La rete come corpus

Marco Baroni

12 novembre 2004

1 La rete come corpus

- Internet è una gigantesca rete di computer connessi che mettono a disposizione, attraverso vari protocolli, database di documenti.
- Questi documenti sono in larga parte serviti al pubblico in html (ossia, come abbiamo visto, in formato testo).
- Quanto testo c'è in rete? Secondo le stime (in numero di parole) di Kilgarriff e Grefenstette (Introduction to the Special Issue on the Web as Corpus, *Computational Linguistics* 29, 2003):

  inglese 76,598,718,000

  tedesco 7,035,850,000

  francese 3,836,874,000

  spagnolo 2,658,631,000

  italiano 1,845,026,000

  finlandese 326,379,000

  esperanto 57,154,000

  latino 55,943,000

  basco 55,340,000

  albanese 10,332,000

- E sono stime ormai vecchie e conservative...
- Sempre più studi dimostrano utilità di corpora basati sulla rete.
- Per esempio: Keller & Lapata, Using the Web to Obtain Frequencies for Unseen Bigrams, *Computational Linguistics* 29, 2003.
- Anche alla SSLMIT, è ormai prassi comune usare la rete come materia prima per la creazione di corpora (e.g., per tesi a carattere terminologico).

2 Automatizzare la navigazione

- Browsers quali Mozilla/Firefox o Internet Explorer espletano principalmente due funzioni:
  - Connettersi a un server e scaricarne il documento richiesto.
  - Visualizzare tale documento secondo le istruzioni specificate dall'html, mostrando la formattazione, i links ipertestuali, le immagini, ecc.
- Se vogliamo estrarre un corpus dalla rete, la funzione di visualizzazione non ci interessa, e scaricare a mano attraverso il browser tutti i documenti potenzialmente interessanti è un processo lento e noioso.
- Per fortuna, esistono tools che ci permettono di ottenere materiale dalla rete in modalità non interattiva: per es., scaricando tutte le pagine di un sito in un sol colpo, seguendo links automaticamente, eseguendo ricerche automatiche su Google, ecc.

3 `wget`

- `wget` è un tool utile soprattutto quando sappiamo già che pagine vogliamo scaricare e/o vogliamo scaricare tutte le pagine da un certo sito.
- Il caso più semplice:

```
$ wget http://sslmit.unibo.it/~baroni/index.html
```

- Qualcosa di più ambizioso:

```
$ wget -P first_webdata_dir -r -nd \
-R pdf,doc,gz,ps,jpeg,png,php,gif,jpg,iso http://sslmit.unibo.it/~baroni
```

- Il man di `wget` è molto chiaro (soprattutto la sezione Examples).
- Un'opzione interessante non usata qui sopra è `-i`, per passere come input una lista di *url* (indirizzi web).
- Un'opzione interessante e pericolosa è `-H`, che permette di oltrepassare i confini di un singolo host quando si seguono links ricorsivamente.

3.1 Cosa ce ne facciamo dell'output di `wget`?

- Semplice:

```
$ ls first_webdata_dir/* | simple_format_pages.pl > web_corpus_kind_of.txt
```

- Che cosa c'è dentro a *web_corpus_kind_of.txt*?

- `simple_format_pages.pl` è un"wrapper" che applica il comando `lynx -dump -nolist -force_html` a tutti i files nella lista che riceve come input.
- `lynx` è un browser per la riga di comando: provate per esempio:

  ```
  $ lynx http://sslmit.unibo.it/~baroni
  ```

- Con l'opzione `-dump`, `lynx` trasforma l'html in testo ben formattato e senza html tags, e lo manda all'output.
- Provate:

  ```
  $ lynx -dump http://sslmit.unibo.it/~baroni | more
  ```

- A cosa serve `-nolist`?

3.2 Robots, spiders, apprendisti stregoni

- Con l'opzione `-H`, `wget` si trasforma in uno spider/robot che naviga la rete da link a link, potenzialmente all'infinito.
- Tutto ciò è molto affascinante, ma un po' pericoloso:
  - Da un lato, la mole di dati scaricati può crescere in maniera spaventosa in tempi molto brevi.
  - Dall'altra, i robot/spiders non sono sempre ben visti dai webmasters, soprattutto se scaricano moli irragionevoli di dati e/o si incantano.
- Onde non fare gli apprendisti stregoni, usate il buon senso e cercate di capire bene come funzionano le opzioni di `wget`.
- Naturalmente, questo si applica anche allo scaricamento in automatico da una lista ristretta di siti, soprattutto se si tratta di siti di grandi dimensioni.

3.3 Pro e contro di `wget`

- `wget` è estremamente utile quando abbiamo già identificato una lista di siti da cui vogliamo scaricare tutte le pagine.
- `wget` (combinato con `lynx`) ci permette di creare un corpus costituito da tutto il testo trovato in tutti i siti di interesse in pochi comandi.
- Inoltre, con i dovuti accorgimenti e con risorse adeguate, `wget -H` può essere utilizzato per creare un corpus di dimensioni enormi in tempi abbastanza brevi.
- Tuttavia, `wget` non è particolarmente utile quando non abbiamo già delle *url* da cui partire.

- Inoltre, abbiamo pochissimo controllo sulla natura dei dati scaricati.
- Il metodo delle ricerche automatizzate su Google che sto per introdurre permette in generale di costruire corpora di dimensioni più piccole, ma più mirati.
- Ovviamente, nulla ci vieterebbe di usare le *url* trovate con ricerche automatizzate su Google come input per `wget`...

4 I BootCaT Tools

- `http://sslmit.unibo.it/~baroni/bootcat.html`
- M. Baroni e S. Bernardini. BootCaT: Bootstrapping Corpora and Terms from the Web. *LREC 2004*.
- Algoritmo ispirato in parte da: R. Ghani, R. Jones, and D. Mladenic. Mining the web to create minority language corpora. *CIKM 2001*.
- "Web mining" applicato alla creazione di corpora e liste di termini di linguaggi specialistici.
- L'idea di base:
  - a) Comincia con un piccolo insieme di "seeds" (parole che sembrano tipiche del dominio d'interesse).
  - b) Combina i seeds a caso in una serie di tuple (triplette, coppie...).
  - c) Usa tali combinazioni come queries in una serie di ricerche su Google, e scarica le prime N pagine trovate.
  - d) Estrai un nuovo insieme di seeds dal corpus così costruito.
  - e) Torna a b) e ripeti *ad libitum*.
- In una fase successiva, la versione finale del corpus così estratto può venire usata per estrarre termini compositi.
- La selezione di nuovi seeds si basa su paragone di frequenza (relativa) di parole in corpus specialistico e corpus "di riferimento": parole con frequenza relativa notevolmente superiore in corpus specialistico probabilmente sono caratteristiche di linguaggio specialistico analizzato.
- Interfaccia a Google tramite Google APIs (`http://www.google.com/apis/`)