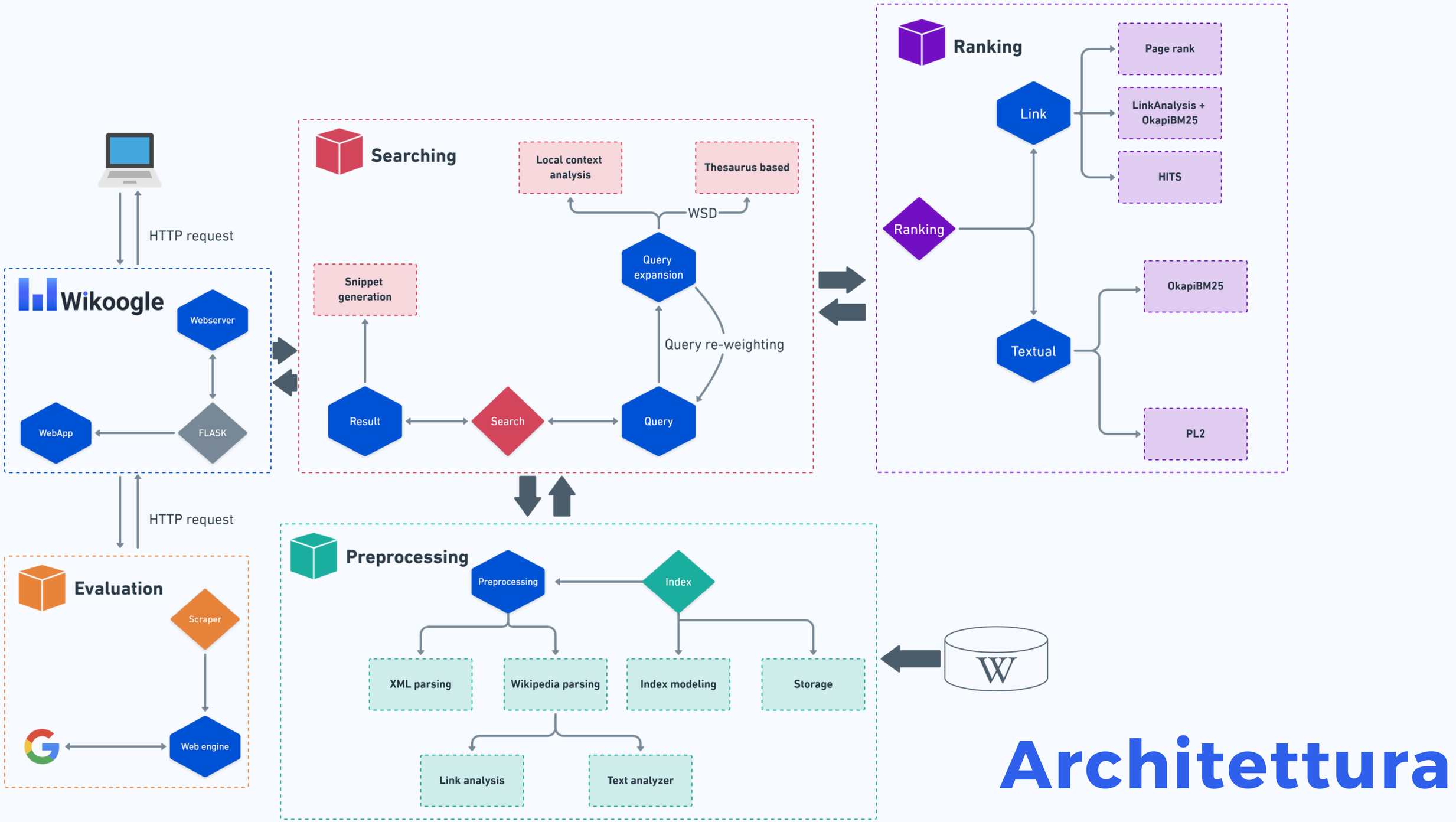
# Wikoogle

DEMO AT **http://212.237.42.43:8080/**

DEMO AT **http://212.237.42.43:8080/**

# Obiettivi

Sviluppare un **search engine** per Wikipedia che restituisca **le pagine di Wikipedia** rilevanti per una richiesta sottomessa e ordini i risultati rispetto alla **rilevanza**




