



# Good Search

Come trovare quello che ti serve,  
esattamente quando ne hai  
bisogno.

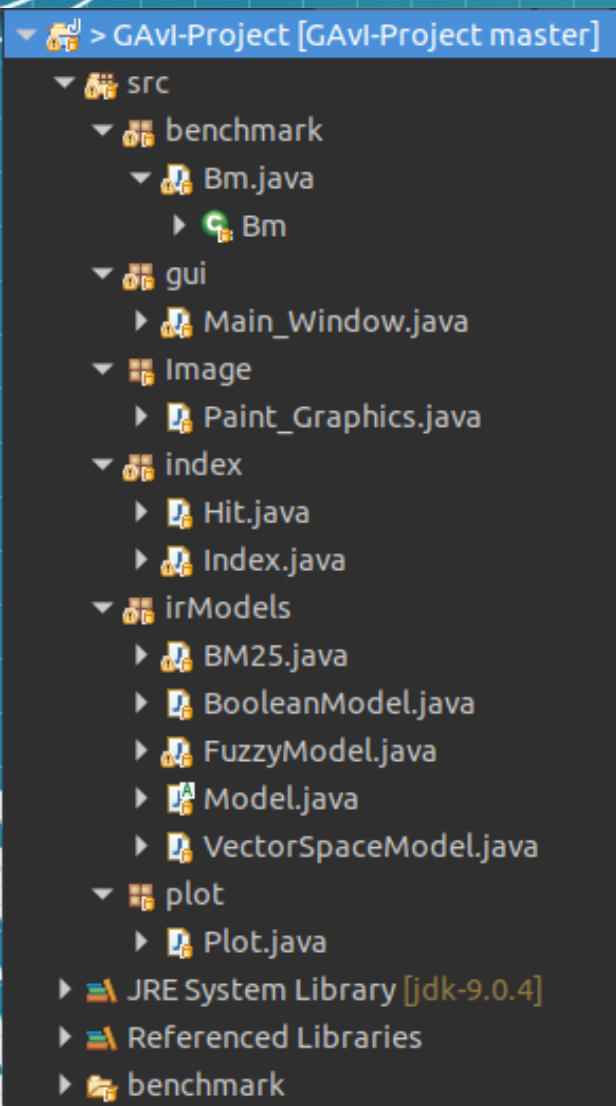
Luca Gherardini  
Riccardo Piccolo  
Simone Mione

# Creazione del progetto



Luca Gherardini (Engine)	Riccardo Piccolo (Front-end)	Simone Mione (Performance measurement)
Sviluppo dei modelli di information retrieval, con meccanismi di elaborazione della query dell'utente.	Realizzazione interfaccia grafica.	Sviluppo del benchmark (LISA).
Creazione e gestione dell'indice.	Funzioni a supporto dell'indice (caricamento/salvataggio dei documenti e dell'intero indice da interfaccia grafica).	Plot dei risultati del benchmark, mostrati tramite interfaccia grafica.

