Information retrieval per la navigazione assistita

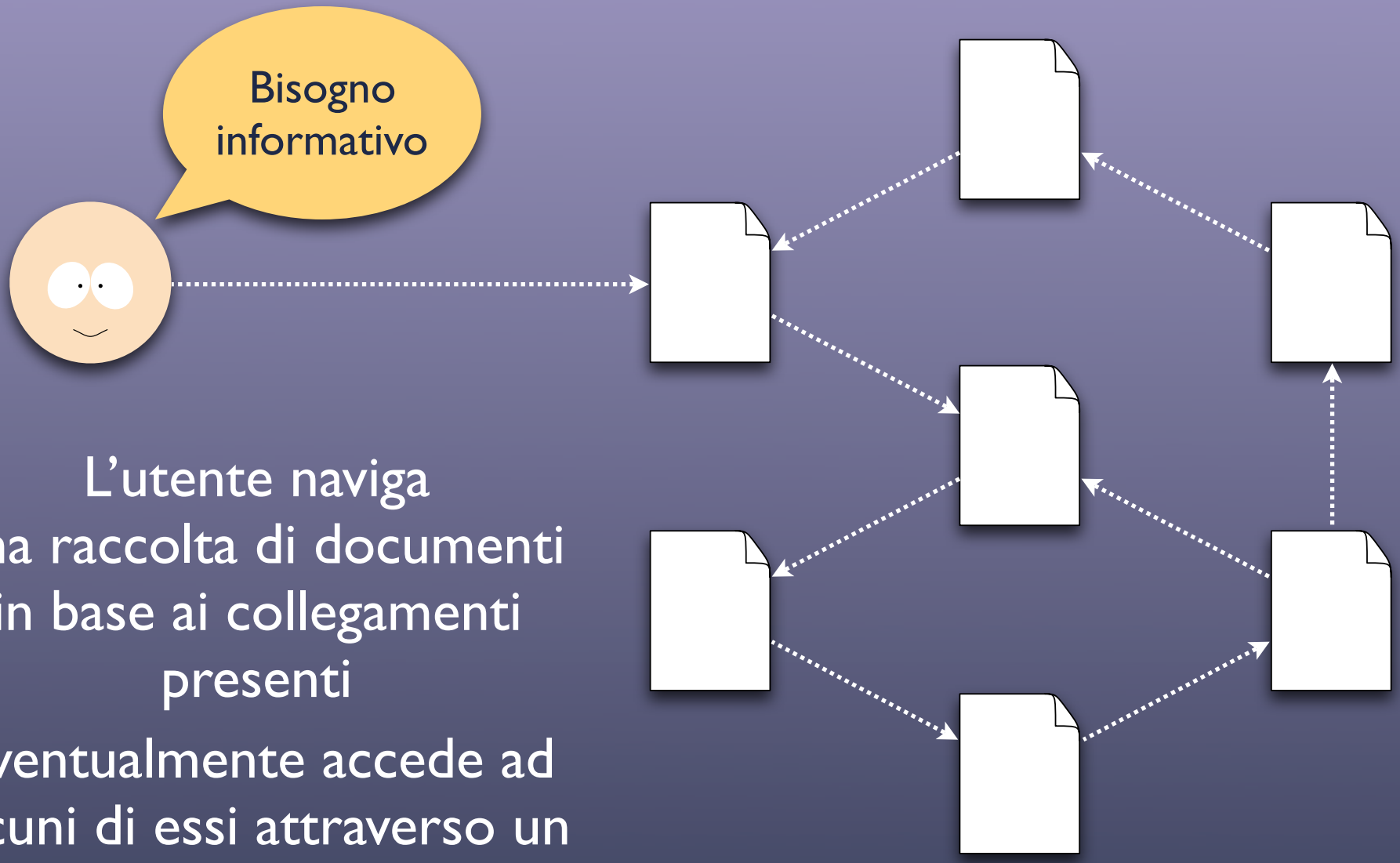
PyCon Due

Firenze, 9-11 Maggio 2008

Marco Spisto
Università degli Studi di Napoli
