EnneLingua
Progetto di Programmazione
Giugno 2013

1 Analisi testuale mediante n-grammi

Fra le tecniche esistenti per la manipolazione e l'analisi di documenti testuali, una delle più popolari è quella basata sugli n-grammi, e verrà brevemente descritta in questa sezione.

Nel seguito, per *carattere* intendiamo un carattere alfabetico minuscolo o uno spazio; i caratteri alfabetici maiuscoli sono preventivamente convertiti in minuscoli, qualunque carattere non alfabetico viene interpretato come uno spazio, e la presenza di due o più spazi consecutivi viene considerata equivalente a un singolo spazio. Ad esempio, il testo

```
La mamma di Carlo, lo sai, ha comprato 5 mele.
```

viene considerato come se fosse:

```
la mamma di carlo lo sai ha comprato mele
```

Un *n-gramma* è una sequenza di n caratteri consecutivi che compaiono nel testo. Ad esempio, i 2-grammi che compaiono in `La mamma di Carlo, lo sai, ha comprato 5 mele.` sono indicati in Figura 1.

Indichiamo con G_n l'insieme dei possibili n-grammi: siccome un n-gramma è una sequenza di n caratteri, e i caratteri possibili sono 27 (le 26 lettere dell'alfabeto più 1 spazio), risulta $|G_n| = 27^n$.

Dato un testo T, e fissato un intero positivo n, considerate la funzione $f_T : G_n \to [0, 1]$ che associa a ogni n-gramma $g \in G_n$ la sua frequenza relativa nel testo T (cioè, il numero di volte che g compare in T, divisa per il numero complessivo di n-grammi in T). Naturalmente, per definizione,

$$\sum_{g \in G_n} f_T(g) = 1.$$

La funzione f_T viene detta *distribuzione degli n-grammi nel testo T*.

2-gramma	numero di occorrenze
la	1
a-	3
-m	2
ma	2
am	1
mm	1
-d	1
di	1
i-	2
-c	2
ca	1
ar	1
rl	1
lo	1
o-	2
-s	1
sa	1
ai	1
-h	1
ha	1
co	1
om	1
mp	1
pr	1
ra	1
at	1
to	1
me	1
el	1
le	1

Figura 1: Frequenza assoluta dei 2-grammi nel testo "La mamma di Carlo, lo sai, ha comprato 5 mele". Per leggibilità indichiamo lo spazio con un trattino. Tutti gli altri possibili 2-grammi che non compaiono nella tabella hanno frequenza 0.

Se T è abbastanza lungo e n è abbastanza grande, f_T risulterà di fatto indipendente dal testo, e dipenderà essenzialmente solo dalla lingua in cui il testo è scritto o, fra testi scritti nella stessa lingua, dallo stile di chi l'ha scritto.

La cosa è già evidente nel semplice caso $n = 1$: in tal caso f_T misura la frequenza delle singole lettere che compaiono nel testo. Ora, è abbastanza evidente che, per esempio, in un testo italiano $f_T(y)$ sarà molto piccolo (poiché la lettera "y" non compare quasi mai nella lingua italiana), mentre in un testo inglese $f_T(y)$ sarà significativa.

Un'altra funzione caratteristica è la cosiddetta *distribuzione condizionale di ordine n*: si tratta di una funzione $h_T : G_n \times C \to [0, 1]$ (qui C è l'insieme dei caratteri) che dice, per ogni n-gramma g e per ogni carattere c, quanto è frequente che l'n-gramma g sia seguito dal carattere c: in pratica, $h_T(g, c)$ è il rapporto fra il numero di occorrenze dell'$(n + 1)$-gramma gc e il numero di occorrenze dell'n-gramma g.

1.1 Applicazione 1: riconoscimento di lingue

Supponete di avere k testi $T_1, T_2, \ldots, T_k$ di ciascuno dei quali conoscete la lingua (p.es.: sapete che T_1 è un testo inglese, T_2 un testo italiano, T_3 un testo danese ecc.), e supponete di avere un testo T scritto in una lingua ignota (una delle k precedenti).

