Università degli studi di Perugia

Facoltà di Ingegneria

Tesi di Laurea in

Ingegneria Informatica e dell'Automazione

Generazione automatica di Word Cloud dinamiche

Relatore

Prof. Carla Binucci

Co-relatore

Prof. Walter Didimo

Candidato

Enrico Spataro

Anno Accademico 2014/2015

EVENTUALE DEDICA

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Capitolo 2

Word cloud statiche

Questo capitolo consiste in una breve introduzione al concetto di word cloud.

Nel paragrafo 2.1 verranno introdotte alcune definizioni e si parlerà brevemente di qualche applicazione, mentre nel paragrafo 2.2 si farà il punto sullo stato dell'arte riguardo le word cloud semantiche.

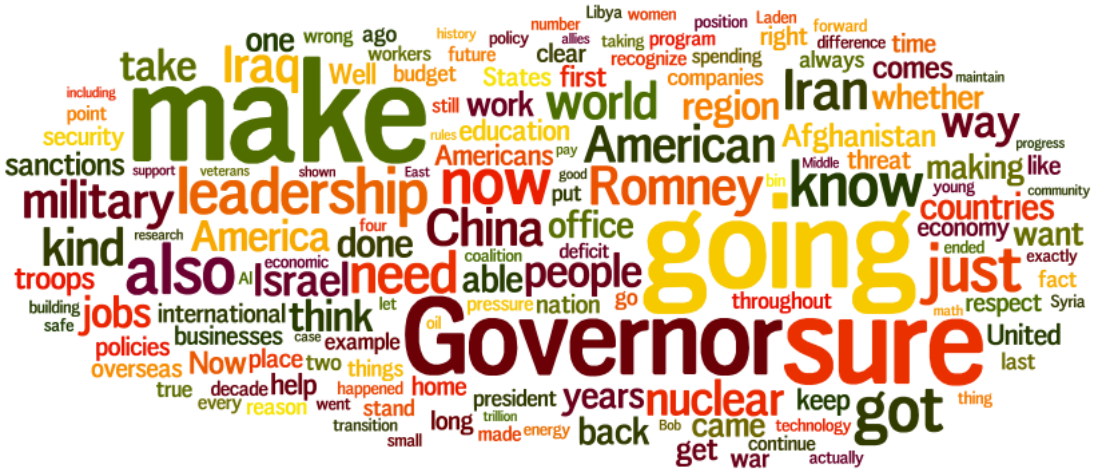
2.1 Definizioni e applicazioni

2.1.1 Cos'è una word cloud?

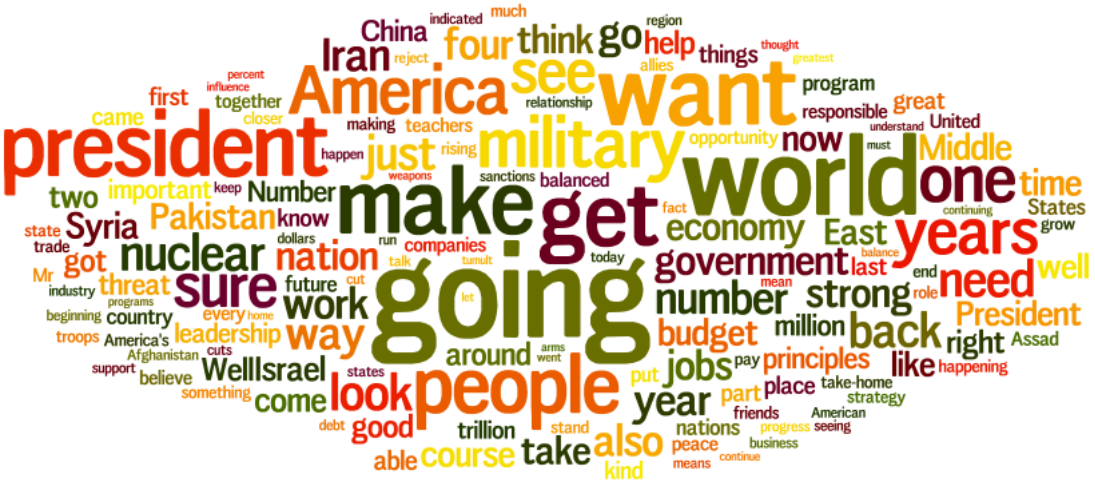
Il recente sviluppo di Internet, con l'avvento del Web 2.0, assieme al grande progresso tecnologico dei calcolatori, ha comportato un'ingente produzione di dati sul web e sulle piattaforme web based, per cui il problema di estrarre, gestire e visualizzare efficacemente tale informazione è diventata, negli ultimi anni, un'area di ricerca piuttosto importante nella visualizzazione dell'informazione.