Federico II

Sommario

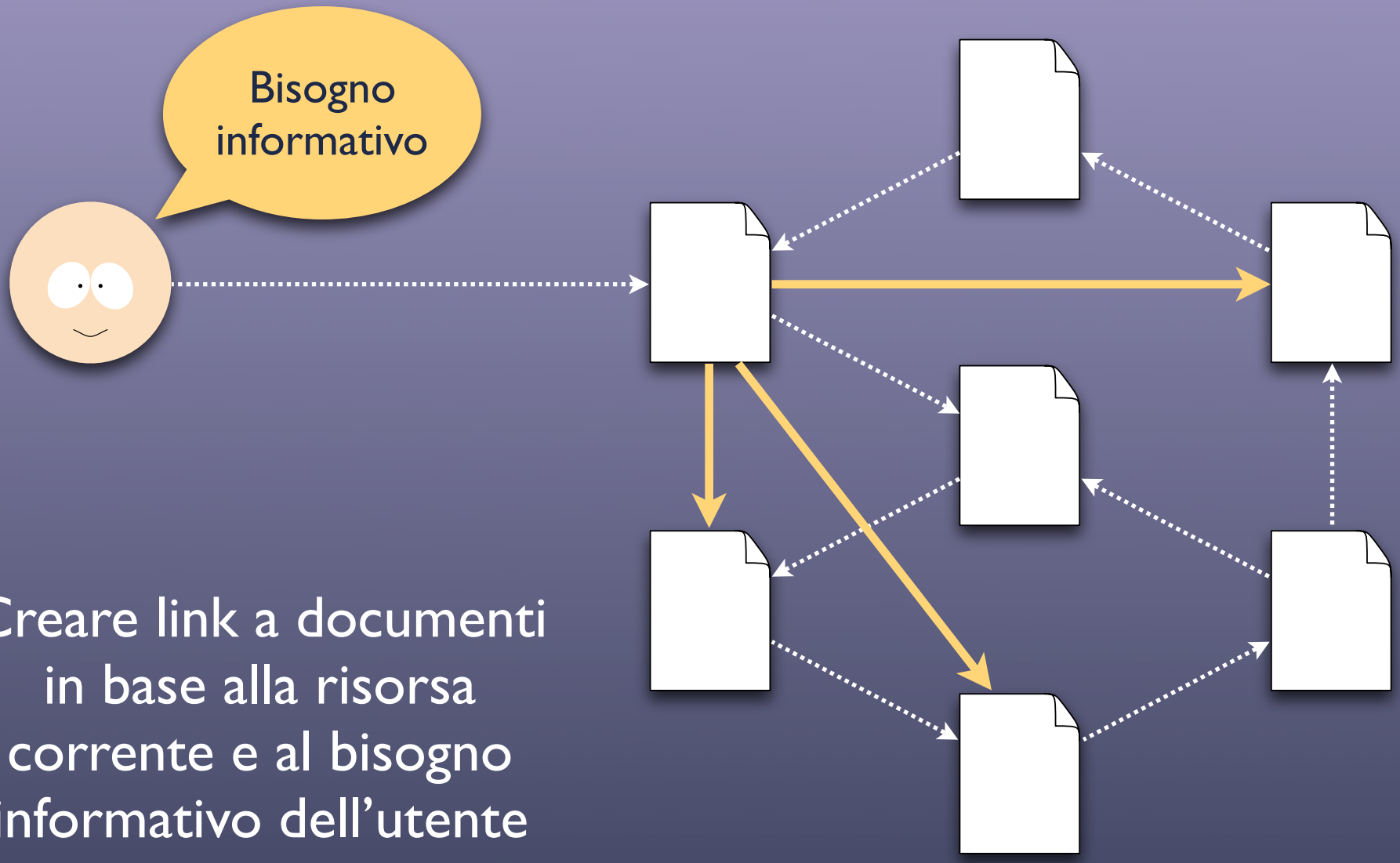
- Introduzione
- Information retrieval
- Rappresentazione vettoriale
- Considerazioni finali