Calcolate i valori $d_1, d_2, \ldots, d_k$ dove

$$d_i = \sum_{g \in G_n} \left(f_T(g) - f_{T_i}(g)\right)^2 .$$

In pratica, d_i è la somma dei quadrati delle differenze fra il valore della distribuzione degli n-grammi in T_i e in T, sommata su tutti gli n-grammi possibili. Ci aspettiamo che d_i sia tanto più piccolo quanto più sono simili le lingue in cui T e T_i sono scritti.

Quindi, concluderemo che T è scritto nella lingua i per cui d_i è minimo.

1.2 Applicazione 2: generatore markoviano di testi

Supponete di aver analizzato un testo T e di aver stabilito la sua distribuzione di n-grammi f_T e la sua distribuzione condizionale di ordine n, h_T. Possiamo usare queste due distribuzioni per generare a caso un testo che imiti la struttura di T, nel modo seguente:

- Generiamo un n-gramma secondo la distribuzione $f_T(-)$; un modo per farlo consiste nel generare a caso un numero fra 0 e 1, diciamo α, e nel prendere in esame gli n-grammi in qualche ordine, diciamo $g_1, g_2, \ldots$ scegliendo l'unico n-gramma g_i per cui $f_T(g_1) + \ldots + f_T(g_{i-1}) < \alpha \leq f_T(g_1) + \ldots + f_T(g_i)$.
- Ad ogni passo, consideriamo gli ultimi n caratteri generati, diciamo g, e scegliamo un nuovo carattere secondo la distribuzione $h_T(g, -)$.

Sebbene qui stiamo lavorando a livello di caratteri, lo stesso si potrebbe fare a partire dalle parole (chiamando, cioè, "carattere" un'intera parola e "n-gramma" una n-pla di parole consecutive). Ecco un esempio di testo di 100 parole ottenuto mediante generazione markoviana (basata sulle parole) con $n = 2$ (cioè, ogni parola viene generata in modo casuale sulla base delle due che la precedono) a partire dalla "Divina commedia":

> così l'animo mio, ch'ancor fuggiva, si volse a lei, «verso questa rivera, tanto ch'io volsi in sù l'ardente corno, quando il dente longobardo morse la Santa Chiesa scomunicati, li quali andaro e non voglio ch'ammiri: ché chi 'l vide qua sù del mortal mondo, convien ch'ai nostri raggi si maturi». Questo conforto del foco furo; per ch'io non mi sarian chiuse le tue voglie piene ten porti che son di tiranni, e qui Marco Lombardo solve uno dubbio a Dante.] Buio d'inferno e di salire al ciel porte, orando a l'alto Sire, in tanta futa quanto sofferser l'ossa sanza polpe. Poscia

2 Il programma

Si chiede di scrivere un insieme di classi per l'analisi testuale basata sugli n-grammi e che implementi in una qualche forma l'una, l'altra o entrambe le applicazioni proposte. Si lascia agli studenti la più ampia libertà sulle scelte progettuali e implementative, ma è tassativo che il programma non faccia assunzioni di alcun tipo né sulla lunghezza dei file di input (cioè, sulla lunghezza del testo o dei testi che il programma usa come esempio) né sulla lunghezza dei file di output. Si può, invece, assumere che il valore di n sia piccolo (p.es., non superiore a 4 o 5).

Parte del progetto consiste in ogni caso nello spiegare in dettaglio le scelte compiute, e le motivazioni dietro a tali scelte.

Sul sito sono forniti alcuni file di esempio, scritti in varie lingue, con cui potete provare i vostri programmi.

3 Modalità di consegna

Il programma:

- dovrà essere scritto in Java standard;
- dovrà essere opportunamente commentato;
- dovrà essere accompagnato da un documento (PDF) di descrizione del programma per l'utente finale;
- dovrà essere accompagnato da un documento (PDF) di descrizione del programma per il programmatore.