In generale, una **word cloud** è una rappresentazione visuale di documenti testuali, che utilizza diversi colori, font e dimensioni per raffigurare le parole più rilevanti, dette **keywords**, di un generico documento. Esse sono utilizzate, quindi, per esaminare un testo, in modo da facilitarne la comprensione, o per confrontare più testi. Ad esempio, nelle elezioni presidenziali del 2008 e del 2012 (fig. 2.1), i media americani hanno confrontato le word cloud generate dai dibattiti dei candidati alla presidenza americana, mettendo in risalto le differenze tra i discorsi dei candidati; anche in Italia, in occasione del discorso di insediamento alla Camera da parte del presidente Mattarella, alcune testate giornalistiche hanno fatto uso delle word cloud per analizzare il contenuto del discorso.

In riferimento al web, si parla invece di **tag cloud**, con evidente richiamo ai tag, ovvero ai metadati che riassumono il contenuto di un sito internet. Ogni tag, rappresenta un link ad una specifica risorsa sul web, consentendo agli utenti di accedervi tramite l'utilizzo di keywords. Il loro utilizzo si è diffuso grazie al sito di *photo sharing* Flickr [1], in cui i tag classificano in diverse categorie le foto che vengono condivise tra gli utenti.



(a) Word cloud generata dal discorso di Obama.



(b) Word cloud generata dal discorso di Romney.

Figura 2.1: Due wordcloud relative ai dibattiti tra i candidati alla presidenza statunitense.

Le parole di una word cloud sono tipicamente pesate in base all'importanza che esse ricoprono nel testo: più le parole hanno un font grande, più sono rilevanti. In questo modo, le word cloud permettono immediatamente di evidenziare ciò che è rilevante in un testo. Ci sono anche altri parametri da tenere in considerazione. In [2], Halvey et al. hanno valutato l'effetto di alcuni fattori: la posizione delle parole, la loro disposizione secondo l'ordine alfabetico e, come detto, la dimensione del font, si sono rivelati essere parametri importanti. Inoltre, hanno notato che gli utenti, piuttosto che leggere tutte le parole, danno uno sguardo generale alla word cloud. In un altro lavoro, Lohmann et al. [3] hanno scoperto che parole posizionate vicino al centro catturano di più l'attenzione rispetto a parole vicine ai bordi, così come parole posizionate in alto a sinistra vengono percepite prima delle altre da parte degli utenti.

2.1.2 Applicazioni

Esistono diversi strumenti per la creazione di word cloud. Un tool web based molto popolare, Wordle [4], grazie alle qualità grafiche e alle sue funzionalità, ha permesso la diffusione delle word cloud come potente strumento per riassumere e analizzare un testo. Con Wordle, ad esempio, è possibile impostare alcuni parametri in modo da personalizzare la word cloud finale, come il numero delle parole, il colore, gli angoli di disegno ecc.. Tuttavia, Wordle non riesce a catturare le relazioni semantiche tra le parole, proprietà che può rivelarsi cruciale nell'analisi e nella comprensione di un testo. Per ovviare a ciò, è stato proposto un ulteriore strumento, basato su Wordle, chiamato ManiWordle [5], il quale offre un buon livello di interazione con l'utente, permettendo a quest'ultimo di manipolare il disegno finale e di modificare le parole visualizzate in termini di posizione, colore e orientamento, risultando quindi più flessibile di Wordle. Un altro sistema, SparkClouds [6], tramite l'uso delle *sparklines*, mette in risalto i cambiamenti tra più word cloud. Collins et al. [7], hanno presentato Parallel Tag Clouds, uno strumento in grado di visualizzare le differenze tra i testi scritti di un ricco dataset. FacetAtlas [8], invece, è un'applicazione che, tramite grafici e mappe di densità, visualizza le relazioni che intercorrono tra i documenti di una vasta collezione di testi.