L'utente naviga
una raccolta di documenti
in base ai collegamenti
presenti

Eventualmente accede ad
alcuni di essi attraverso un
motore di ricerca

Navigazione tradizionale



Creare link a documenti
in base alla risorsa
corrente e al bisogno
informativo dell'utente

Navigazione assistita

Obiettivi

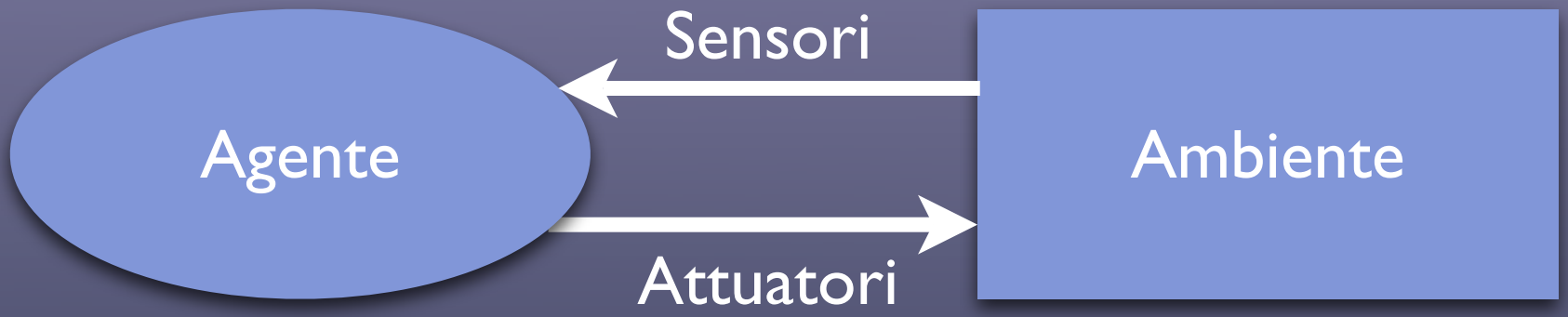
- Creare dinamicamente collegamenti ad altri documenti scegliendoli in base al documento e al bisogno informativo dell'utente
- Integrazione di più raccolte di documenti

