

GIOVANNI GAMBIGLIANI ZOCCOLI
HELMİ ANDREA
RAUSA GRETA

Hegregio

LUCENE BASED
INFORMATION RETRIEVAL TOOL

Caratteristiche

- Libreria: Lucene 7.4
- Dataset: TREC 2016

Features

- Tolerant retrieval
- Possibilità di ordinamento dei risultati secondo diversi modelli di information retrieval
- Valutazione dell'efficacia

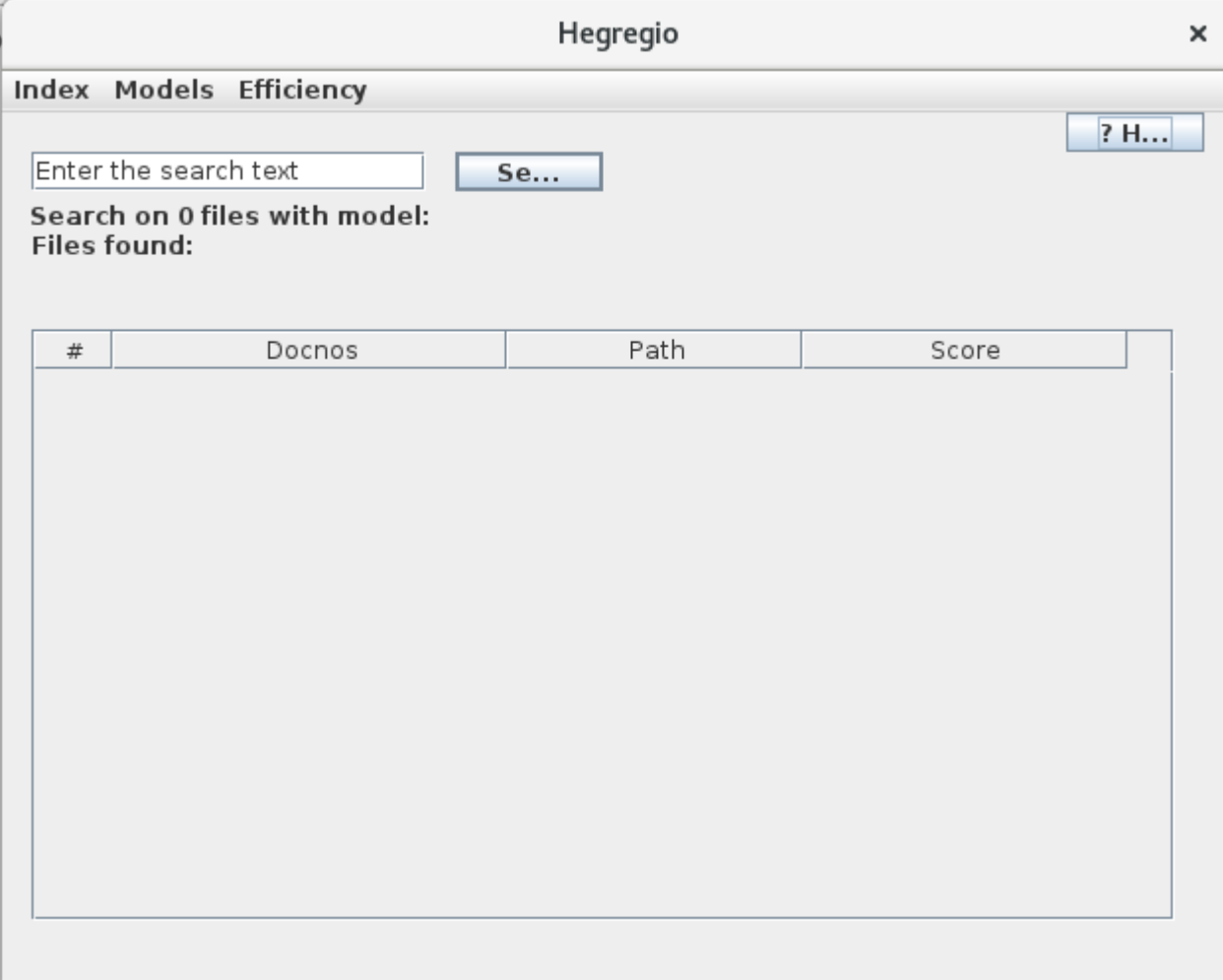
SCOPO DEL PROGETTO

Realizzazione di un sistema di information retrieval utilizzando i documenti forniti da TREC Precision Medicine.

Più di un 1 miliardo di documenti disponibili in formato nxml.

30 query con relativi documenti rilevanti per effettuare dei benchmark.