In generale, dunque, negli anni, sono state sviluppate varie applicazioni che generano word cloud, ognuna con pregi e difetti, le quali permettono di comparare diversi documenti da più prospettive. Il nostro lavoro è in direzione di quello che è il trend degli ultimi tempi, cioè quello di creare word cloud semantiche, in cui la disposizione delle parole riflette la loro correlazione semantica.

2.2 Word cloud semantiche

Recentemente, la maggior parte dei tool che generano word cloud si è posta come obiettivo quello di raggruppare semanticamente le parole estratte, utilizzando tecniche di elaborazione del linguaggio naturale per correlare parole simili tra loro. Infatti, la possibilità

di disegnare, vicine nella word cloud, parole correlate semanticamente, può migliorare l'esperienza dell'utente, come notato da Deutsch et al. in [9].

Tree Cloud [10], ad esempio, è un applicazione in cui le parole vengono disposte secondo un albero, in modo tale da preservare la loro vicinanza semantica. In [11], Cui et al., tramite misure di similarità, mirano a collocare, vicine nel disegno, parole correlate semanticamente, utilizzando poi un metodo force directed per compattare la word cloud. Wu et al. [12], utilizzano una tecnica ispirata al *seam carving* per ottenere una word cloud semantica e compatta. Questo lavoro di tesi, invece, prende spunto dal recente lavoro svolto da Kobourov et al. [13], in cui vengono implementati due nuovi algoritmi di visualizzazione, da confrontare con altri algoritmi esistenti, per analizzare la qualità delle word cloud in base a diverse metriche, ovviamente partendo dalla base comune costituita dalla coerenza semantica nella disposizione delle parole.

Capitolo 3

Word cloud dinamiche

—— INTRO DA SCRIVERE ——

3.1 Definizioni e applicazioni

Negli ultimi anni, sono state proposte diverse applicazioni per la creazione di word cloud. Oltre alle word cloud semantiche e non semantiche, tali applicazioni si possono suddividere in due ulteriori categorie: word cloud **statiche** e word cloud **dinamiche** (o **tempo varianti**). La principale differenza tra queste due classi è chiaramente costituita dal fattore tempo: le word cloud dinamiche, infatti, hanno come obiettivo quello di illustrare l'evoluzione temporale di un documento o di un set di documenti. I grafici a barre, per esempio, sono tipicamente utilizzati per rappresentare l'andamento temporale di una qualche variabile e consentirne l'analisi visuale [14] [15]; Dubinko et al. [16] hanno sviluppato un tool che mostra l'evoluzione dei tag in Flickr e che permette l'interazione con gli utenti; Cui et al. [11] hanno proposto un sistema che abbina un grafico di tendenza (*trend chart*) alle word cloud di un insieme di documenti, per illustrarne l'evoluzione semantica.

Sebbene tutti questi lavori abbiano come finalità quella di visualizzare il trend temporale di un insieme di documenti, le informazioni spaziali e temporali sono rappresentate da immagini statiche. Diversamente, un lavoro interessante è stato svolto di recente da Chi et al. [17], in cui viene mostrato, in modo dinamico, il progresso temporale di un set di documenti tramite l'utilizzo di tecniche di *morphing*, che permettono di passare gradualmente da una word cloud di un documento ad un'altra, modificandone anche la forma. Tuttavia, in questo studio, non si tiene conto dell'evoluzione semantica dei documenti.

