Agentes inteligentes

Agentes para Recuperación de Información

Rubén Fuentes Fernández
Dep. de Ingeniería del Software e Inteligencia Artificial

UCM UNIVERSIDAD COMPLUTENSE MADRID

http://grasia.fdi.ucm.es

Sumario

- Recuperación de información en Internet
- Situación actual
  - Técnicas estadísticas clásicas
  - Hiperenlaces
- Potenciales soluciones
- Agentes
  - Búsqueda dinámica
  - Búsqueda focalizada
  - Masiva evolutiva
- Conclusiones

Introducción

Estado actual

Recuperación de información en Internet

- La Recuperación de Información (RI) abarca las técnicas
  - para ayudar a los usuarios
  - en sus intentos de localizar y recuperar información
  - en bases documentales
  - con el soporte de herramientas automatizadas.
- Internet es una base documental particular.
  - Información
    - Distribuida
    - Sin estrucura
    - Volatil
    - Dinámica (parte)
    - Heterogénea
    - Redundante
  - Usuarios con intereses y contextos diferentes.

Situación actual

- Herramientas de RI
  - Motores
    - Altavista → www.altavista.com
  - Directorios
    - Yahoo! → www.yahoo.com
  - Información topológica
    - Google → www.google.com
  - Metabuscadores
    - MetaCrawler → www.metacrawler.com
- Problemas
  - Bajo *recall*
  - Baja precisión.
  - Información desactualizada.

Potenciales soluciones

- Distribución
- Especialización en usuarios y temas.
- Uso de múltiples fuentes y formatos de documentos.
- Información topológica.
- Colaboración con sistemas centralizados.
- Exploración de la web oculta.
- ...
- ¡¡¡Integración de soluciones!!!

Recuperación de información

Agentes

Búsqueda dinámica

Búsqueda dinámica

- Los servicios de búsqueda tradicionales usan índices preconstruidos.
  - La búsqueda real se realiza en el servidor sobre los datos del índice.
- La búsqueda dinámica trae los datos en el momento en que se está realizando la búsqueda.

Búsqueda dinámica

Letizia

Letizia

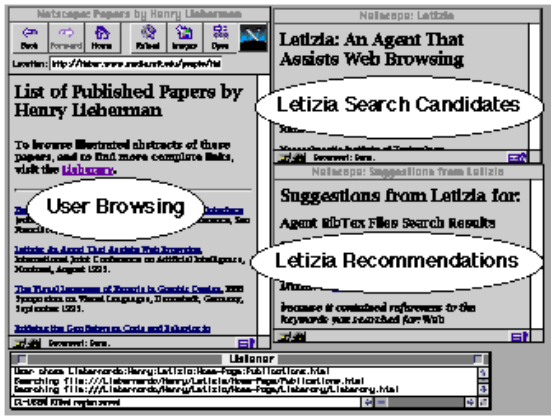
- Intuición:
  - Búsquedas incrementales en vecindades semánticas.
  - frente a búsqueda global realizada con un motor.
- Cooperación usuario-agente.
  - El agente busca rápido.
  - El hombre decide cuando una página es interesante.
- *Letizia* sugiere al usuario páginas a explorar.

El algoritmo de *Letizia*

- Búsqueda incremental primero en anchura.
  - Desde la página actual.
- Usa la información de los enlaces.
  - Primero en profundidad en vecindades relevantes.
- Leer una página es evidencia de interés en ella.
  - Análisis TFIDF modificado a cada página.
  - El perfil de usuario como lista de palabras con pesos.

Interfaz de *Letizia*

- Configuración multiventana en *browser* convencional.



Búsqueda dinámica

Fish y *shark* search

Fish-search

- Intuición.
  - Los documentos relevantes tienen vecinos relevantes.
- Agentes de búsqueda como peces.
  - La comida es información relevante.
  - El agua está contaminada cuando el ancho de banda es pobre.
  - Con comida los peces se reproducen y buscan.
  - Los peces pueden morir.
    - En ausencia de comida.
    - Por agua contaminada.