HEGREGIO: COME SI PRESENTA



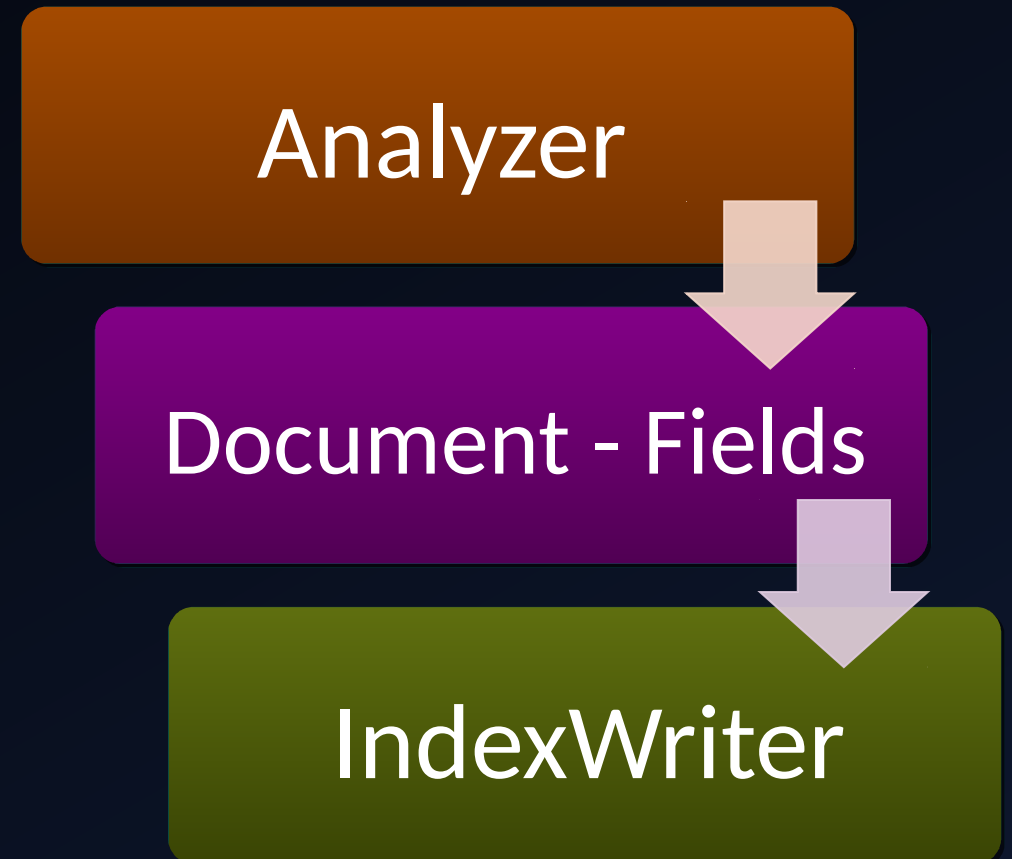
The screenshot shows a window titled "Hegregio" with a close button (X) in the top right corner. Below the title bar is a tabbed interface with three tabs: "Index", "Models", and "Efficiency". The "Index" tab is currently selected. In the top right corner of the window, there is a button labeled "? H...". Below the tabs, there is a search input field with the placeholder text "Enter the search text" and a button labeled "Se...". Below the search field, the text "Search on 0 files with model:" and "Files found:" is displayed. At the bottom of the window, there is a table with four columns: "#", "Docnos", "Path", and "Score". The table is currently empty.

#	Docnos	Path	Score
---	--------	------	-------

Lucene information retrieval: come funziona?

Indicizzazione

- Definire un modulo di preprocessing (classe Analyzer)
- Per ogni documento crea un oggetto Document e gli aggiunge i Field scelti
- Creare un indexwriter e aggiungere all'indice i campi precedentemente riempiti