# Architettura

PREPROCESSING

# Wikidump

Di seguito sono elencate le caratteristiche del dump preso in considerazione

**enwiki-20200620-pages-articles-multistream1.xml-p1p30303.bz2**

**enwiki-20200620-pages-articles-multistream2.xml-p30304p88444.bz2**

**enwiki-20200620-pages-articles-multistream3.xml-p88445p200509.bz2**

| Dump size | Index size | Total articles | Total indexed |
|---|---|---|---|
| 2.7 Gb | 2.7 Gb | 140901 | 109965 |

# XML PARSING

## Performance

Il processo di lettura e estrapolazione delle informazioni necessarie deve essere veloce

## Memory

Il consumo di memoria deve essere il più efficente possibile.

## Big files

Il processo deve potere gestire file di qualsiasi dimensione

# XML PARSING

## Performance

Il processo di lettura e estrapolazione delle informazioni necessarie deve essere veloce

> *The lxml XML toolkit is a Pythonic binding for the C libraries libxml2 and libxslt. It is unique in that it combines the speed and XML feature completeness of these libraries with the simplicity of a native Python API*
>
> https://lxml.de

# XML PARSING

## lxml 

Legge tutto il file e crea l'albero xml in memoria con disastrose complicazioni di performance

## Memory

Il consumo di memoria deve essere il più efficente possibile.

## etree.iterparse

L'idea è quella di leggere il file iterativamente, in questo modo il file non viene caricato tutto in memoria.

```
from lmxl import etree

etree.iterparse(path, events=('end',), tag=self._base_tag,
huge_tree=True)
```

# XML PARSING

## Big files

Il processo deve potere gestire file di qualsiasi dimensione

## etree.iterparse ⚠

Il metodo tiene comunque i riferimenti ai nodi dell' file xml. e questi non vengono rilasciati. Con file grandi si ha un memory leak notevole e quindi é necessario togliere i riferimenti ogni volta che si finisce di parsare un nodo

```
from lxml import etree


context = etree.iterparse(path, events=('end',), tag=self._base_tag,
huge_tree=True)
    for _, root in context:
        yield root
        while root.getprevious() is not None:
            del root.getparent()[0]
        root.clear()
```

# WIKIPEDIA PARSING

## Wikipedia dump

```
{{short description|political movement that is skeptical towards authority and
rejects involuntary, coercive hierarchy}}
{{redirect2|Anarchist|Anarchists|other uses|Anarchists (disambiguation)}}
{{pp-move-indef}}
{{good article}}
{{use dmy dates|date=March 2020}}
{{use British English|date=January 2014}}
{{anarchism sidebar}}
{{basic forms of government}}
'''Anarchism''' is a [[political philosophy]] and [[Political movement|movement]]
that rejects all involuntary, coercive forms of [[hierarchy]]. It [[Radical
politics|radically]] calls for the abolition of
```

PREPROCESSING

# WIKIPEDIA PARSING

## Perchè

### Index

Il markup di wikipedia non è utile ai fini dell' indicizzazione

### Link analysis

Analizzando l'abstract syntax tree è possibile ottenere informazioni sul testo e interpretarlo conseguentemente