Il nostro lavoro, invece, a differenza degli altri citati precedentemente, si pone come obiettivo quello di mostrare lo sviluppo temporale di un solo testo, utilizzando tecniche di morphing tra le word cloud generate durante l'elaborazione del testo, mantenendo però il vincolo di vicinanza geometrica tra parole correlate semanticamente.

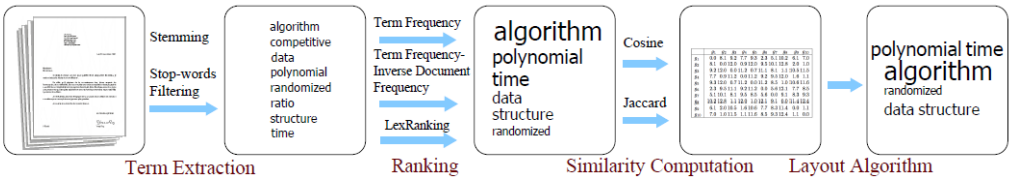


Figura 3.1: Generazione di una word cloud semantica.

3.2 Algoritmi di generazione di una word cloud semantica

Tutti gli algoritmi di visualizzazione di una word cloud prendono come input un grafo pesato, i cui vertici sono le parole, rappresentate da rettangoli. Tuttavia, sono necessari alcuni passi di preprocessing, che consentono di estrarre questa informazione dall'input. L'algoritmo di visualizzazione disegna, per quanto possibile, le parole più simili vicine tra loro. Il processo di creazione di una word cloud è indicato in figura 3.1. Infine, viene applicato un algoritmo di clustering per assegnare lo stesso colore a parole dello stesso cluster.

3.2.1 Estrazione keywords

Il processo di estrazione delle keywords prevede una serie di passaggi preliminari, con tecniche di elaborazione del linguaggio naturale, le quali predispongono, in maniera appropriata, il testo in ingresso all'algoritmo di estrazione delle parole.

Innanzitutto, il testo viene suddiviso in frasi e ogni parola viene suddivisa in *token*: ciò può essere eseguito mediante l'ausilio di librerie di elaborazione del linguaggio naturale (e.g. *Apache OpenNLP* [18]). Successivamente, dal testo vengono eliminate le *stop words*, cioè articoli, congiunzioni e parole di uso comune che sono poco rilevanti dal punto di vista informativo. Le parole rimanenti vengono quindi raggruppate in base alla radice (in inglese, *stem*), tramite un algoritmo di *stemming*: in questo modo, parole del tipo *player*, *play* e *playing* vengono raggruppate secondo la radice comune *play*. Nella nostra implementazione, è stato utilizzato il noto algoritmo di Porter [19]. Alla fine, nella word cloud finale, viene visualizzata la variante più frequente della parola.

Una volta eseguiti questi passaggi, si procede all'estrazione delle parole e al loro ranking in modo da trovare quelle più rilevanti, utilizzando tecniche di Information Retrieval. Ogni algoritmo assegna alle parole un punteggio e ne seleziona le n più frequenti, dove n è il numero di parole da visualizzare nella word cloud.

Term Frequency

Il modo più intuitivo di assegnare un peso alle parole è quello di contare le singole occorrenze. Questo è ciò che viene fatto dall'algoritmo Term Frequency (TF). Tuttavia, la rilevanza di un termine non aumenta linearmente con il numero delle occorrenze, per cui spesso il punteggio ottenuto viene scalato tramite una qualche funzione. In tabella

3.1 sono riportate le tipiche funzioni che vengono applicate per pesare tale contributo, dove l'argomento $tf_{t,d}$ è la frequenza del termine t nel documento d.