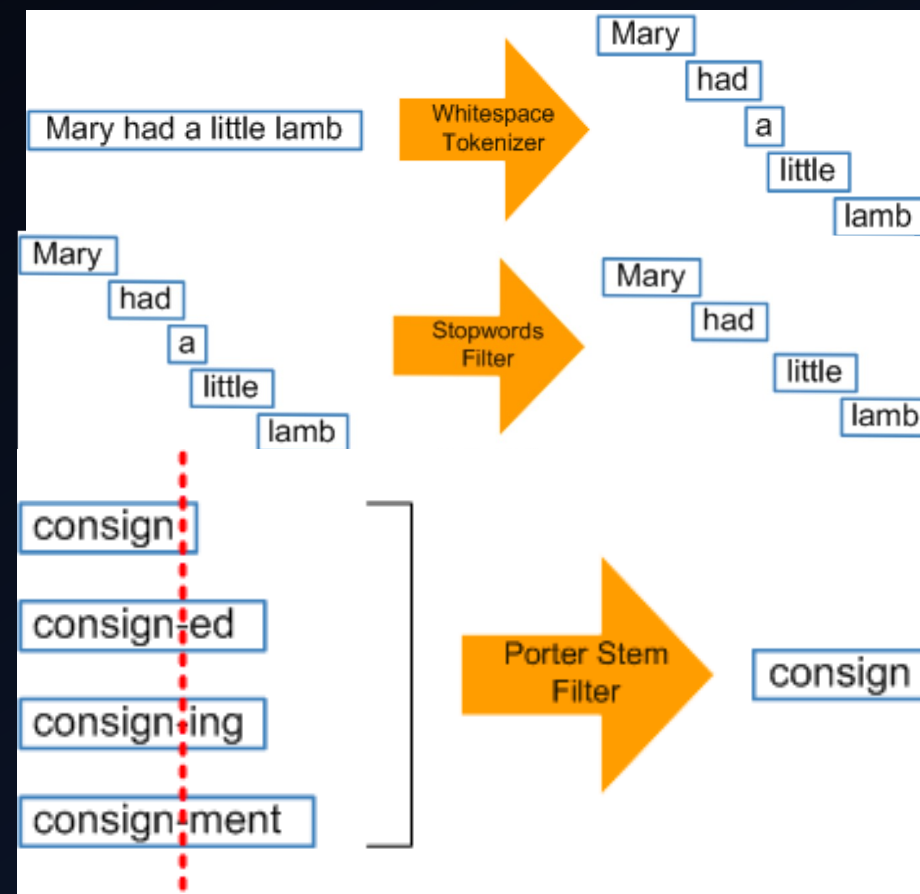
COSA SONO GLI ANALYZER

- Gli Analyzer sono usati sia in fase di indicizzazione, sia in fase di interrogazione. Esaminano il testo dei campi e generano un flusso di token. Possono essere una singola classe o possono essere composti da una serie di classi di filtro e tokenizer.
- I tokenizer spezzano i dati del campo in unità lessicali o token.
- I filtri esaminano un flusso di token e li mantengono, trasformano o eliminano. Tokenizzatori e filtri possono essere combinati per creare Analyzer e l'output risultante di un analizzatore viene utilizzato per confrontare risultati di query o indici di creazione.

COME ABBIAMO ANALIZZATO LE STRINGHE

L'analyzer utilizzato sia in fase di indicizzazione che in fase di ricerca è lo Standard Analyzer, ed è stato customizzato con l'aggiunta dei seguenti componenti:

- Tokenizer:
 - StandardTokenizer
- Filter:
 - StopFilter (customizzato)
 - porterStemFilter



CREAZIONE DI UN DOCUMENTO

Un documento su Lucene viene inteso come insieme di campi (coppie chiave-valore) che possono essere analizzati e indicizzati.

La scelta di tali campi è fondamentale per la creazione dell'indice. Nel nostro caso sono stati scelti i seguenti campi per ogni file da indicizzare:

- CONTENTS (I dati su cui avviene la ricerca)

È stato necessario un ulteriore preprocessing dato il formato dei documenti: NXML)

- FILE_NAME (Il nome del file, da visualizzare nei risultati)
- FILE_PATH (Per la visualizzazione del documento da parte dell'utente)

CREAZIONE DI UN INDEXWRITER

L'IndexWriter è un componente di Lucene che crea e modifica l'indice nella fase di creazione (aggiunge i documenti all'indice).

Viene inizializzato con una configurazione (IndexWriterConfig) e con il puntatore alla cartella che contiene l'indice.

L'indexWriterConfig basa il suo comportamento sulla scelta dell'Analyzer.

MODELLI

Il modello di default è il probabilistico

Dopo la definizione dell'indexWriter viene definito il modello di ranking attraverso la funzione:

```
“indexWriterConfig.setSimilarity(SIMILARITY);”
```