### Text extraction

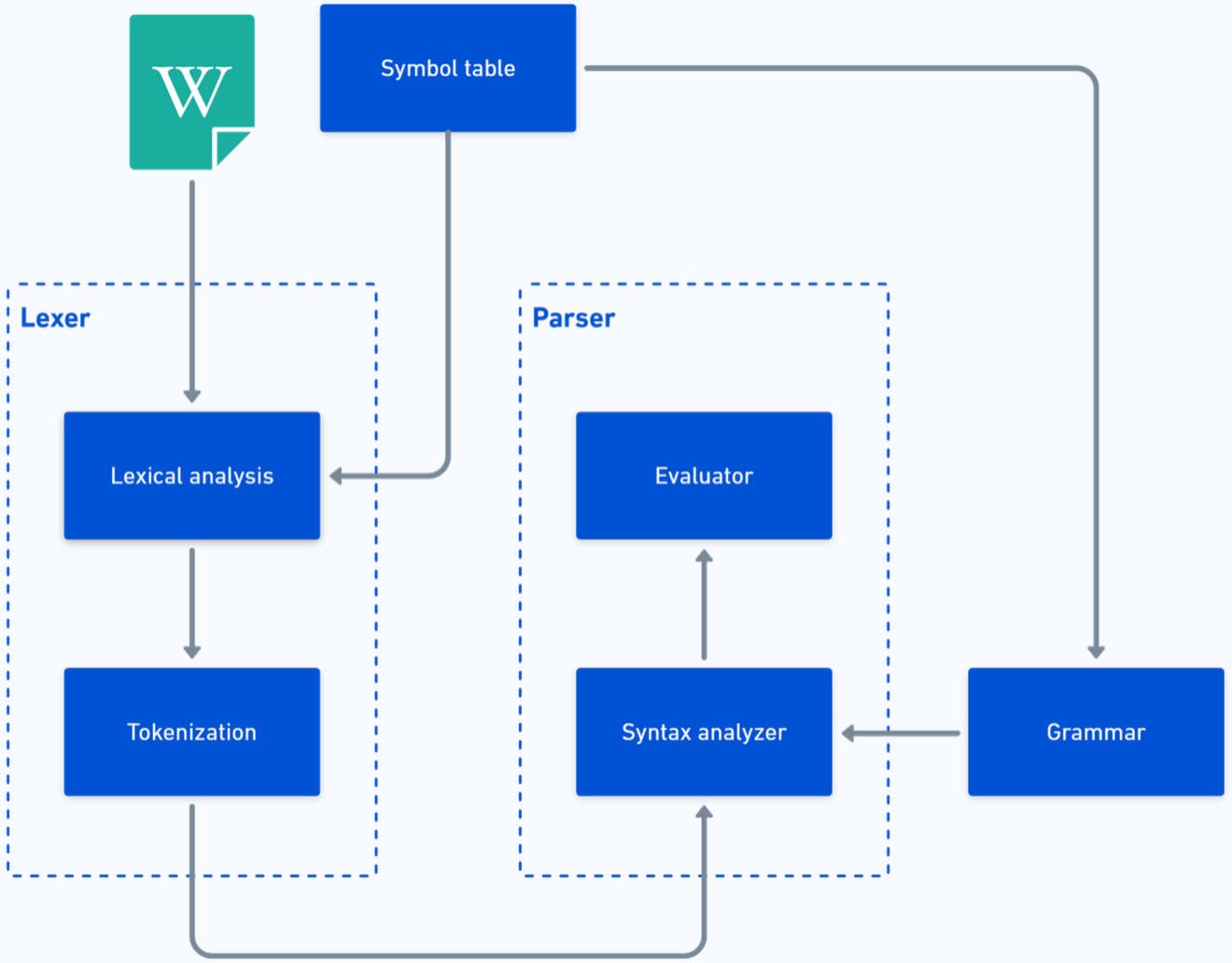
La risoluzione del markup permette di ottenere un testo "pulito"

PREPROCESSING

# Recursive descent parser

Ogni articolo wikipedia viene compilato seguendo precise regole grammaticali

# WIKIPEDIA PARSING



PREPROCESSING

# Markup spec

[[link|text]]