Caso di studio

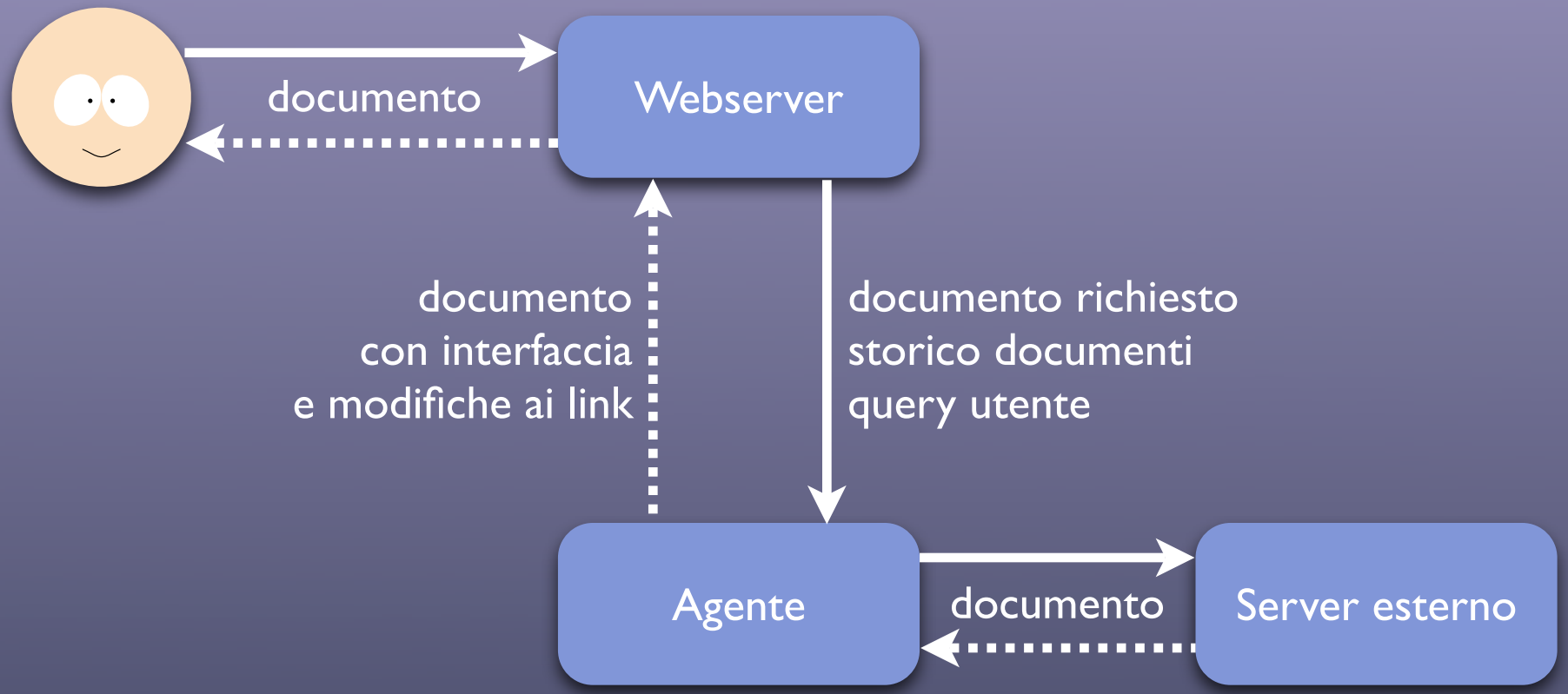
- Documentazione Python
  - Documentazione librerie Python
  - Tutorial Python
  - Libro Dive into Python
- Documenti disponibili liberamente
- Dati in formato HTML

Sistema adattivo

Sistema che crea un modello utente
e lo utilizza per fornire le informazioni
che potrebbero essere più interessanti