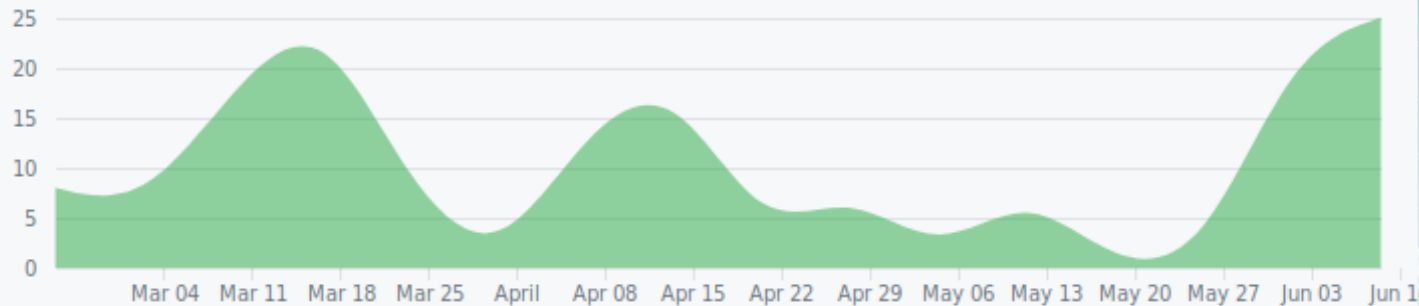
# Dimensione del progetto



Feb 25, 2018 – Jun 10, 2018

Contributions: **Commits**

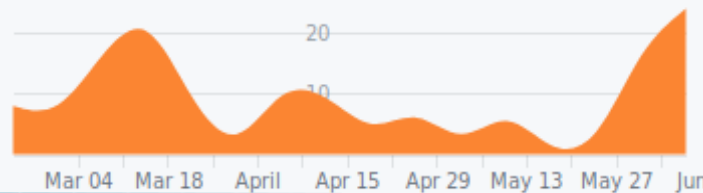
Contributions to master, excluding merge commits



**LucaGherardini**

#1

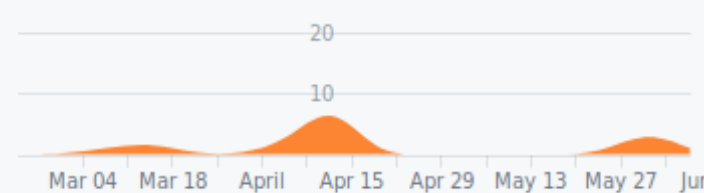
148 commits 143,759 ++ 200,602 --



**mions1**

#2

19 commits 210,485 ++ 12,050 --





# Scopo del progetto

- ✓ Realizzare un sistema di Information Retrieval basato su Lucene.
- ✓ Facilità di utilizzo e strumenti a supporto di ricerche avanzate.
- ✓ Vari modelli di IR implementati, per dare flessibilità alla ricerca.
- ✓ Analisi delle prestazioni di Lucene attraverso un benchmark (LISA), consultabile anche graficamente.
- ✓ Gestione grafica dei documenti (aggiunta e rimozione di singoli documenti, di intere gerarchie di cartelle, salvataggio e caricamento dell'indice su disco)

# Lucene, il “motore”



Adottando il “full-featured search engine library” Lucene, abbiamo potuto definire i meccanismi e le strutture sui quali si basa la nostra applicazione, quali:

- Modelli di IR utilizzati
- Indice dei documenti
- Indicizzazione dei documenti
- Parsing delle query



# Modelli di IR



I modelli di Information Retrieval utilizzati sono:

- Modello Booleano
- Modello Fuzzy
- Modello Vettoriale
- Modello Probabilistico (BM25)

Ogni modello ha la funzione di effettuare il parsing della richiesta dell'utente, creando un oggetto Query da utilizzare nella ricerca sull'indice. Ogni modello è inoltre fondamentale in quanto determina la similarità adottata dal sistema, che influenzerà la rilevanza (e quindi il ranking) di ogni documento.

# L'Indice



L'indice contiene l'insieme dei documenti sul quale il sistema cercherà i risultati rilevanti per l'utente (Lucene sfrutta l'inverted indexing per indicizzare). I documenti sono prevalentemente di tipo testuale (plain text, .txt).