{{template|param}}

#REDIRECT

=heading=

==heading2==

===heading3===

===heading4===

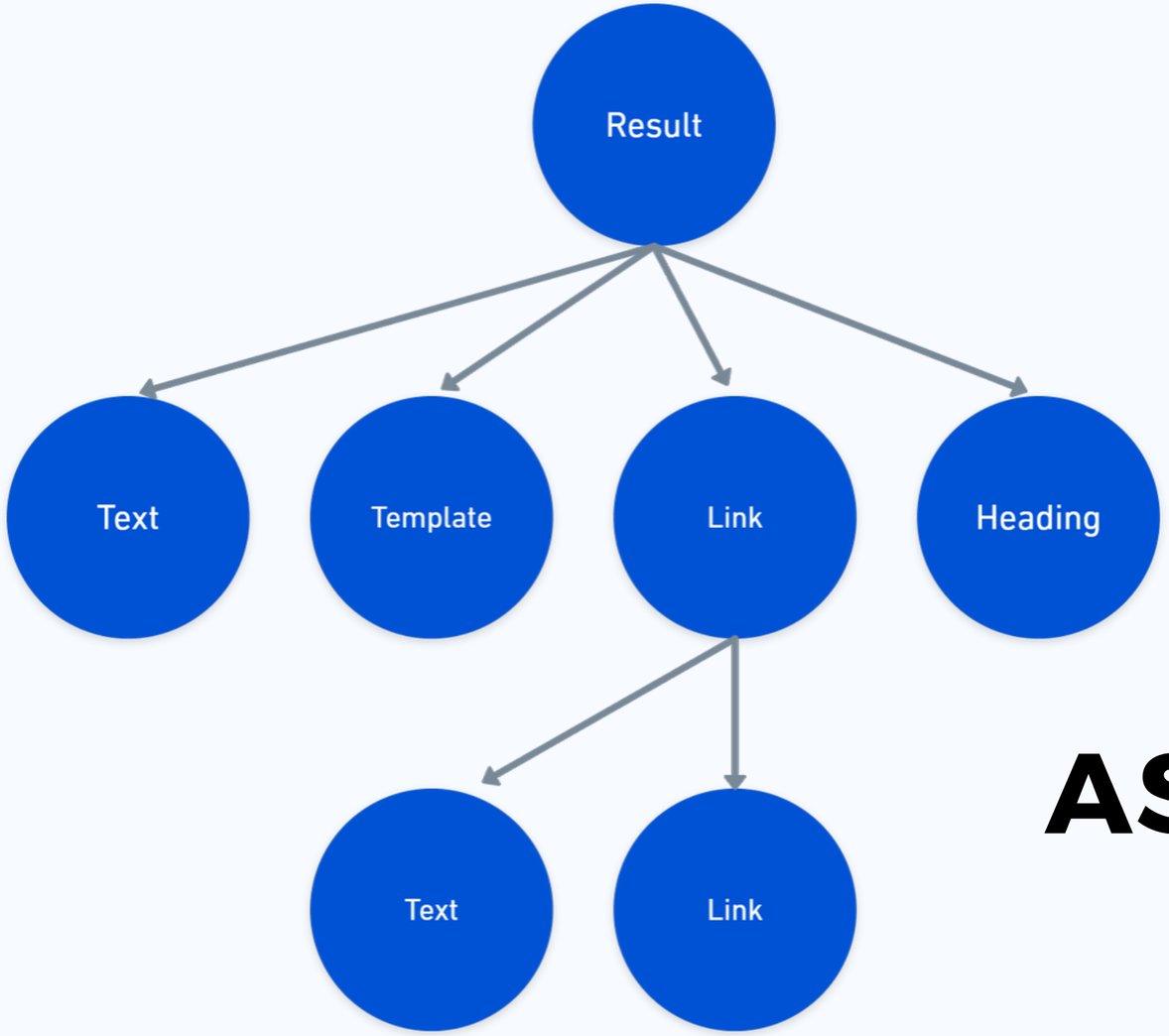
.....

WIKIPEDIA PARSING

PREPROCESSING

# Evaluation

## WIKIPEDIA PARSING




**AST**

La fase di evaluation perlustra l'albero e ogni nodo compila il proprio sottoalbero

PREPROCESSING

# WIKIPEDIA PARSING

## Statistiche

**con 667 MB dump**

Indice senza parsing wikipedia: 1.2GB

Indice con parsing wikipedia: 599MB

PREPROCESSING

# Text pipelines

**Text analyzer**

Bisogna trattare il testo prima dell'indicizzazione o durante la fase di query prepocessing

PREPROCESSING

# Whoosh

# Index modeling

## Title

L'indice deve poter ritornare un documento in base al titolo del documento. Il titolo inoltre è parte dell'URL del documento

## Text

Il testo del articolo è indicizzato e viene salvato un riferimento al testo intero per potere fare ad esempio Snippet Generation o Query expansion