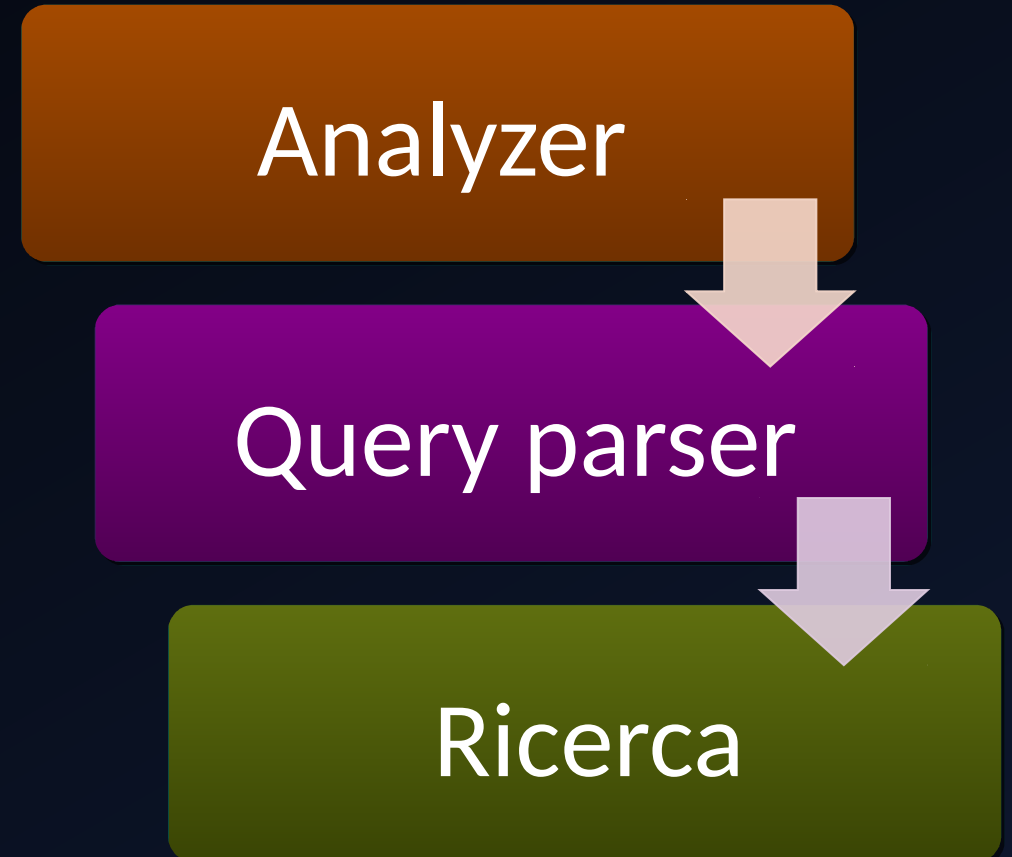
Modelli proposti:

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

Lucene information retrieval: come funziona?

Ricerca

- Definire lo stesso modulo di preprocessing usato per l'indicizzazione
- Aprire l'indice dei documenti
- Definire il parser della query (QueryParser)
- Definire il modello di query da adottare
- Ricerca dei documenti secondo il modello scelto



TOLERANT RETRIEVAL

- Spelling correction “Did you mean..?”

Viene proposta una correzione delle keyword di ricerca grazie all’ausilio di un dizionario di lingua inglese specializzato in ambito medico.

- Wildcard queries

Lucene supporta nativamente le wildcard su caratteri singoli e multipli, per ricerche per termini singoli (ma non all’interno di frasi).

Per applicare una wildcard su singolo carattere si usa il simbolo "?".

Per sostituire invece una moltitudine di caratteri si usa il simbolo "*".

BENCHMARK

- TREC mette a disposizione 30 query per l'esecuzione del benchmark e la valutazione dell'efficacia del programma.
 - Si effettua una ricerca con un dato modello
 - Si confrontano i risultati ottenuti con quelli attesi, specificati nel benchmark
- Misure utilizzate:
 - Precision & Recall
 - R-Precision

