

```

# Script di controllo che calcola il numero
# totale di token presenti nel testo
{
nf = 0
while(nf < NF)
  {
  nf++
  if ($nf ~ /\#/ || $nf ~ /\@/ || $nf ~ /\%/ || $nf ~ /\$/ || $nf ~ /\$/ || $nf ~ /\$/ ||
$nf !~ /\_/)
      continue
  # omette gli elementi di markup
  counter++
  }
}
END {
print "Nel testo sono presenti " counter " token"
}

```

Tav. 70: Listato di checksum, script in GAWK.

7.5.1.5 IL "CHECKLINE". [MT] Come risulta evidente da quanto finora descritto, il sistema di controllo basato sul semplice conteggio dei token non può essere sufficiente a garantire un'individuazione rapida delle lacune all'interno del corpus disambiguato: pur avendo prova del fatto che il testo risulta incompleto e carente di un certo numero di forme, si conosce ben poco altro.

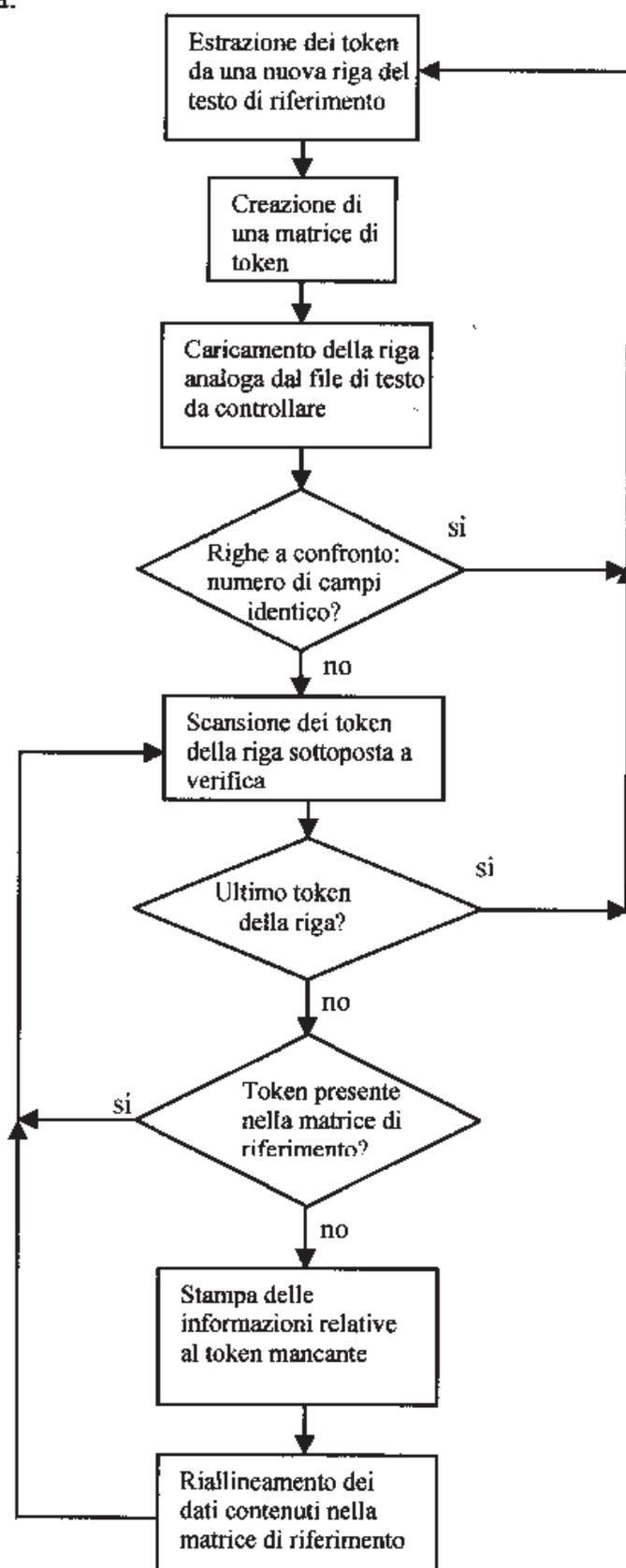
Proprio per porre rimedio a questa povertà di informazioni è stato sviluppato un ulteriore programma, denominato checkline, di cui si riporta il listato completo:

```

# Script di verifica dei token nel testo
{
count ++
nf = 0
tf = 0
campi = NF
while(nf <= campi)
  {
  nf++
  if ($nf ~ /\#/ || $nf ~ /\@/ || $nf ~ /\%/ || $nf ~ /\$/ || $nf ~ /\$/ || $nf ~ /\$/ ||
      continue
  # omette gli elementi di markup
  #
  s = split($nf, fd, "_")
  token = fd[1]
  tab[nf]=token
  }
if ((getline < "text_end") > 0)
  {
  campi2 = NF
  if (campi == campi2)
    next
  else
    {
    while(tf <= campi2)
      {

```


viduate, al fine di garantire al sistema il proseguimento regolare dell'attività di controllo fino al termine della riga.



Tav. 72: Diagramma di flusso di checkLine, script in GAWK.

7.5.1.6 IL "TRANSORD". [MT] Come accennato a inizio paragrafo, oltre agli importantissimi sistemi di controllo, nel corso dell'opera si è rivelato necessario possedere un livello