## Categories

Dagli articoli vengono indicizzati anche le categorie a cui appartengono

# Query

Le query sono sottoposte al text prepocessing (text pipeline) come nella fase di indicizzazione

**field: term | "phrase" | wildcard**

Il query language supportato è
quello di default di Whoosh

whoosh.readthedocs.io/en/latest/querylang.html

# Snippet generation

## Perchè

### User

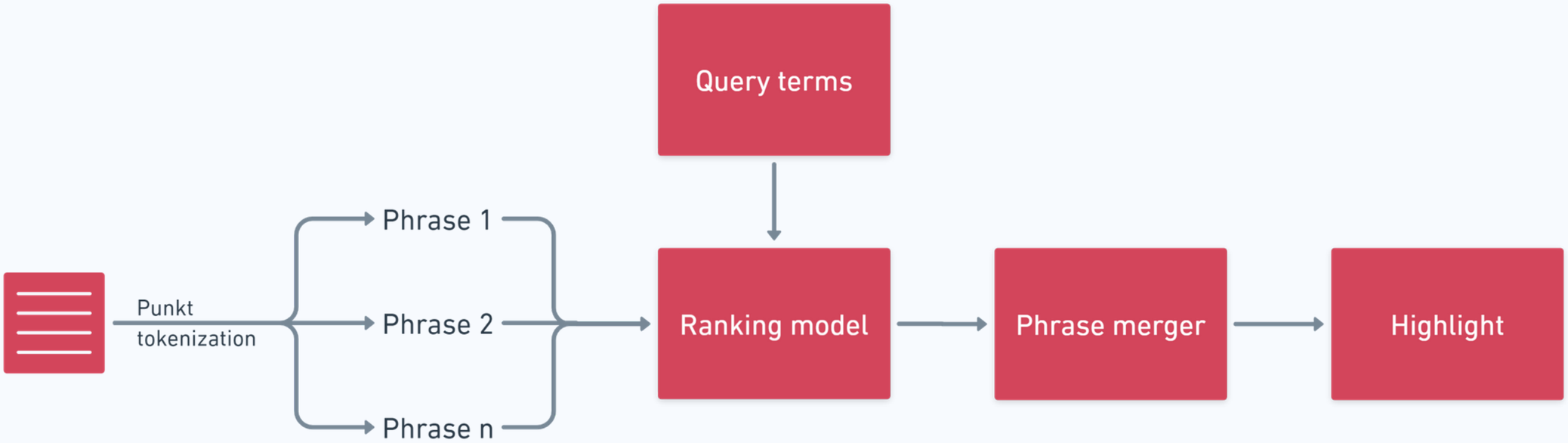
L'utente può identificare se il documento è rilevante o no evitando di visualizzarli tutti uno ad uno

### Query expansion

Local context analysis (trattato in seguito)

# Snippet generation

## L'algoritmo **Query-dependent**



# Snippet generation

## Ranking model

$$w_p = \begin{cases} 2 & \text{p is first line} \\ 1 & \text{p is second line} \\ 0 & \text{otherwise} \end{cases}$$

$$f_q(p) = \text{number of distinct matches between p and q}$$

$$N = \text{number of query terms}$$

$$\text{score(p)} = \frac{f_q(p)^2}{N} + w_p * \frac{f_q(p)}{N}$$

# Snippet generation

## Esempi

**Query**: DNA

### Google

Deoxyribonucleic acid is a molecule composed of two polynucleotide chains that coil around each other to form a double helix carrying genetic instructions for ...

### Wikoogle

Deoxyribonucleic acid DNA does not usually exist as a single strand, but instead as a pair of strands that are held tightly together.

# Snippet generation

## Esempi

## **Query**: Anarchism etymology

## Google

Etymology, terminology and definition - The etymological origin of anarchism is from the Ancient Greek anarkhia, meaning "without a ruler", composed of the prefix an- (i.e. "without") and the word arkhos (i.e. "leader" or "ruler"). ... Anarchism appears in English from 1642 as anarchisme and anarchy from 1539.

## Wikoogle

Etymology, terminology and definition The etymological origin of anarchism is from the Ancient Greek anarkhia, meaning "without a ruler", composed of the prefix an- (i.e.Hence, it might be true to say that anarchism is a cluster of political philosophies opposing authority and hierarchical organization ...