Algoritmo del *fish-search*

- Entrada:
  - Conjunto semilla de URL's.
  - *Query* a realizar.
- Lista de prioridad con las URL's a explorar.
- En cada iteración:
  - Obtener el primer documento de la lista.
  - Evaluar si el documento es relevante para la *query*.
  - Decidir si se sigue o no la exploración en esa dirección.
  - Insertar los hijos del documento en la lista de prioridad según la relevancia del padre.

Limitaciones del *fish-search*

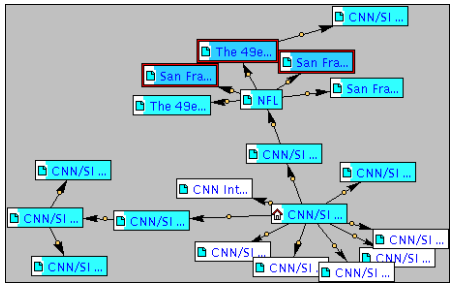
- La diferenciación de la prioridad de las páginas en la lista es muy baja.
- La reducción del número de hijos considerados debido al parámetro *ancho* es arbitraria.
- La relevancia de un documento es dada sólo por su contenido.
- Bajo restricciones de tiempo no descubre suficientes direcciones relevantes.

Shark-search

- Evolución del *fish-search*.
- La calificación de la relevancia es un valor real.
- Cálculo de similitud según
  - Modelo de espacio vectorial.
  - Metainformación en los enlaces.
- Estas heurísticas eliminan el uso del parámetro *ancho*.

Mapuccino

- *Shark-search* está embebido en *Mapuccino*.



- Código de colores.
  - Azul para las páginas de interés
  - Blanco para las páginas no relevantes.

Búsqueda focalizada

Búsqueda focalizada

- Los usos del lenguaje dependen del contexto.
- Sistemas centrados en temas o usuarios concretos.
- Centrar la búsqueda permite
  - Adoptar convenciones precisas respecto al lenguaje.
  - Mayor poder discriminante de los términos.

Búsqueda focalizada

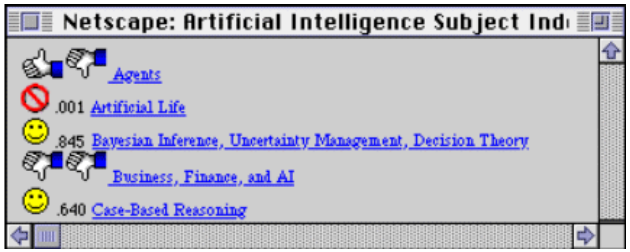
Syskill & Webert

Syskill & Webert

- Recoge evaluaciones del usuario sobre el interés que tienen para él páginas de la Web.
- En función del perfil se determina que otras páginas pueden interesar al usuario.
- Aprende un perfil separado para cada tema.

Recolección de información

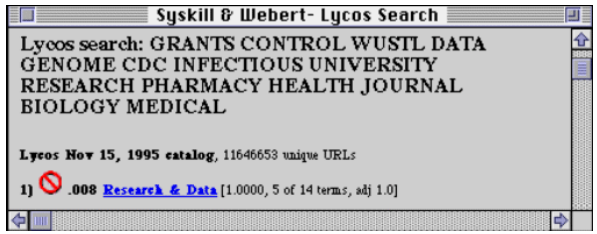
- Funcionalidad adicional al código HTML.
  - Elementos para puntuación.



- El sistema hace un sumario de las páginas puntuadas.

Perfil

- El perfil de usuario contiene:
  - Palabras comunes en las páginas interesantes.
  - Palabras discriminantes para las páginas relevantes.
- A partir del perfil, *Syskill & Webert* permite generar consultas para LYCOS.