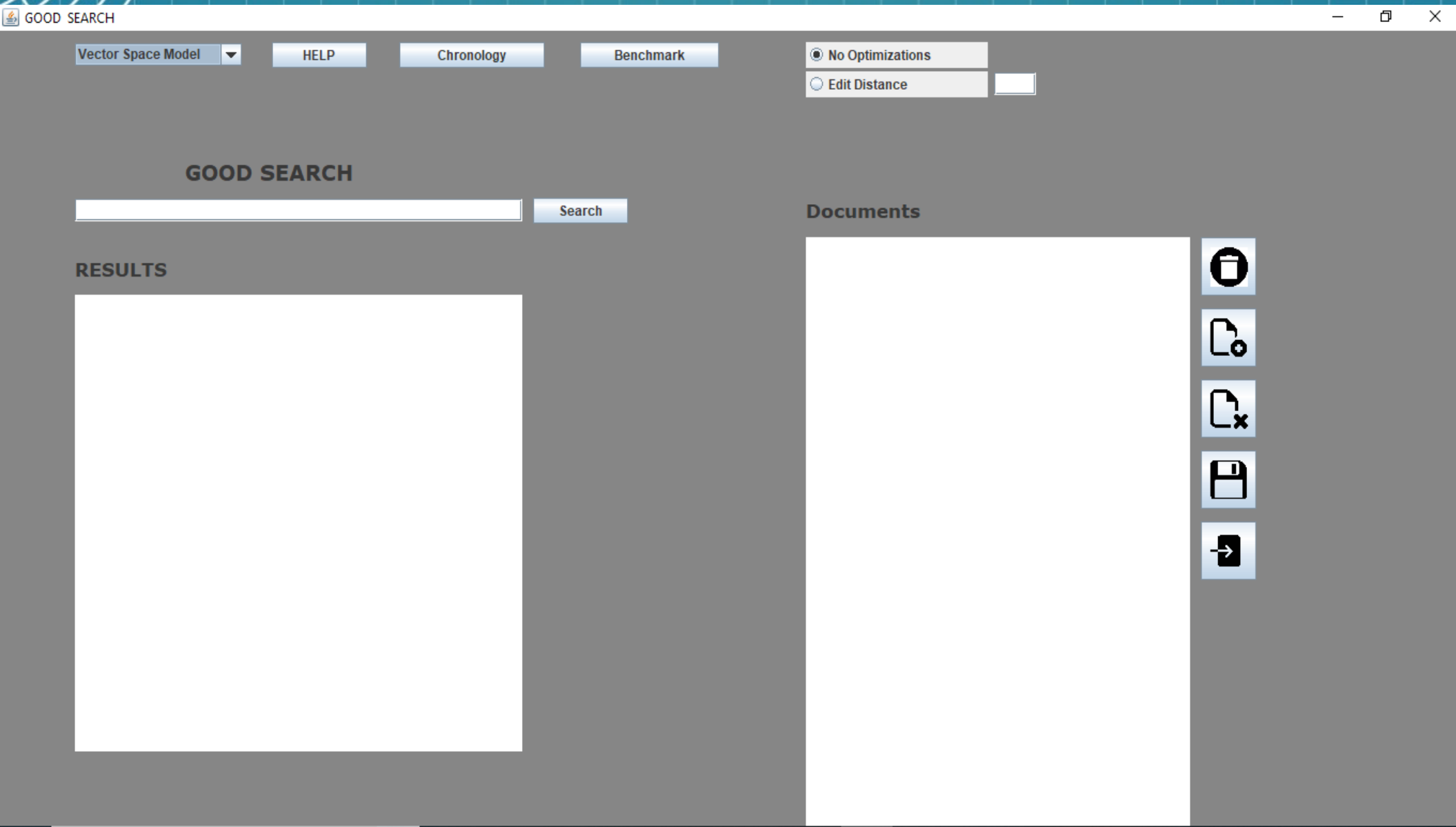
# Interfaccia grafica



- L'interfaccia utente è stata pensata per essere la più semplice e completa possibile, per permettere una ricerca agevole.
- La schermata principale dà la possibilità all'utente di:
  - Scegliere il modello di ricerca ed eventuali ottimizzazioni.
  - Visualizzare la cronologia delle ultime ricerche effettuate.
  - Visualizzare i documenti su cui si desidera effettuare la ricerca.
  - Visualizzare i risultati ottenuti
  - Fare Benchmark.