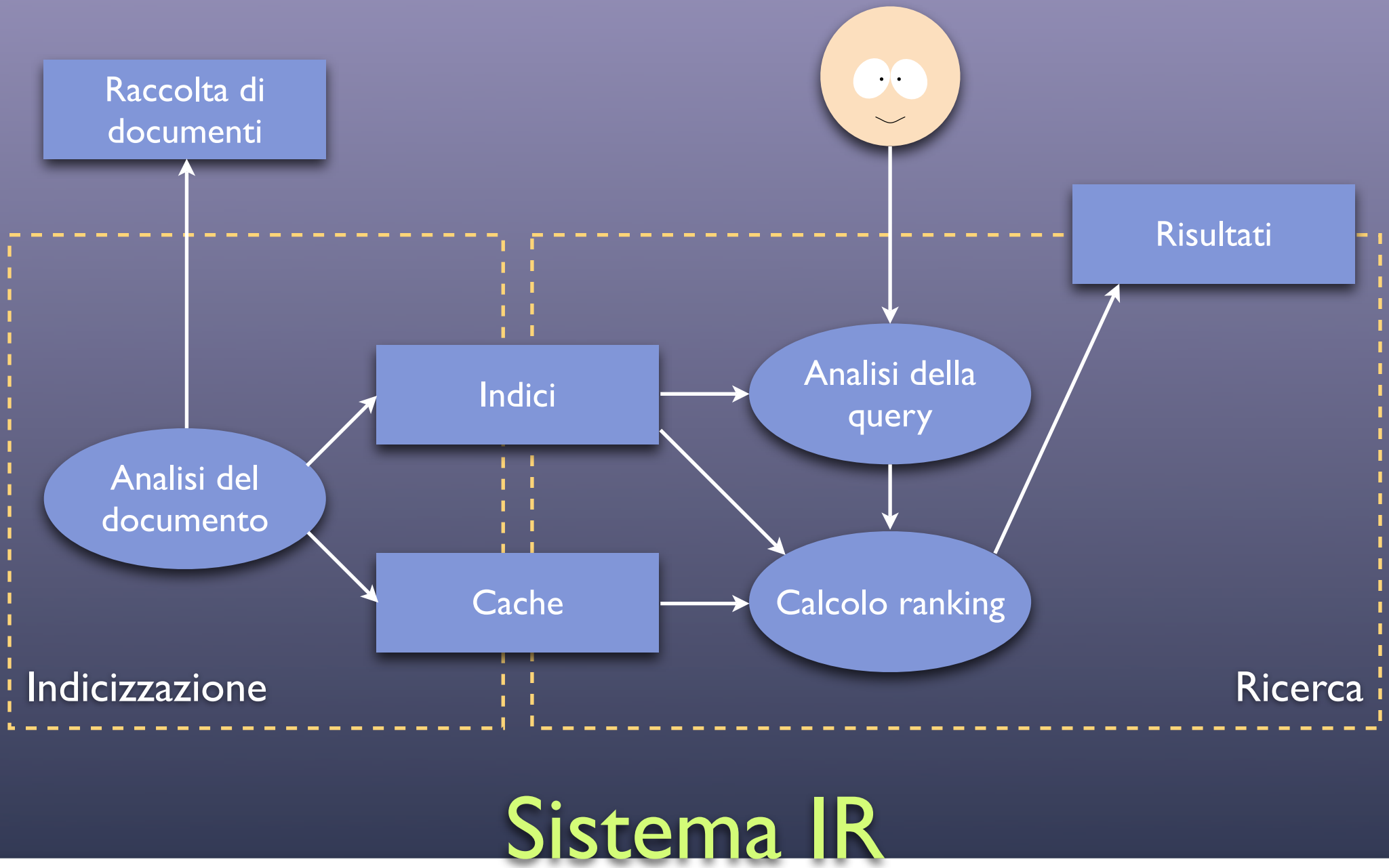
Agente



Architettura

Information retrieval

Recupero di informazioni
solitamente non strutturate
da una raccolta dati
a fronte di un bisogno informativo


Raccolta di documenti
Risultati
Indici
Analisi della query
Analisi del documento
Cache
Calcolo ranking
Indicizzazione
Ricerca

Sistema IR

Raccolta di documenti

- Un sistema di information retrieval dipende fortemente dalla raccolta di documenti
- Caratteristiche
  - Dimensione della raccolta
  - Tipologia dei dati
  - Linguaggio
  - Privatezza e accessibilità

Segmentazione

- Segmento
  Unità di informazione restituita come risultato della ricerca
- Un segmento può essere:
  - un documento intero
  - una sezione del documento
  - un paragrafo
  - una frase

Estrazione dei termini

- Termine
  Sequenza di lettere, numeri, punteggiatura...
  che è possibile estrarre da un documento
- L'insieme dei termini è il vocabolario
  del sistema
- Le occorrenze di un termine all'interno
  di un documento sono dette token

Bag of words

- Documento visto come un contenitore di termini
- Perdita delle informazioni posizionali dei termini nel documento
- Rappresentazione alla base di molti modelli per l'information retrieval

Doc1	Doc2	Doc3	Doc4	Doc5
Stan Kyle Cartman Kenny	Butters Stan Stan	Kyle Stan Kenny Kyle	Kenny Stan Kenny	Cartman Stan

	DF	Doc1	Doc2	Doc3	Doc4	Doc5
Butters	1	0	1	0	0	0
Cartman	2	1	0	0	0	1
Kenny	3	1	0	1	2	0
Kyle	2	1	0	2	0	0
Stan	5	1	2	1	1	1