Aprendizaje del perfil

- El perfil para un tema depende de todas las calificaciones previas de páginas en ese tema.
- Páginas Web como vectores de booleanos.
- Los términos discriminantes proporcionan la mayor ganancia de información en la clasificación.
- Clasificador Bayesiano simple.
  - Aprendizaje lineal.
  - Predicción constante.

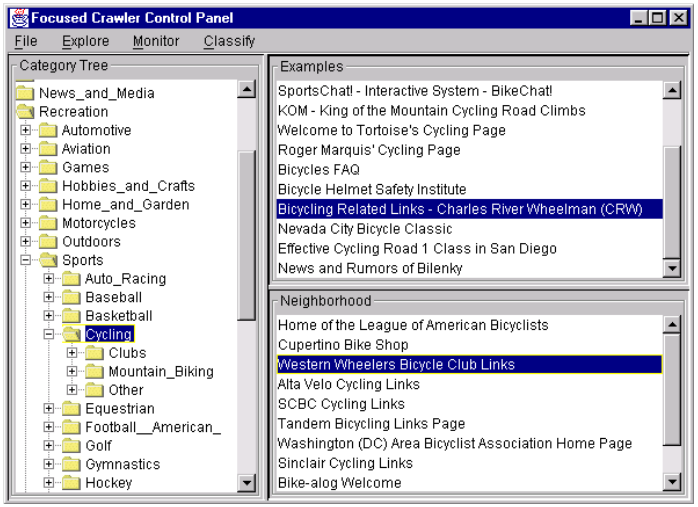
Búsqueda focalizada

Crawler focalizado

Crawler focalizado

- *Crawlers* de propósito general conducen a
  - Una indexación pobre de partes de la Web.
  - Una generalidad excesiva de los índices.
- El *crawler* focalizado mantiene páginas sobre un conjunto específico de temas.
- Objetivo final de esta especialización
  - Imponer estructura temática en la Web.

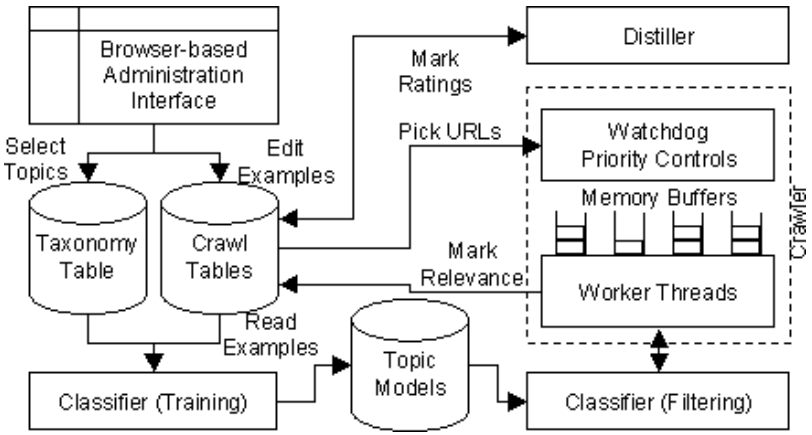
Taxonomía



Uso del *crawler* focalizado

- Creación de una taxonomía canónica.
- Recolección de ejemplos.
- Selección y refinamiento de la taxonomía.
- Exploración interactiva
  - El sistema propone URL's como ejemplos.
- Entrenamiento
  - Integración de los refinamientos del usuario.
- Descubrimiento de recursos.
- Destilación de *hubs*.
- Realimentación
  - El usuario indica las páginas útiles.