# Query expansion

Query Expansion Using Local and Global Document Analysis. Jinxi Xu - W. Bruce Croft

## Local context analysis

L'idea è quella di selezionare i migliori n "concetti" dai primi m documenti considerati rilevanti. Ogni documento è rappresentato da uno snippet di 300-400 parole dal quale vengono presi i concetti

## Automatic query expansion

Espandere la query tramite WordNet. Richiede metodi di WSD per scegliere il senso dei termini correttamente

## Thesaurus expansion

# Local context analysis

## Concept

“ *A noun group (phrase) is either a single noun, two adjacent nouns or three adjacent nouns.*

Query Expansion Using Local and Global Document Analysis. Jinxi Xu - W. Bruce Croft ”

## Snippet

“ A passage is a text window of fixed size (300 words in these experiments [Callan, 1994]). [..] Since documents can be very long and about multiple topics, a co-occurrence of a concept at the beginning and a term at the end of a long document may mean nothing and It is also more efficient ”

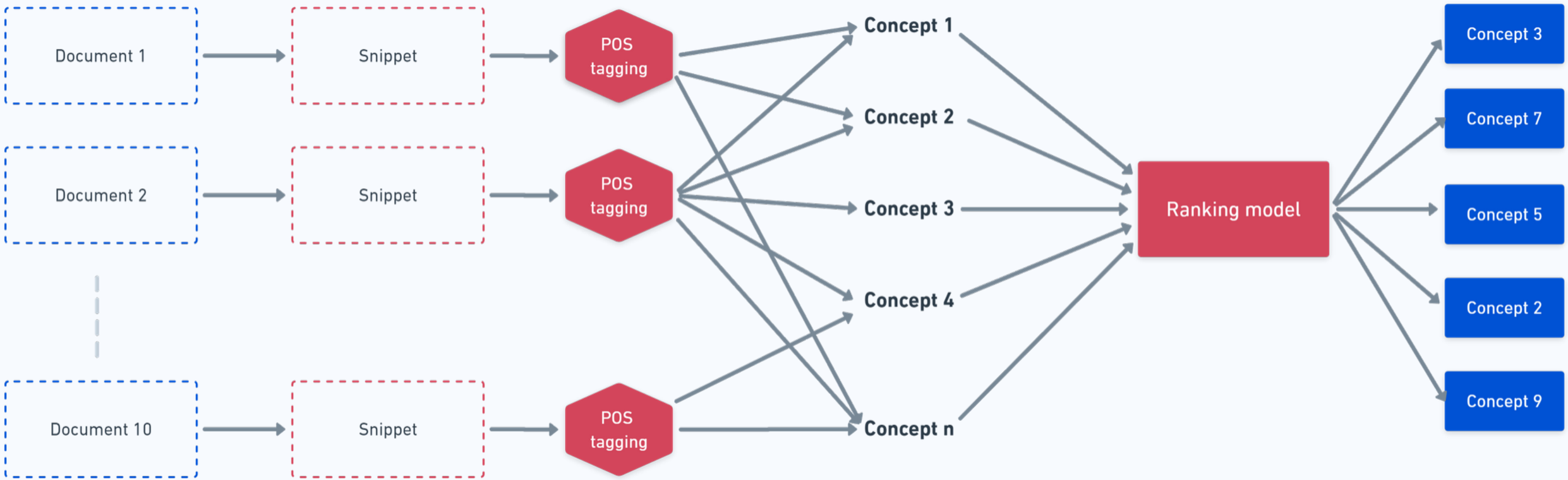
Query Expansion Using Local and Global Document Analysis. Jinxi Xu - W. Bruce Croft