Document frequency del termine Stan

Term frequency del termine Stan nel documento Doc4

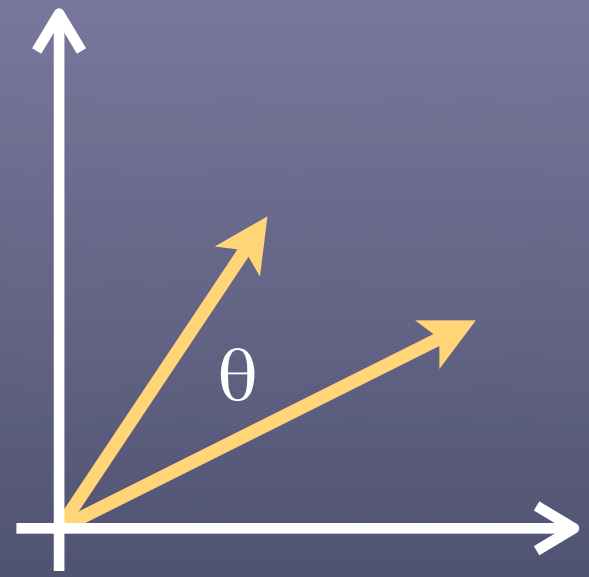
Creazione dell'indice

Rappresentazione vettoriale

Rappresentazione di documenti e query tramite vettori

Significato geometrico

- Similarità di vettori data dal coseno dell'angolo che formano
- Vettori più vicini hanno un coseno più alto



0 ⟷ 1

vettori lontani

vettori vicini

Vettore documento

- Un documento è un vettore di dimensione pari a quella del vocabolario
- Ogni elemento del vettore è un valore che rappresenta il termine nel documento

Elementi del vettore

- Si vorrebbero valori che siano:
  - Alti per termini che occorrono molte volte in un documento
    Term frequency: tf
  - Bassi per termini che occorrono in molti documenti
    Inverse document frequency: idf = log(N/df)
- Un buon compromesso: tf * idf

Vettore query

- Anche la query viene vista come un vettore di dimensione pari a quella del vocabolario
- Nel vettore gli elementi hanno valore
  - 1: se il termine è presente nella query
  - 0: se il termine è assente nella query

Calcolo della similarità

Coseno usato per calcolare un punteggio di similarità per ogni documento rispetto alla query

Computazionalmente pesante $O(N*M + T_{sort}(N))$

```
cosine(d,q):
  dot ← sum(d_t*q_t for t in
                   dictionary)
  score ← dot / (size(d)*size(q))
  return score
```

```
size(x):
  result ← sum((x_e)^2 for e in x)
  return sqrt(result)
```

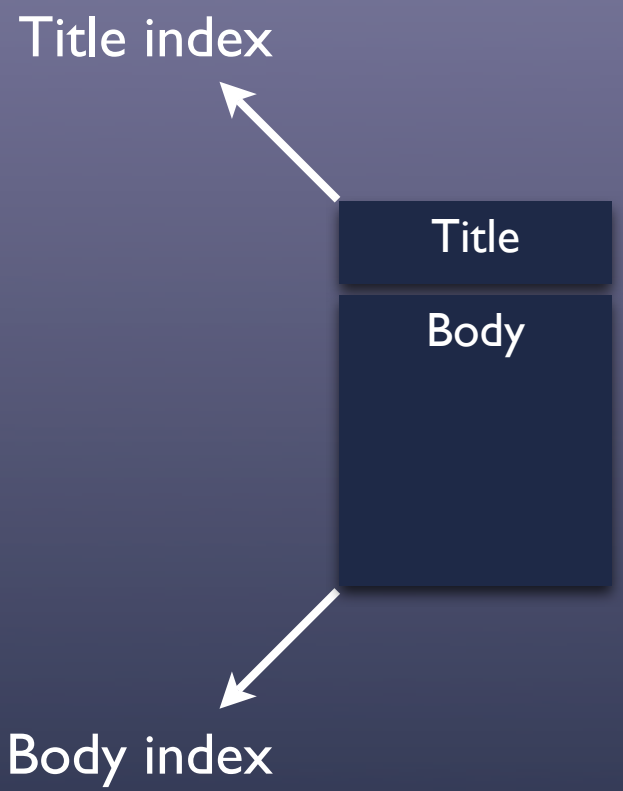
Calcolo rapido del coseno

Non è necessario iterare sugli elementi dei vettori pari a 0

Scarto a-priori dei risultati molto probabilmente non rilevanti utilizzando le postings

```
fastCosine(d,q):
  dot ← sum(d_t*q_t for t in d∩q)
  score ← dot / (d.size * q.size)
```

