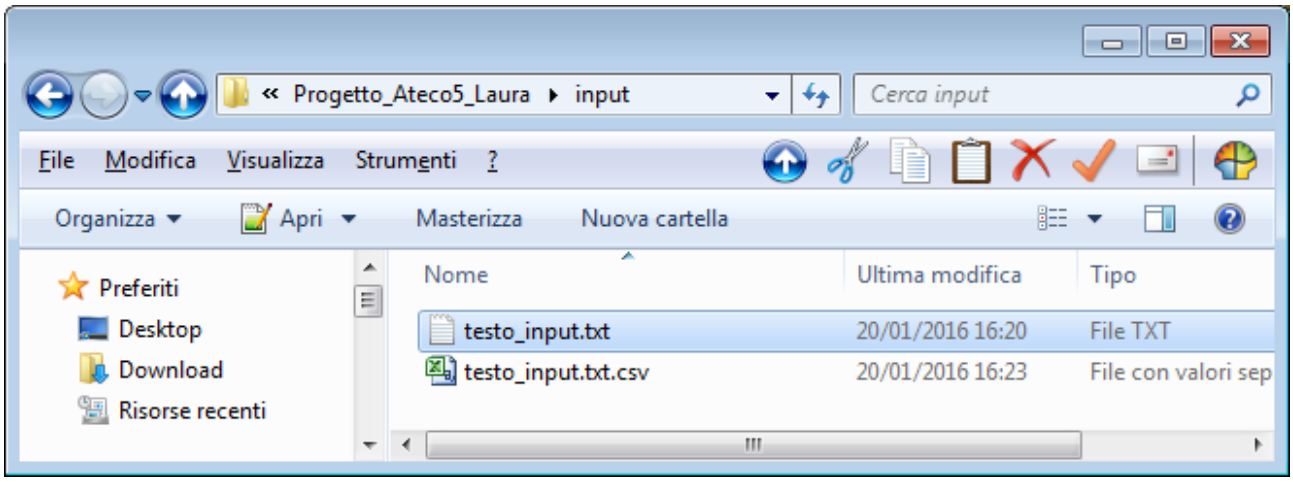

Figura 30: Visualizzazione del file generato

4.5 Funzioni speciali

Esistono delle funzioni speciali che stabiliscano i valori di default per effettuare le sostituzioni all'interno dei file da codificare e per stabilire i soli caratteri che dovranno essere lasciati all'interno dei file di input. Queste funzioni non sono al momento accessibili dalla GUI.

Nella Figura 31 sono mostrati, a titolo esemplificativo, i caratteri da sostituire: ogni carattere accentato è sostituto dal successivo carattere non accentato e maiuscolo

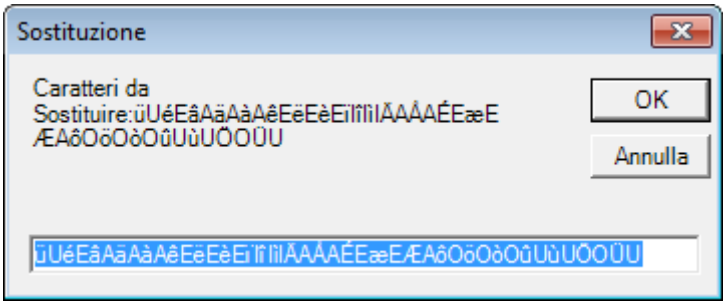

Figura 31: Caratteri accentati

Cliccando su "OK" appare un'altra schermata (Figura 32) con la visualizzazione dell'espressione regolare che indica i soli caratteri validi. Tutti gli altri verranno cancellati.

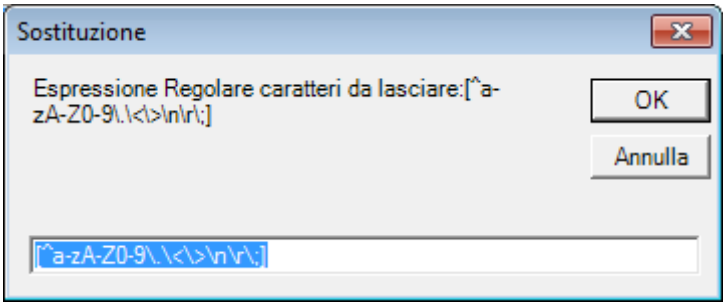

Figura 32: Caratteri da lasciare

In modo particolare occorre segnalare che l'ultimo carattere all'interno dell'espressione regolare è il separatore di colonna di default di CIRCE

5 Demo

Nell'installazione del pacchetto è prevista una demo che supporterà l'utente nella comprensione e nell'utilizzo di CIRCE.

All'interno della cartella "Demo" è stato creato un contesto di codifica che può essere utilizzato come base di partenza per la definizione di ambienti di codifica personalizzati.

Il documento "Circe_demo.pdf" descrive in dettaglio i singoli passi da seguire per realizzare un progetto di codifica batch o per effettuare codifiche interattive: dalla predisposizione del file di input, alla definizione del progetto di codifica, alla consultazione di risultati della codifica.