# Interfaccia grafica: come si presenta



Vector Space Model ▼

HELP

Chronology

Benchmark

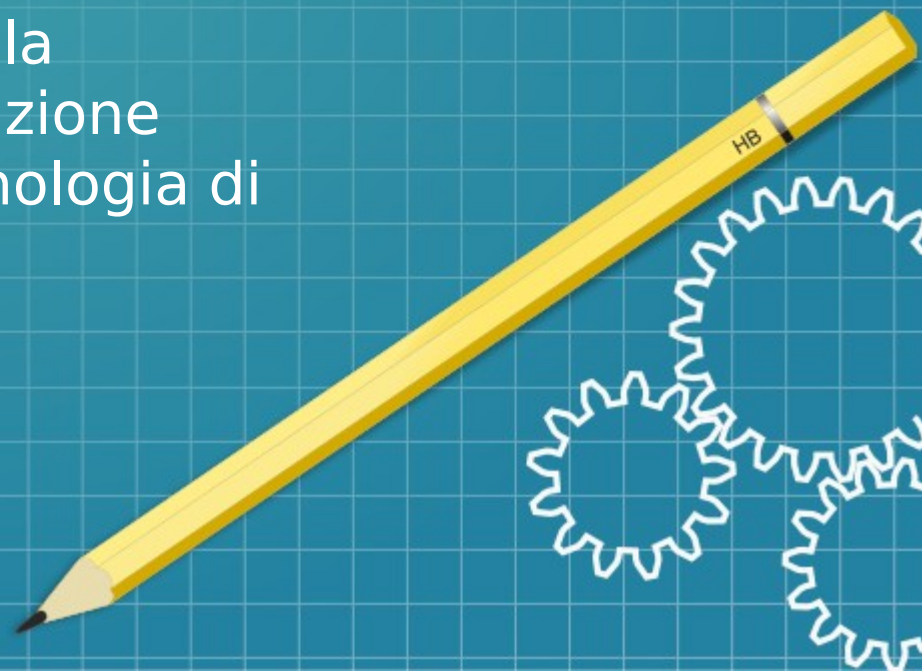
☒ No Optimizations☐ Edit Distance 

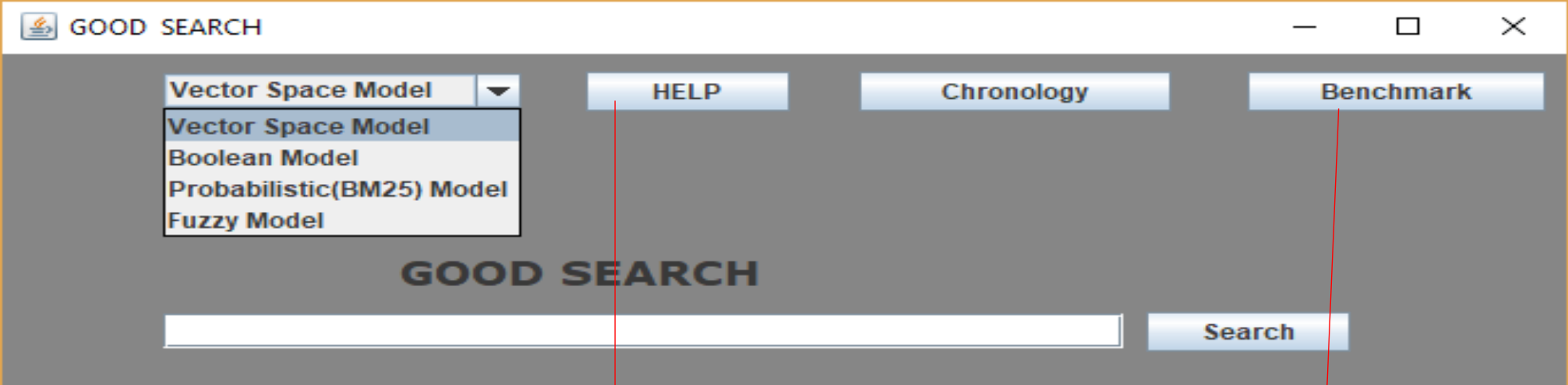
I Modelli implementati sono:

- ♦ Vector Space Model
- ♦ Boolean Model
- ♦ Probabilistic Model
- ♦ Fuzzy Model

Permette la Visualizzazione della Cronologia di Ricerca

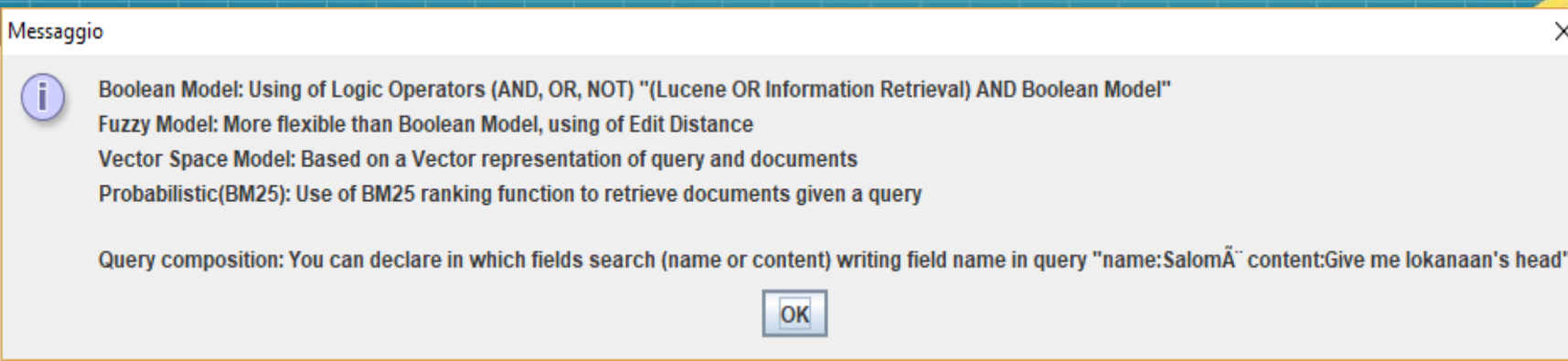
L'utente ha la possibilità di selezionare o meno le ottimizzazioni da attuare sulla ricerca





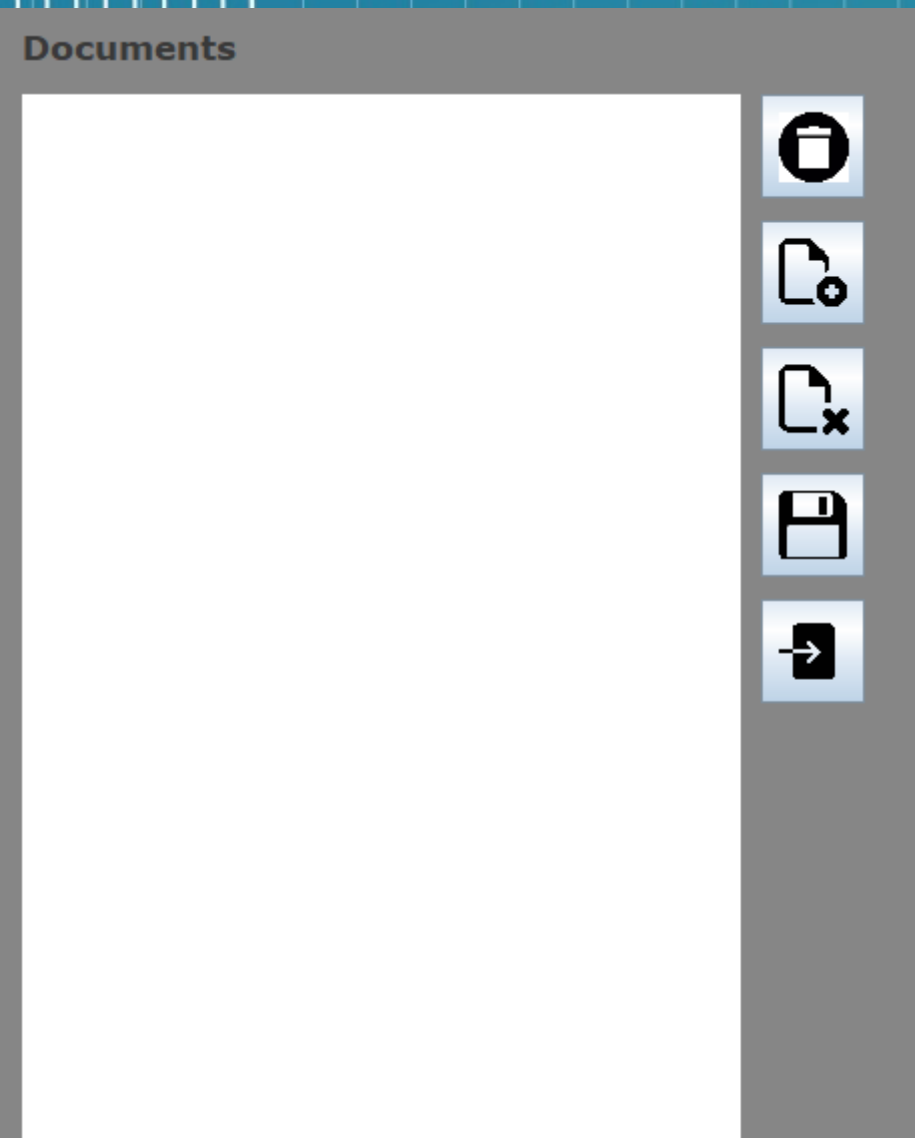
Aiuta l'utente a scrivere le Query in riferimento al Modello prescelto.