Tipo funzione	Peso
Binaria	0,1
Lineare	$tf_{t,d}$
Radice quadrata	$\sqrt{tf_{t,d}}$
Logaritmo	$1 + \log tf_{t,d}$
Doppia normalizzazione con parametro K	$K + (1-K) * \frac{tf_{t,d}}{\max_{t' \in d} tf_{t',d}}$

Tabella 3.1: Term Frequency ranking: funzioni

Tuttavia, pur dopo aver rimosso le stop words, l'algoritmo Term Frequency tende ad assegnare punteggi troppo alti a termini poco rilevanti. Termini rari, invece, hanno contenuto informativo più alto rispetto a termini frequenti, per cui ad essi vanno assegnati punteggi più elevati. In particolare, si definisce il parametro IDF (*Inverse Document Frequency*) di un termine t la quantità:

$$idf_t = \log \frac{N}{df_t}, \tag{3.1}$$

dove N è la dimensione di una collezione di documenti e df_t è la *document frequency*, cioè il numero totale di documenti in cui il termine t compare. Per termini frequenti in una collezione, tale valore tende a zero, mentre per termini rari il punteggio sarà più alto. Lo scaling che viene applicato è solitamente logaritmico, con qualche variante.

TF-IDF

Ora si possono combinare le due definizioni di TF e IDF per produrre un ulteriore algoritmo, noto come TF-IDF, il quale assegna, ad ogni termine t di un documento d, la quantità

$$tfidf_t = tf_t \times idf_t. \tag{3.2}$$

Ne segue che:

1. se t è un termine comune nella collezione, avrà un tf_t alto, ma un idf_t basso;
2. se t è un termine raro nella collezione, ma frequente nel documento d, allora avrà entrambi i contributi elevati.

In pratica, con questo approccio, vengono filtrati i termini molto comuni, mentre quelli davvero rilevanti per il documento vengono estratti.

Uno degli schemi più noti in letteratura per calcolare la $tfidf_t$, come suggerito in [20], è il seguente:

$$tfidf_t = (1 + \log tf_t) \times \log \frac{N}{df_t}. \tag{3.3}$$

LexRank

Il terzo algoritmo di ranking è **LexRank** [21], già usato in [12] per la creazione di word cloud semantiche. Tale algoritmo prende spunto dall'algoritmo PageRank [22], utilizzato da Google per assegnare un punteggio alle pagine web e quindi migliorare le ricerche che si effettuano con il noto motore di ricerca. LexRank è un algoritmo basato su un grafo $G = (V, E)$, dove i vertici sono le parole, collegati da archi che rappresentano le co-occorrenze di due parole all'interno di una frase. Ogni arco (i, j) ha un peso w_{ij}, che rappresenta il numero di occorrenze della parola i e della parola j all'interno di una stessa frase. I valori vengono poi calcolati sfruttando il concetto di centralità dei vertici di G. Sia R il vettore di ranking, di dimensione $1 \times n$, dove n è il numero di keywords. R è definito come:

$$1 \tag{3.4}$$

3.2.2 Calcolo similarità

Il passo successivo è quello di calcolare la similarità tra le keywords, ovvero quanto esse sono correlate tra loro. Data la lista delle n parole estratte, viene calcolata la matrice $n \times n$ delle similarità tra coppie di parole. Ogni valore è compreso tra 0 (nessuna correlazione) e 1 (massima correlazione). Esistono diversi algoritmi per il calcolo delle similarità. Tutti usano uno spazio vettoriale di dimensione n (pari al numero di parole estratte), dove il generico vettore $w_i = \{w_{i1}, w_{i2}, ..., w_{in}\}$ rappresenta la co-occorrenza della parola i con le altre $n - 1$ parole.
Di seguito sono esposte le tecniche di calcolo da noi implementate.

Cosine Similarity

La cosine similarity tra due vettori w_i e w_j viene calcolata come:

$$sim_{ij} = \frac{w_i \cdot w_j}{||w_i||\,||w_j||}. \tag{3.5}$$

In pratica, tale quantità corrisponde alla misura dell'angolo formato tra i vettori w_i e w_j. Se la similarità è 1, allora l'angolo formato è pari 0°, mentre se la similarità è 0, allora i due vettori sono perpendicolari e non condividono alcuna frase.