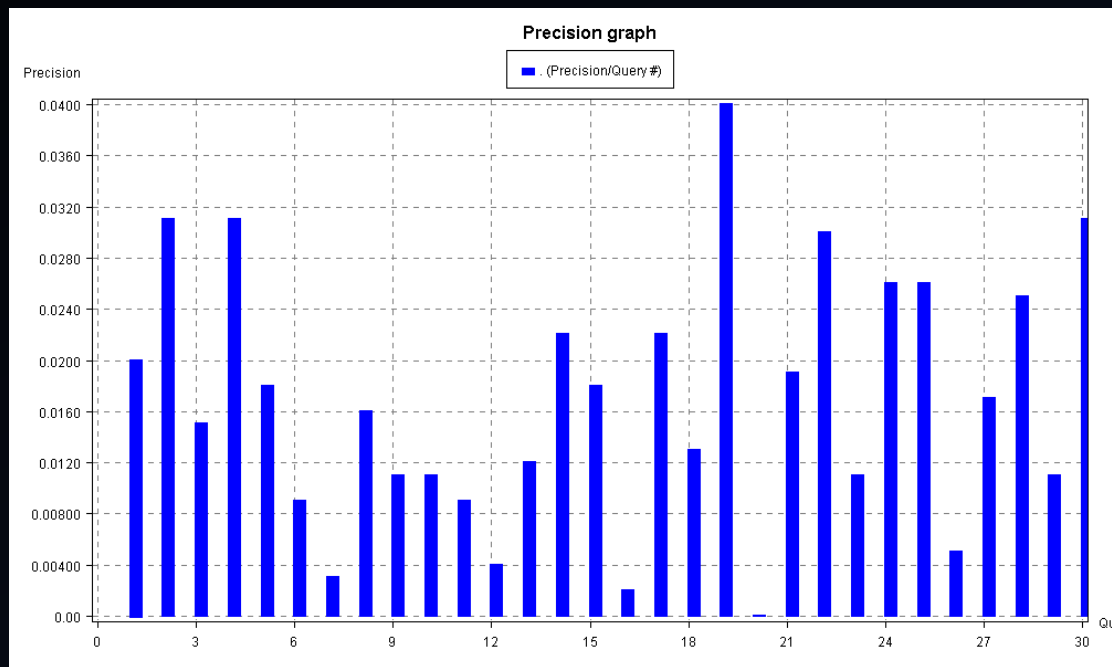
BENCHMARK

Per il calcolo delle misure sono usate funzioni e Classi native di Lucene inizializzate dalla classe `QualityBenchmark`.

- `TrecTopicsReader` (Lettura dei documenti in formato TREC dal file dei Topic)
- `TrecJudge` (Definisce che un documento sia rilevante o meno per una data `QualityQuery`, basandosi sul file `qrels` di TREC.)
- `judge.validateData` (Controlla che la query e il “giudice” si riferiscano alla stessa query)
- `qrun.execute` (Esegue il benchmark).

GRAFICO VECTOR SPACE MODEL

PRECISION



RECALL

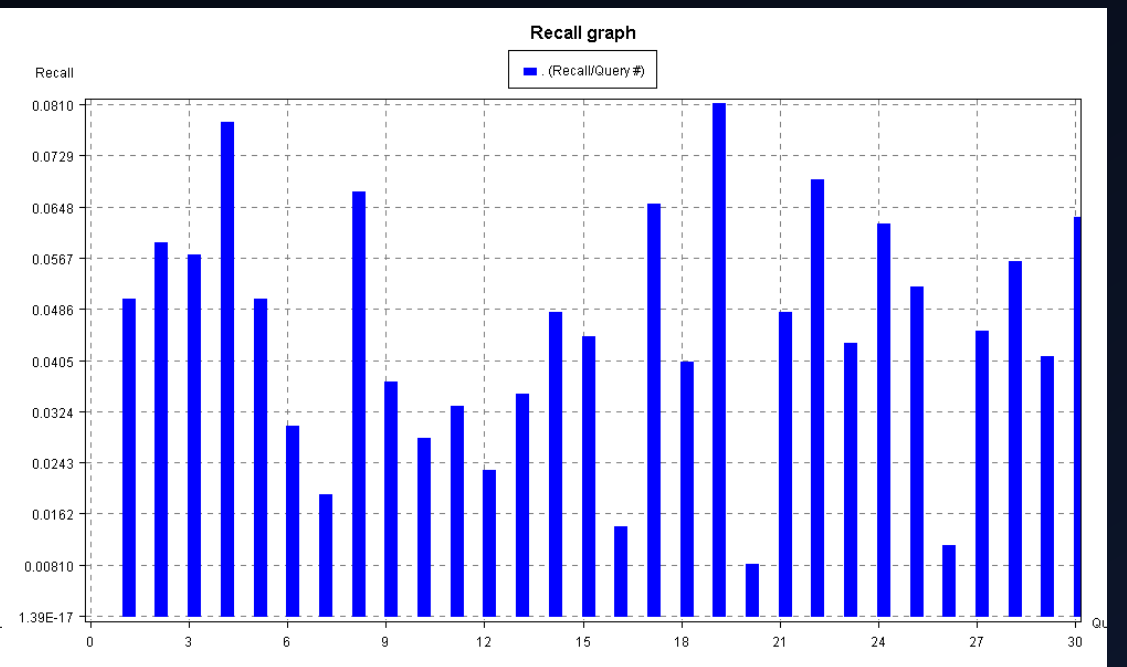
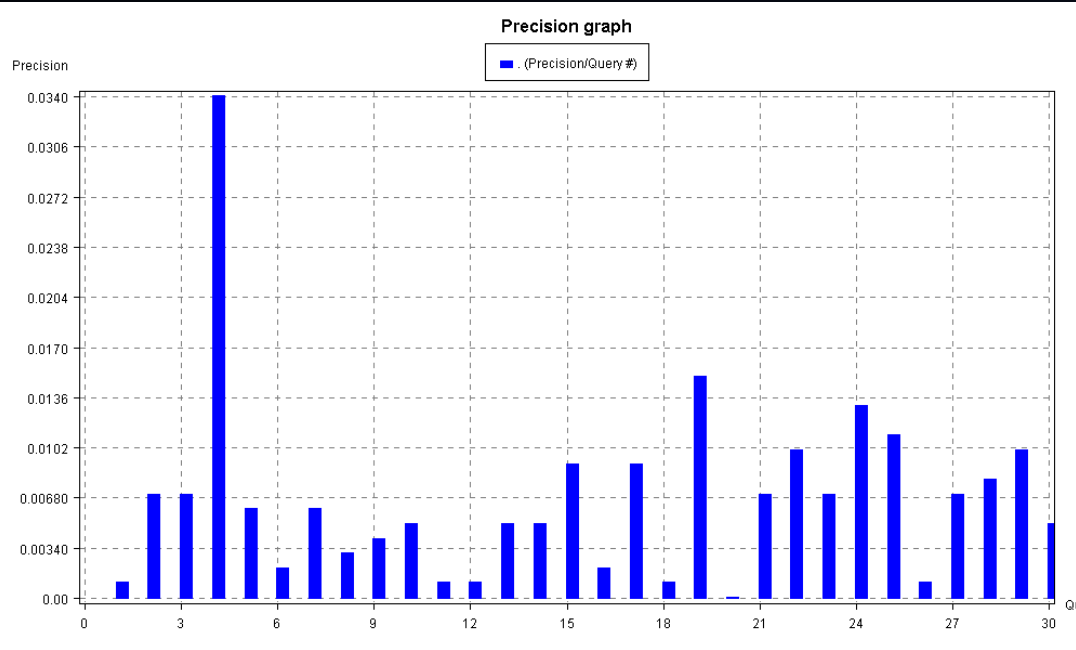