Dà la Possibilità all'utente di testare il Programma con un Benchmark precaricato (Lisa) e di visualizzarne i risultati.





# Uno “sguardo” all’indice



Nella sezione Documents sono visualizzati tutti i documenti contenuti nell’indice, sui quali viene effettuata la ricerca. L’utente può:

- ✓ Eliminare tutti i documenti presenti nella tabella
- ✓ Aggiungere un singolo file, una selezione multipla di file e anche intere cartelle
- ✓ Eliminare un singolo documento
- ✓ Salvare i documenti in un file
- ✓ Caricare documenti da file






## GOOD SEARCH

## RESULTS

Nella sezione Results sono visualizzati tutti i documenti recuperati con i rispettivi score per la query ricercata.



# Benchmark: istruzioni per l'uso

La classe IRBenchmark si occupa di eseguire e mostrare i risultati del benchmark, attraverso i seguenti passi:

- 1. Caricamento dei documenti nell'indice.
- 2. Esecuzione delle query tramite il modello selezionato e ottenimento delle intersezioni tra i documenti recuperati e quelli attesi.
- 3. Calcolo della precision e della recall
- 4. Plot dei grafici risultanti



# Benchmark



Il Benchmark LISA è composto da tre file, contenenti l'insieme dei documenti, delle query e dei documenti attesi. Abbiamo diviso i documenti in singoli file, per facilitare il loro inserimento all'interno dell'indice.

Le query sono state divise in singoli file, e vengono parsate dal modello usato dalla classe Benchmark.

Si misurano poi, per i livelli standard (0.33, 0.66, 1.0), la precision e la recall per ogni query, la R precision e la AVG precision.

Abbiamo scelto di mostrare all'utente solo la AVG-Precision per ogni livello standard, la R-precision per i valori di  $R = 5, 10, 15$  che sono, grossomodo, i primi risultati osservati dagli utenti, e la recall per ogni query.



# Strumenti esterni per il Benchmark

- LISA (Benchmark)  
[http://ir.dcs.gla.ac.uk/resources/test\\_collections/](http://ir.dcs.gla.ac.uk/resources/test_collections/)
- Plot (libreria per il plotting da java)  
<http://yuriy-g.github.io/simple-java-plot/>



# Benchmark: conclusioni

Le conclusioni osservabili dai dati del benchmark sono che il nostro sistema recupera quasi sempre tutti i risultati rilevanti, nonostante ne recuperi anche molti non rilevanti.