Arquitectura (I)

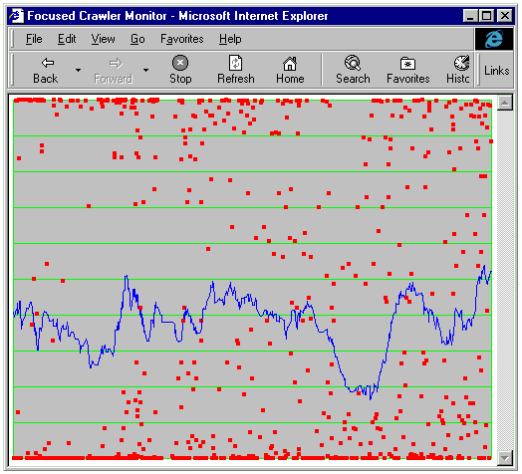


Arquitectura (II)

- 3 componentes principales
  - Clasificador
    - Juzga la relevancia de las páginas.
  - Destilador
    - Determina la centralidad de las páginas recorridas.
  - *Crawler*
    - Gobernado por el clasificador y el destilador.
- El *crawler* está compuesto de varias hebras.
- El clasificador es invocado por cada hebra cuando encuentra una nueva página.

Ventajas del *crawler* focalizado

- Mantiene el tema centrado.
- El destilador mejora la clasificación de las páginas.



Masiva evolutiva

La metáfora ecológica

- La Web es un ecosistema de información.
  - Pastos → páginas web
  - Herbívoros → servicios de búsqueda tradicionales
  - Carnívoros → usuarios
- La dieta del depredador es óptima si
  - Clasifica rápidamente las presas.
    - Tipo.
    - Frecuencia en el lugar.
    - Beneficios que proporcionan.
  - Decide que categorías perseguir y a cuales ignorar.

Manadas de agentes carnívoros

- El usuario es servido por una manada.
  - Agentes especializados.
- Los agentes adaptativos pueden manejar el contexto.
- Los agentes personales se adaptan al usuario
  - Incluso si sus intereses cambian en el tiempo.
- Uso racional de los recursos.

Masiva evolutiva

InfoSpiders

InfoSpiders

- *InfoSpiders* es un SMA.
- Agentes con una representación adaptativa.
  - Topología estadística → palabras
  - Topología estructural → enlaces
- Cada agente
  - Navega de documento en documento en línea.
  - Decide autónomamente los enlaces a seguir.
  - Se adapta al contexto local.
  - Se adapta a las preferencias del usuario.

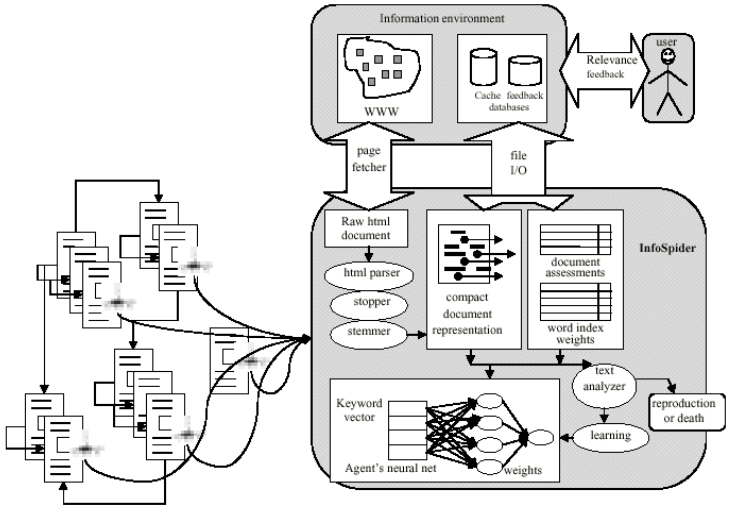
Búsqueda en *InfoSpiders*

- El usuario proporciona
  - *Query*
  - Páginas de partida.
- Un agente por cada página de partida.
  - Comportamiento aleatorio y reserva inicial de energía.
- Un agente analiza la página donde se encuentra.
  - Sigue los enlaces en función de su relevancia estimada.
- La energía del agente se actualiza durante su operación.