GRAFICO BOOLEAN MODEL

PRECISION



RECALL

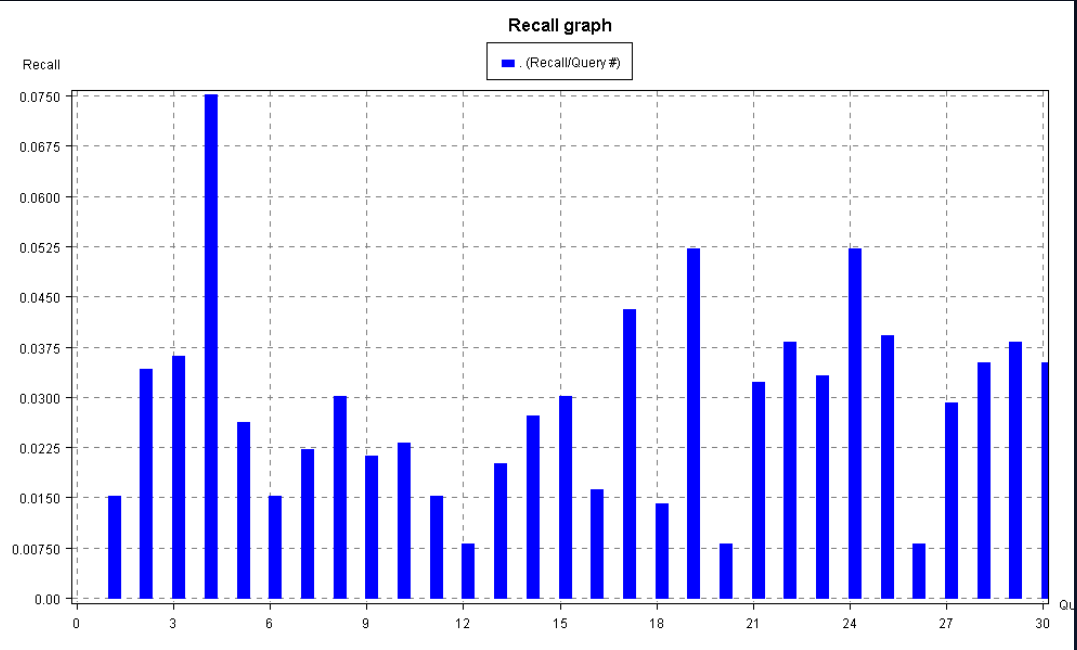
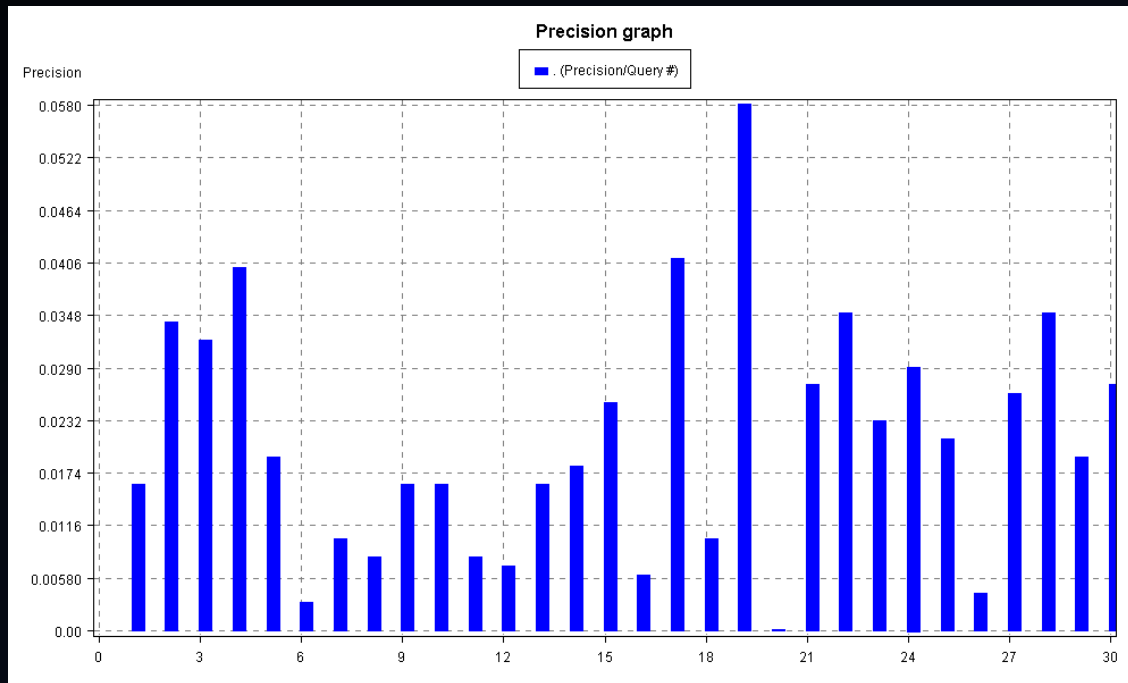