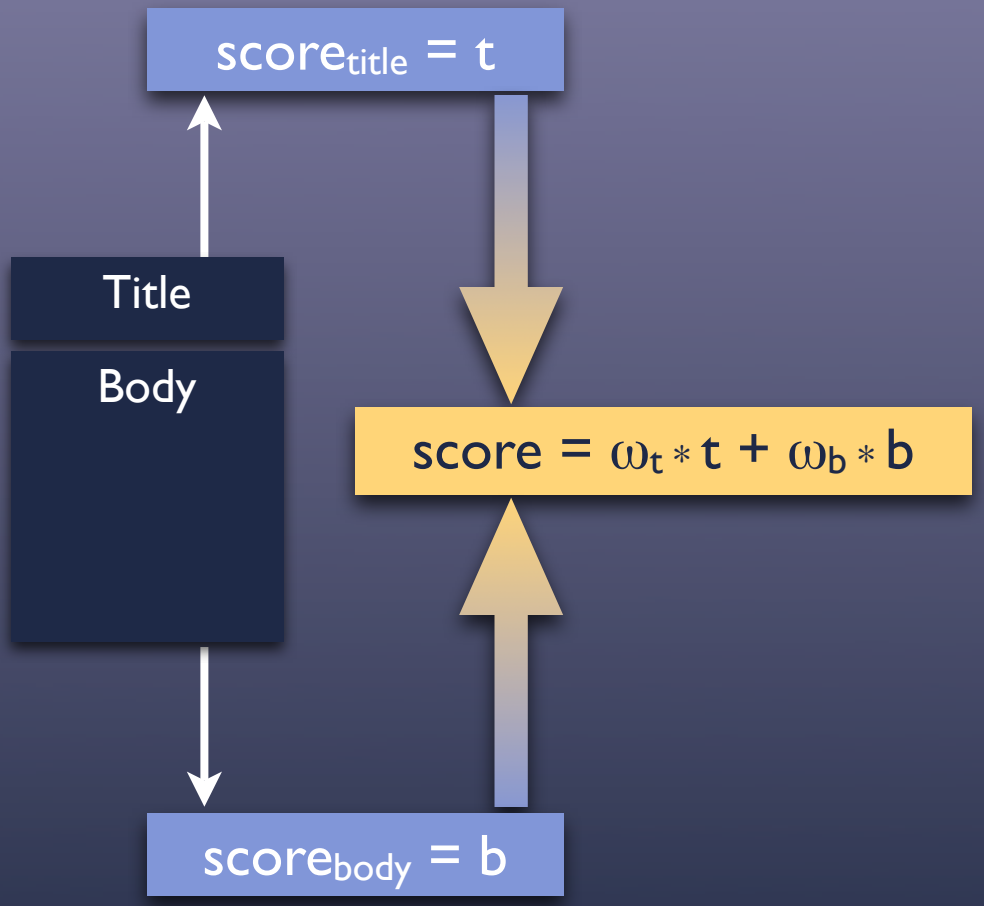
Zone di interesse



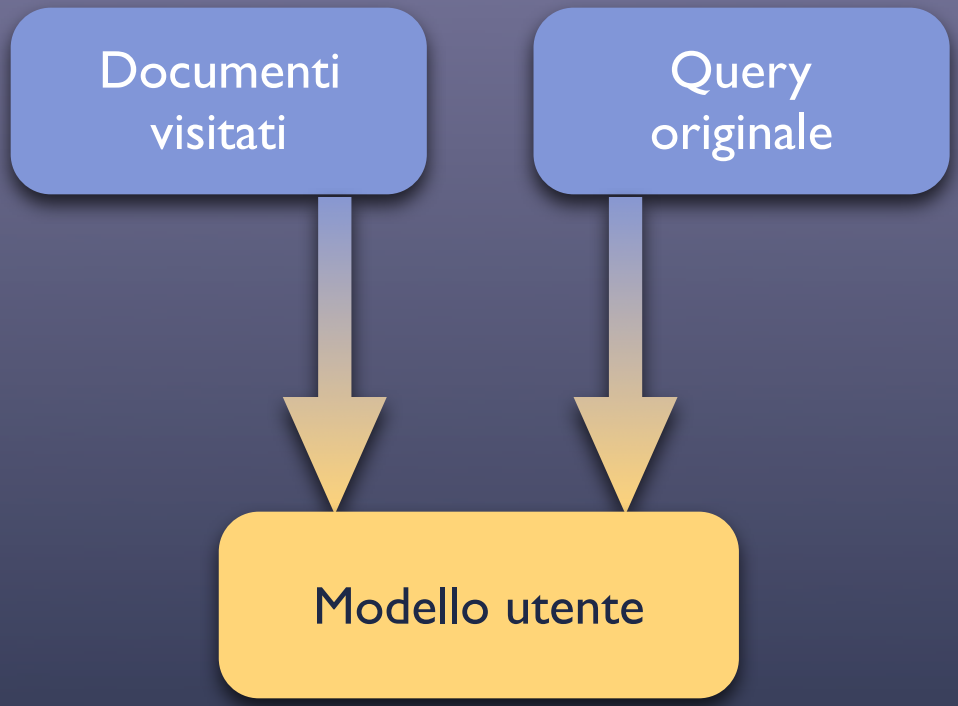
- Documento composto da più zone di interesse
  es. titolo, corpo, ...
- Creazione di un indice per ogni zona

Zone scoring

- Assegnazione di un peso ad ogni zona
- Score del documento calcolato combinando i punteggi delle zone



Modello utente



- Documenti visitati di recente rappresentati dai loro termini più significativi
- L'utente sceglie se includere o meno la query originale di ricerca

Keywords

- Termini che rappresentano meglio il documento
- Documento rappresentato solo dalle sue keyword
- Termini del documento con tf · idf molto alto

	Originale	Keyword
Butters	10.4	0
Cartman	25.0	25.0
Clyde	6.1	0
Craig	2.6	0
Jimmy	2.9	0
Kenny	15.4	0
Kyle	32.7	32.7
Stan	28.1	28.1
Token	8.4	0
Timmy	6.2	0
Tweek	1.5	0

Combinazione di score

- Score finale di un documento è una combinazione di più score parziali pesati:
  - Score parziale ottenuto calcolando la similarità del vettore documento con un vettore di keyword del modello
  - Eventuale score originale

Considerazioni finali

Perché Python e non ...

- ... PHP?
  - Motore di ricerca mantenuto in memoria costantemente dal webserver
    (con PHP in teoria è possibile, ma ...)
  - Processo computazionalmente efficiente
    (PHP meno efficiente)

Perché Python e non ...

- ... C (ANSI)?
  - Gestione immediata di: stringhe, liste, tabelle hash, persistenza, webserver
    (con C è un pelo più difficile)
- Ma C è computazionalmente più efficiente
  - Vero, ma la raccolta di dati non è particolarmente grande, quindi l'effettivo vantaggio è minimo

Perché Python e non ...

- ... Java?
  - ... dai ...

Sviluppi futuri

- Indicizzazione dei frammenti di codice con clustering a-posteriori
- Migliore selezione delle keywords
- Calcolo dei pesi per l'espansione della query mediante metodi di machine learning
- Valutazione del sistema utilizzando apposite raccolte di documenti
- Attivazione di link ipertestuali sulle parole del testo

Utile lettura

C. D. Manning, P. Raghavan and H. Schütze
Introduction to Information Retrieval
Cambridge University Press. 2008.