Jaccard Similarity

La Jaccard similarity è definita come il rapporto tra il numero delle frasi condivise tra due parole e il numero totale delle frasi in cui esse compaiono. In formule:

$$sim_{ij} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}, \tag{3.6}$$

dove S_i è l'insieme delle frasi in cui compare l'i-esima parola.

Jaccard Similarity estesa

Un ulteriore modo per il calcolo della similarità è rappresentato dalla Jaccard similarity estesa (anche nota come **coefficiente di Tanimoto**), definita come:

$$sim_{ij} = \frac{w_i \cdot w_j}{||w_i||^2 + ||w_j||^2 - w_i \cdot w_j} \tag{3.7}$$

Tale misura si riduce alla Jaccard Similarity nel caso di vettori binari.

Bibliografia

[1] Flickr: photo sharing site, `http://www.flickr.com/photos/tags`.

[2] Martin J. Halvey and Mark T. Keane. An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1313–1314, New York, NY, USA, 2007. ACM.

[3] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I*, INTERACT '09, pages 392–404, Berlin, Heidelberg, 2009. Springer-Verlag.

[4] Fernanda B. Viegas, Martin Wattenberg, and Jonathan Feinberg. Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, November 2009.

[5] Kyle Koh, Bongshin Lee, Bo Hyoung Kim, and Jinwook Seo. Maniwordle: Providing flexible control over wordle. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1190–1197, 2010.

[6] Bongshin Lee, Nathalie Henry Riche, Amy K. Karlson, and Sheelash Carpendale. Sparkclouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, November 2010.

[7] Christopher Collins, Fernanda B. Viegas, and Martin Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE VAST*, pages 91–98. IEEE Computer Society, 2009.

[8] Nan Cao, Jimeng Sun, Yu-Ru Lin, David Gotz, Shixia Liu, and Huamin Qu. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1172–1181, 2010.

[9] Stephanie Deutsch, Johann Schrammel, and Manfred Tscheligi. Comparing different layouts of tag clouds: Findings on visual perception.

[10] Philippe Gambette and Jean Véronis. Visualising a Text with a Tree Cloud. In *IFCS'09: International Federation of Classification Societies Conference*, Studies in

Classification, Data Analysis, and Knowledge Organization, pages 561–569, Dresde, Germany, March 2009. Springer Berlin / Heidelberg.

[11] Weiwei Cui, Yingcai Wu, Shixia Liu, Furu Wei, Michelle X. Zhou, and Huamin Qu. Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, 30(6):42–53, 2010.

[12] Yingcai Wu, Thomas Provan, Furu Wei, Shixia Liu, and Kwan-Liu Ma. Semantic-Preserving Word Clouds by Seam Carving. *Computer Graphics Forum*, 2011.

[13] Lukas Barth, StephenG. Kobourov, and Sergey Pupyrev. Experimental comparison of semantic word clouds. In Joachim Gudmundsson and Jyrki Katajainen, editors, *Experimental Algorithms*, volume 8504 of *Lecture Notes in Computer Science*, pages 247–258. Springer International Publishing, 2014.

[14] Lee Byron and Martin Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, November 2008.

[15] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *Proc. IEEE Symposium on Information Visualization*, pages 115–123, 2000.

[16] Micah Dubinko, Ravi Kumar, Joseph Magnani, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Visualizing tags over time. *ACM Trans. Web*, 1(2), August 2007.

[17] Ming-Te Chi, Shih-Syun Lin, Shiang-Yi Chen, Chao-Hung Lin, and Tong-Yee Lee. Morphable word clouds for time-varying text data visualization. *IEEE Trans. Vis. Comput. Graph.*, 21(12):1415–1426, 2015.

[18] . Apache OpenNLP, `http://opennlp.apache.org`, .

[19] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[20] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[21] Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:2004, 2004.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.