APPENDICE A

Il file di progetto è costituito dalle seguenti sezioni:

[batch]
INPUTFILE=nome e path assoluto del file di input che verrà codificato nella codifica batch
[Input]
TEXTCOL=nome della colonna, del file di input, contenente il testo da codificare
FILTERCOL=nome della colonna, del file di input, avente la funzione di filtro
NCOLINPUT=numero di colonne del file di input da riportare nei file di output di CIRCE (sono sequenziali partendo da sinistra)
SEPCHAR=carattere asci di separatore di colonna del file di input
[output]
OSEPCHAR=carattere asci di separatore di colonna dei files di output
[Strategy]
CSTRATNAME=nome della strategia
SCWIN= soglia massima
SCPOS= soglia minima
SCDIFF= differenziale (delta)
MAXMATCH=numero massimo di match (record), per ogni testo da codificare, che verranno scritti nei files di output, (solo per match multipli e possibili)
FILTERDROP=utilizzo o no del filtro
[strategy]
CSTRATNAME=….
SCWIN=….
SCPOS=…
SCDIFF=…
MAXMATCH=…
FILTERDROP=…
[Coding]
DEFSTRAT=nome di una delle strategie precedentemente definite all'interno del file, nei vari blocchi [strategy], da utilizzare come default

Note:

- il primo blocco [batch] indica dove si trova e come si chiama il file di input utilizzato per la codifica batch;
- il secondo blocco [input] descrive le caratteristiche del file di input;
- il terzo [output] indica che tipo di separatore si vuole utilizzare per i file contenenti i risultati della codifica.
- I blocchi successivi [Strategy], per i quali non c'è un limite prestabilito al loro numero, permettono di costruire le strategie di codifica che si vogliono utilizzare sia nella modalità batch che in quella interattiva.
- L'ultimo blocco [Coding] indica quale strategia viene utilizzata come default nella modalità batch

APPENDICE B

Software di supporto al pacchetto CIRCE per la verifica della coerenza dei file di *parsing*

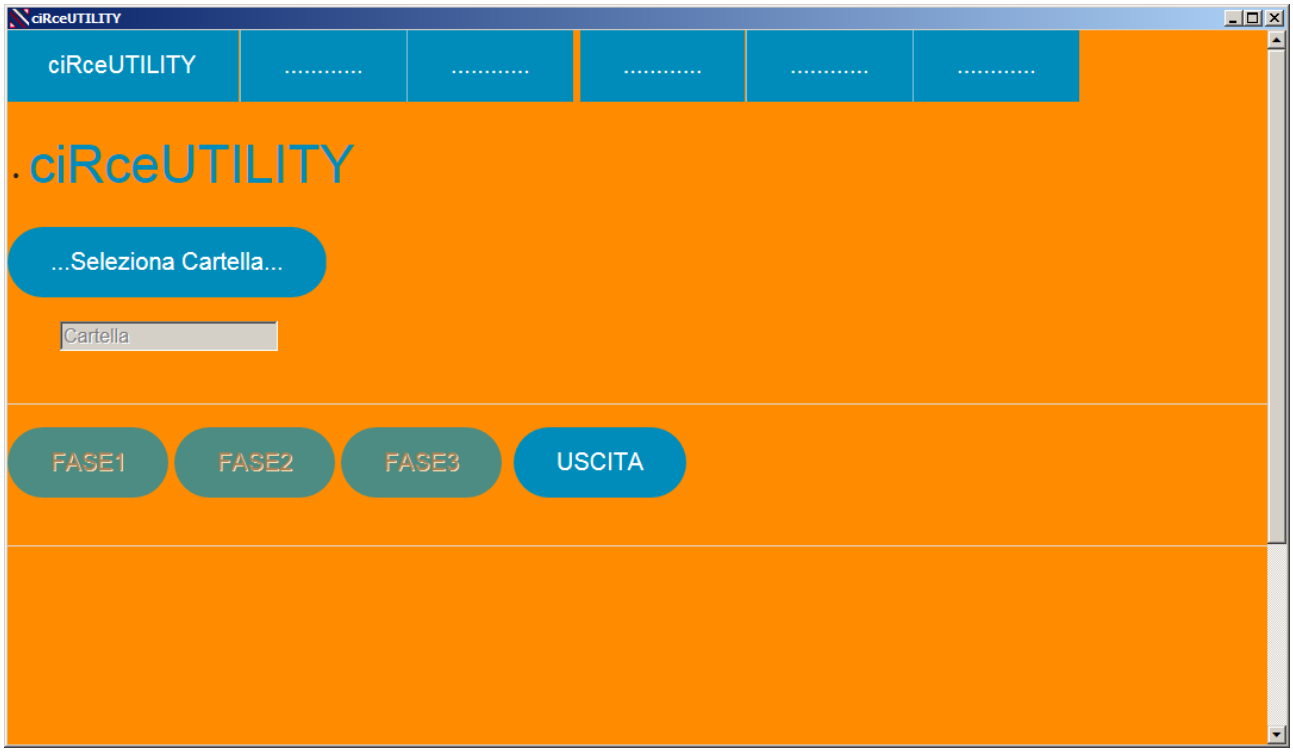

Figura 33: Utility per il controllo dei file di *parsing*

Il pacchetto mostrato in figura permette di realizzare dei controlli su alcuni file utilizzati dal *parsing*. Questo prodotto era già stato sviluppato per ACTR v3, ma è stato riscritto per adattarlo al nuovo sistema. Al momento non è richiamabile dalla GUI di CIRCE perché per il suo utilizzo è richiesto un certo livello di formazione/competenza.

