Eric Rossetto (134020), Stefano Travasci ()

[rossetto.eric@spes.uniud.it](mailto:rossetto.eric@spes.uniud.it), [travasci.stefano@spes.uniud.it](mailto:travasci.stefano@spes.uniud.it)

Relazione di progetto del corso di Informatica Medica

# Problema e requisiti

La richiesta per il progetto è la seguente:

*creare un programma di traduzione di certificati di morte per poter estrarre un sottoinsieme di certificati tradotti in maniera immediata.*

I certificati di morte fornitoci contenevano codici nel formato standard ICD-10 da tradurre attraverso un mapping predefinito in codici ICD-11. In aggiunta a ciò, vi si richiedeva di fare una statistica sull’insieme dei certificati di morte tradotti.

# Soluzione

La soluzione presentata è basata sugli oggetti e le classi definite nel package strumentiConvertiCertificati:

* **Code**: questo oggetto rappresenta una patologia secondo i formati ICD-10 e ICD-11. Tale oggetto possiede 3 campi: un codice ICD-10, il rispettivo codice ICD-11 (se presente) e il tipo di mapping che vi sussiste. Le tipologie di mapping sono: NotConvertedYet, Equivalent, SubClass, SubClassGood, NoMapping.
* **Certificate**: questo oggetto rappresenta un certificato di morte. Come tale contiene i campi per l’anno di battitura, età e sesso del deceduto e infine tutti i codici di tipo Code che rappresentano le varie cause di morte. L’ultimo codice, UCOD – Underlined Cause Of Death, ci dice quale sia stata la causa principale del decesso.
* **CodeConverter**:
* **MapNode**:
* **CertificateConverter**: tale oggetto rappresenta un convertitore di certificati, possiede un campo privato contenente il certificato e si occupa di estrarre i certificati dal file in input, delegarne la traduzione e infine aggiungere il nuovo codice tradotto.

Nel definire l’oggetto Code abbiamo fatto riferimento a delle diverse tipologie di mapping che vengono in parte definire nel file di mapping. Tali tipologie sono definite nel seguente modo:

* NotConvertedYet
* Equivalent
* SubClass
* SubClassGood
* NoMapping

# Analisi statistica

La prima statistica vede il confronto tra il numero di certificati che sono stati tradotti *correttamente*, ossia che presentano un match di un tipo **Equivalent** o **SubClassGood** o **SubClass**, e il numero di certificati che invece sono stati scartati per **NoMapping** oppure perché il codice ICD10 inserito non ha avuto una corrispondenza nel file di mapping (si presume per un qualche errore di battitura da parte del medico/operatore sanitario).

|  |  |  |
| --- | --- | --- |
| Certificati totali | 100000 | 100% sul totale |
| **Certificati tradotti correttamente** | 99788 | 99.788% sul totale |
| **Certificati scartati** | 212 | 0.212% sul totale |
| Certificati scartati per NoMapping | 185 | 87.264% sul totale dei certificati scartati |
| Certificati scartati per codice inesistente | 27 | 12.735% sul totale dei certificati scartati |

Un ulteriore dato che potrebbe risultare interessante è il numero di certificati convertiti, come prima, correttamente divisi opportunamente per tipologia di conversione.

|  |  |  |
| --- | --- | --- |
| Certificati Equivalent | 16868 | 19.903% |
| Certificati SubClassGood | 628 | 0.629% |
| Certifcati | 82292 | 82.467% |

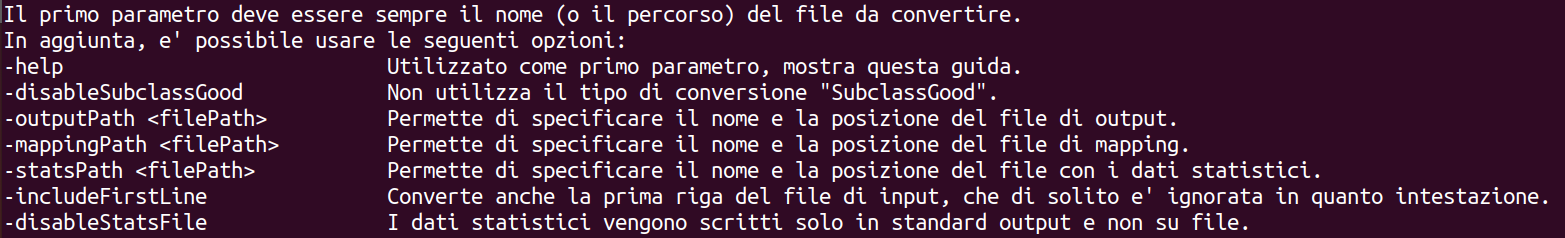
Molti dei campi all’interno dei certificati sono vuoti, è possibile quindi analizzare quale sia la dimensione media dei certificati, anche in base alla tipologia di conversione.

# Compilazione ed esecuzione del progetto

Su sistemi Unix e Unix-like è possibile compilare con il comando javac. Spostandosi sulla directory /src, digitare:

javac ConvertiCertificatiInICD11.java

In aggiunta è possibile modificare il comportamento del programma mediante l’utilizzo di alcune opzioni digitabili via linea di comando. Per aprire il menu della guida digitare:

java ConvertiCertificatiInICD11 -help

Le opzioni sono esplicative ma per completezza ne chiariamo l’uso. Oltre al comando -help, abbiamo:

* -disableSubClassGood, in riferimento a quanto detto precedentemente sui tipi di conversione, tale comando disabilita la suddivisione tra i SubClass e va a considerare una maggiore quantità di certificati, per via della più lasca relazione;
* -outputPath <filePath>, permette di specificare il nome e la posizione del file dove andranno i certificati convertiti in ICD11;
* -mappingPath <filePath>, permette di specificare il nome e la posizione del file utilizzato per il mapping;
* -statsPath <filePath>, permette di specificare il nome e la posizione del file dove andranno i risultati dell’analisi statistica effettuata;
* -includeFirstLine, permette di includere nell’output anche l’intestazione

YEAR SEX AGE P\_1\_1\_1 P\_1\_1\_2 … … UCOD

* -disableStatsFile, disabilita la scrittura dell’analisi statistica sul file e limitandosi a stamparla solo su stdout.

Per l’esecuzione del progetto è necessario conoscere la posizione del file da convertire, nello stesso formato fornitoci in cert2017100K.txt, e la posizione del file di mapping ICD10-to-ICD11:

java ConvertiCertificatiInICD11 <pathFileDaConvertire> -mappingPath <pathFileMapping>

L’esecuzione di tale commando produrrà di default, a meno di opzioni inserite dall’utente, due file il cui nome è generato proceduralmente:

1. out\_*LocalTime*.txt, quale contiene il l’insieme dei certificati di morte tradotti in ICD11;
2. stats\_*LocalTime*.txt, quale contiene tutti i dati relativi all’analisi statistica, che di default compare anche come output a terminale.