SEARCHING

# Local context analysis

## L'algoritmo

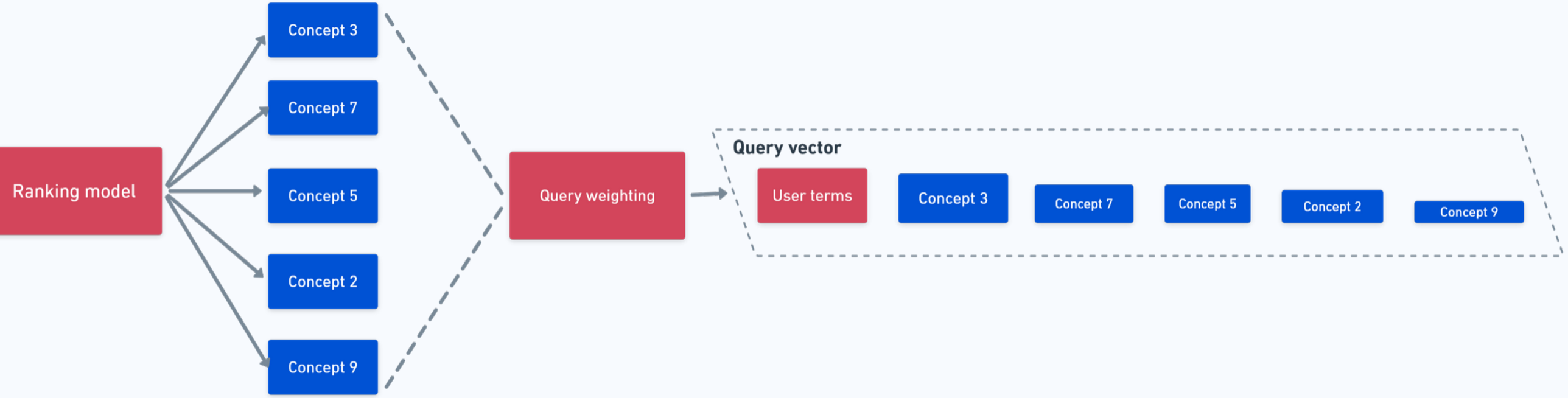
Viene fatta una prima query e vengono considerati i primi 10 documenti come rilevanti



# Local context analysis

## L'algoritmo

I concetti selezionati vengono classificati in base alla query e la query viene riformulata pesando i concetti opportunamente



# Ranking model

$$\text{score}(Q,\ c) \ = \prod_{t_i \in Q} (\delta \ + \ log(af(c, t_i))\ idf_c \ / \ log(n))^{idf_i}$$

where

$$af(c,\ t_i) \ = \sum ft_{ij} fc_j$$
$$idf_i \ = \max(1.0,\ log_{10}(N / N_i) / 5.0)$$
$$idf_c \ = \max(1.0,\ log_{10}(N / N_c) / 5.0)$$

$c$ is a concept
$ft_{ij}$ is the number of occurrences of $t_i$ in $p_j$
$fc_j$ is the number of occurrences of c in $p_j$
$N$ is the number of snippets in the collection
$N_i$ is the number of snippets containing $t_i$
$N_c$ is the number of snippets containing c
$\delta$ is 0.1 to avoid 0

# Local context analysis

$$weight(c) \ = \ 1.0 - 0.9\ i_c / m$$

where

m = number of concept selected
i = position the ranking of c

I migliori concetti vengono pesati per evitare di stressare i termini dell'utente

Query weighting

# Local context analysis

## Esempi

## Query: DNA

dna replic
group
singl strand dna
doubl strand dna
singl strand rna
singl strand
molecular biologi
doubl helix
polymerase chain
gel electrophoresi
dna viru

agaros gel
dna depend dna
dna ligas
origin dna
complementari dna
specif dna
genet inform
baltimor classif
genet materi
dna essenti
deoxyribonucleic acid

Esempi

# Query: Microsoft

microsoft offic
offic suit
microsoft access
user interfac
first version
internet explor
microsoft word
word processor
microsoft corpor
microsoft excel

## Local context analysis

graphic user
databas manag
version histori
softwar graphic
visual basic
relat databas
softwar develop tools.it
programm languag
databas engin
american multin
microsoft edg

# Considerazioni

La presuzione di prendere come rilevanti i primi 10 documenti può essere azzardata per alcune query: query drift.
Concetti con score al di sotto un certo threshold sono scartati

La selezione dei concetti dipende dal processo di POS tag, snippet generation e da come il testo è stato parsato dal parser di Wikipedia.