Uso de la energía

- Los agentes son dotados de una energía inicial.
- Energía para sobrevivir y moverse.
  - Aumenta al visitar documentos relevantes.
  - Se pierde al transferir documentos.
- Aprendizaje Q
  - Cambios instantáneos en la energía como refuerzo.
- Reproducción con cierto nivel de energía.
- Cuando un agente agota su energía muere.

Arquitectura (I)



Arquitectura (II)

- El agente interactúa con el entorno.
  - Web.
  - Datos almacenados en discos locales.
- El usuario interactúa con el entorno.
  - El estado actual de la búsqueda
  - Un documento sugerido por los agentes.
  - Proporcionando información sobre relevancia.
- Cada documento es representado por una lista de raíces de palabras y de enlaces.

Representación adaptativa

- Representación adaptativa.
  - Genotipo que determina el comportamiento.
  - Estrategias de búsqueda sobre el genotipo.
- El genotipo contiene
  - Grado de confianza en las descripciones de los enlaces.
  - Lista de palabras con pesos
    - Discriminan los documentos relevantes.
  - Una red neuronal.
- Adaptación por evolución del genotipo.

Cálculo de relevancia

- La relevancia se calcula con la red neuronal.
  - Entrada con la frecuencia de una palabra clave en las cercanías del enlace a atravesar.
- Aprendizaje de la red.
  - Comparación de relevancias.
    - Documento.
    - Enlace que condujo a él.
  - Actualización de los pesos por *back-propagation*.

Conclusiones

Conclusiones

- Buscadores centralizados con técnicas estadísticas.
  - Largas listas de respuestas.
  - Bajo *recall*.
  - Baja precisión.
- Solución
  - Combinación de técnicas.
    - Búsqueda dinámica.
    - Búsqueda focalizada.
    - Masiva evolutiva.
  - Herramientas adaptadas al usuario concreto.
- Compromiso.
  - Mayor tiempo de respuesta.
  - Mejores resultados.

Referencias

- Brin, S., Page, L.: *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Proceedings 7th International World Wide Web Conference (WWW7), Brisbane, Australia, 14-18 Abril 1998, pp. 107-117.
- Chakrabarti, S., van den Berg, M., Dom, B.: *Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery*. Proceedings 8th International World Wide Web Conference (WWW8), Toronto, Canada, 11-14 Mayo 1999, pp. 1623-1640.
- Etzioni, O.: *Moving Up the Information Chain*. AI Magazine, Summer, pp. 11-18. 1997.
- Hersovici, M., Jacovi, M., Maarek, Y. S., Pelleg, D., Shtalhaim, M., Ur, S.: *The shark-search algorithm – An application: tailored Web site mapping*. In Proceedings of the 7th International World Wide Web Conference, 1998.
- Lieberman, H.: *Autonomous Interface Agents*. Proceedings ACM Conference on Computers and Human Interface (CHI-97), Atlanta, Georgia, USA, 22-27 March 1997, pp. 67-74.
- Menczer, F., Monge, A. E.: *Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study*. In M. Klusch (Ed.), Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet. Berlin, Germany: Springer. 1999, pp. 323-347.
- Moukas, A.: *Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem*. Proceedings 1st Conference on Practical Application of Intelligent Agents & Multi-Agent Technology (PAAM), Londres, UK, pp. 22-24 April 1996, pp. 421-436.
- Pazzani, M., Maramatsu, J., Billsus, D.: *Syskill & Webert: Identifying interesting web sites*. Proceedings 13th National Conference on Artificial Intelligence (AAAI-96), Portland, Oregon, USA, 4-8 Agosto 1996, vol. 1, pp. 54-61
- Salton, G.: *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Reading, MA, USA: Addison-Wesley. 1998.
- Selberg, E., Etzioni, O.: *The MetaCrawler Architecture for Resource Aggregation on the Web*. IEEE Expert, vol. 12(1), pp. 8-14. 1997.
- Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. NY, USA: Addison-Wesley, ACM Press. 1999.

¿Preguntas?