GRAFICO PROBABILISTIC MODEL

PRECISION



RECALL

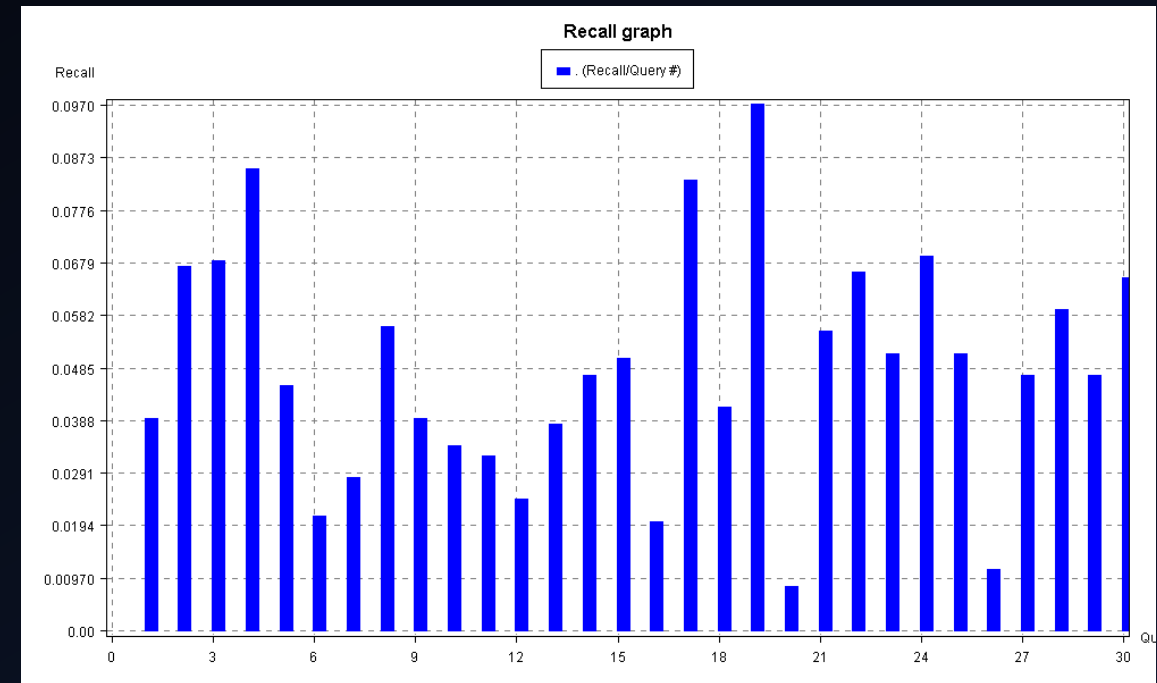
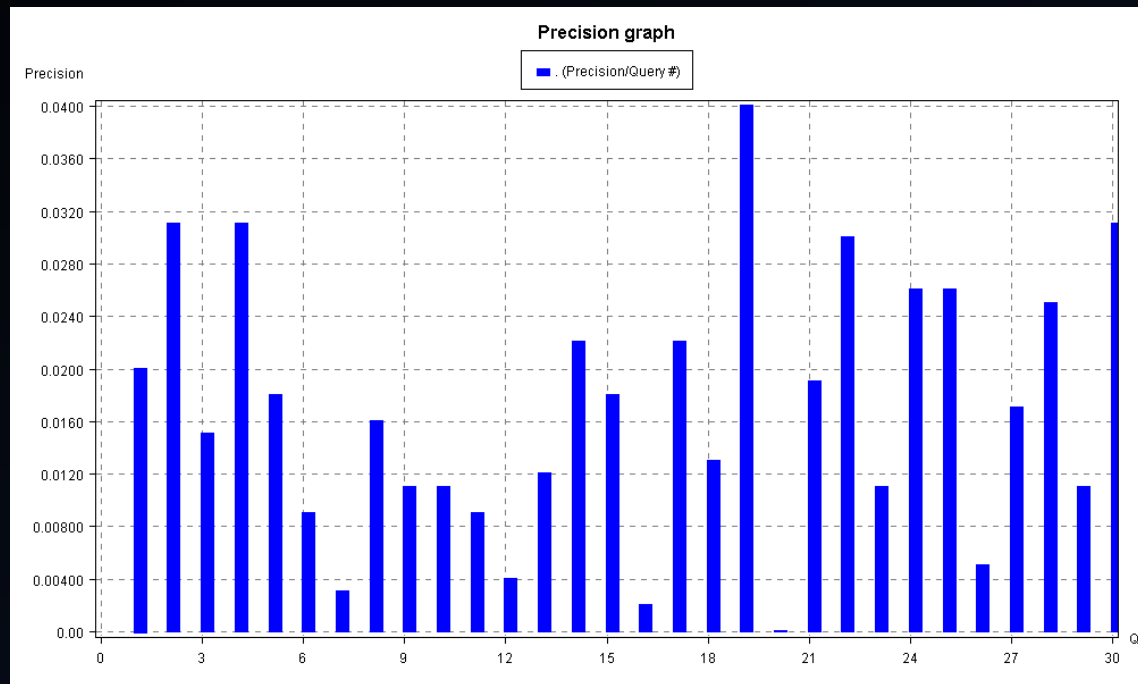
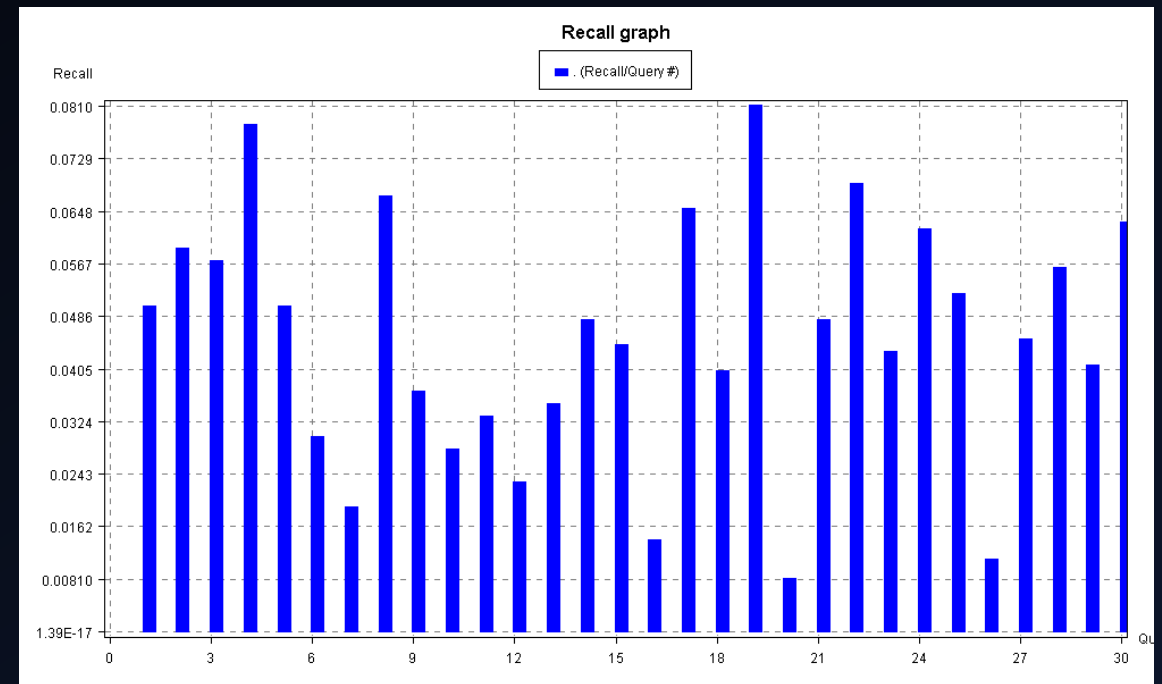


GRAFICO FUZZY MODEL

PRECISION



RECALL



R-PRECISION

È data dalla Precision ad R, dove R è il numero di documenti pertinenti per quell'interrogazione nella collezione.