La descrizione che segue relativa ai controlli sui file di *parsing* è stata estratta dal volume Istat *"Metodi e norme – n.41 2009"* a cui si rimanda per ulteriori dettagli informativi.

Il pacchetto CIRCE, al momento della creazione del contesto effettua solo un controllo finalizzato a non caricare i record "non validi", ossia quelli che, a seguito del *parsing* presentano la stessa descrizione (standardizzata). In questo caso si interrompe la creazione del contesto e viene creato il file dei duplicati, da analizzare successivamente per correggere i file di *parsing*.

Non viene effettuato alcun controllo di coerenza sui file di *parsing;* tale attività resta pertanto completamente a carico dell'utente. Poiché si è ritenuto che la gestione di questi controlli sia piuttosto pesante, considerando sia la complessità della classificazione che le dimensioni dei file dei sinonimi, si è provveduto a convertire il precedente software di supporto per ACTR, sviluppato per evidenziare eventuali incoerenze, allo scopo di lasciare la risoluzione al gestore dell'applicazione. I file sottomessi ai controlli sono:

- RSTR *Replacement String;*
- RWRD *Replacement Word;*
- DWRD *Double Word.*

I controlli sono effettuati internamente a ciascun file (un file su se stesso) e tra un file e l'altro, mantenendo come ipotesi di partenza uno *strategy file* che esegua i processi di *parsing* secondo l'ordine RSTR => RWRD => DWRD. Il software è dotato di un'apposita interfaccia che consente l'elaborazione di tre fasi di controlli.

La **prima fase,** che viene lanciata cliccando sull'apposito bottone "FASE1", effettua una serie di controlli singolarmente su ciascun file; i passaggi realizzati sono i seguenti:

1. ordinamento del file, secondo i seguenti criteri:
   RSTR => ordinamento alfabetico sulla seconda colonna
   RWRD => ordinamento alfabetico sulla seconda colonna, quindi sulla terza
   DWRD => ordinamento alfabetico sulla terza colonna, quindi sulla quarta
2. eliminazione delle righe duplicate
3. eliminazione delle righe nelle quali i termini delle colonne di destra sono uguali a quelli delle colonne di sinistra (trasformazione A => A)

La **seconda fase**, invece, è finalizzata a mettere in luce diversi tipi di incoerenze all'interno di ciascuno dei file citati ed a memorizzarle su appositi file Excel. In particolare si procede:

1. all'individuazione delle trasformazioni incoerenti, quali ad esempio:
   trasformazione di A => B
   trasformazione A => C
2. all'individuazione delle trasformazioni ricorsive, quali ad esempio:
   trasformazione di A => B
   trasformazione di B => C
   oppure
   trasformazione di A => Z B
   trasformazione di B => C D

Come può vedersi dalla Figura 30, viene richiesta l'esecuzione della procedura, relativa ai tre file di *parsing* citati, in una sequenza prestabilita; le incoerenze sono registrate in appositi file Excel. Al termine di ciascun passaggio, non soltanto è evidenziato il bottone "VISUALIZZA", che consente di editare i file Excel, ma sono valorizzati una serie di contatori, nei quali sono riportati i totali delle righe da correggere.

Con la **terza fase**, infine, vengono realizzati i controlli tra il file delle RWRD e quello delle DWRD.
In particolare si procede:

1. all'individuazione di coincidenze tra la prima colonna del file RWRD e le prime due colonne del file DWRD, ad esempio:
   trasformazione in RWRD di A => Z
   quindi trasformazione in DWRD di A B=> C D
   oppure trasformazione in DWRD di B A => C D
2. all'individuazione di coincidenze tra la prima colonna del file RWRD e le seconde due colonne del file DWRD, quali ad esempio:
   trasformazione in RWRD di A => B
   quindi trasformazione in DWRD di C D => A B
   oppure trasformazione in DWRD di C D => B A

Si fa presente, comunque, che quest'ultimo tipo di coincidenze non costituisce necessariamente un errore, quindi non è indispensabile un intervento correttivo, può essere sufficiente verificare che non si tratti di una svista, ma di una trasformazione voluta.

L'utilizzo corretto di questa applicazione implica che, ogni qual volta si effettui un intervento correttivo in un qualsiasi passaggio di ciascuna fase, sia opportuno eseguire nuovamente la procedura dall'inizio, in modo da evidenziare le effettive righe incoerenti corrispondenti all'ultima versione di ciascun file di *parsing.*