## Local context analysis

## POS tag

```
grammar = r"""
        NBAR: {<NN|JJ><|JJ|NN>} # Nouns and Adjectives, terminated with Nouns
         # If pattern not found just a single NN is ok
    """
```

# Thesaurus based

## Global analysis

Espandere i termini della query ricercando sensi da un dizionario: Wordnet

## WSD

É necessario un metodo di word sense disambiguation per selezionare il senso corretto di termini polisemici: Lesk

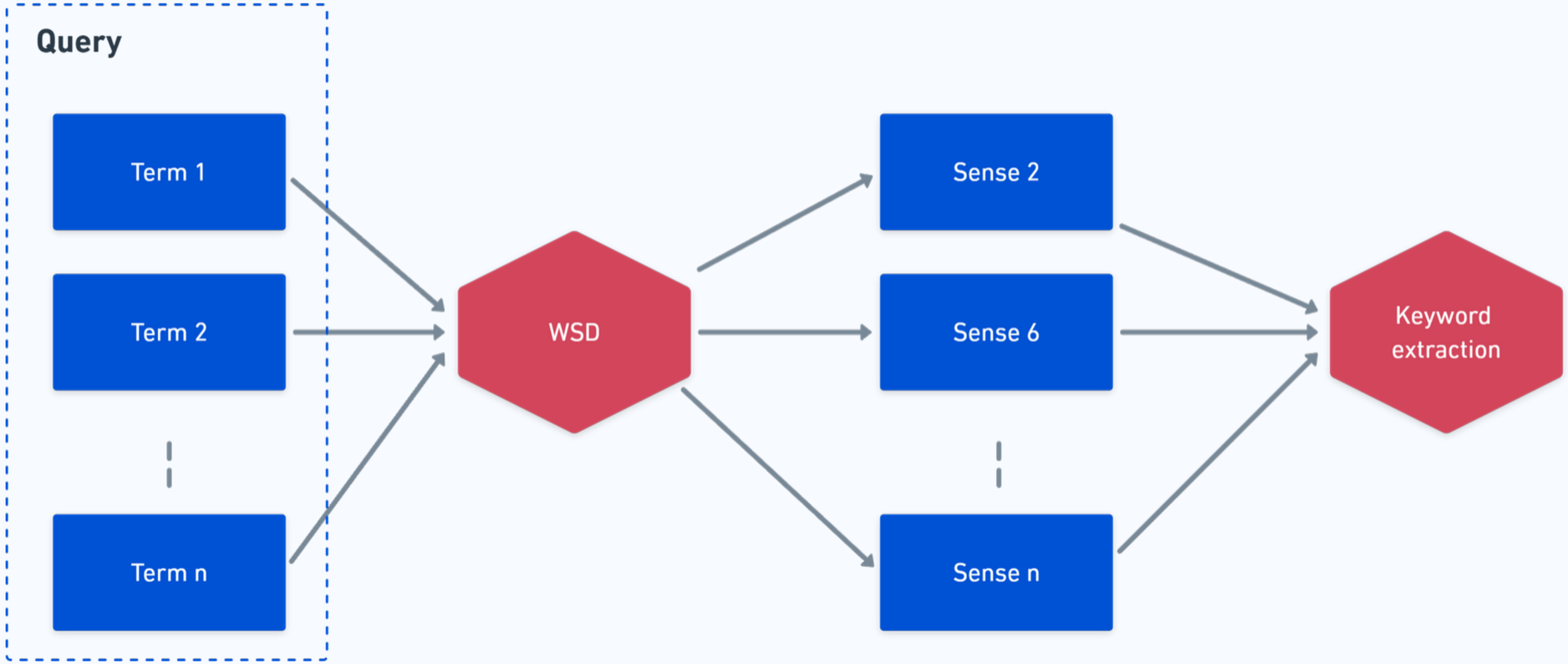
## Adaptive LESK

Adapted/Extended Lesk (Banerjee and Pederson, 2002/2003)

L'idea è vedere termini che co-occorrono entro una certa finestra di termini presi dalla definizione o dall'albero relazionale

# Thesaurus based

## L'algoritmo




Una volta selezionati i sensi vengono estratti i termini dal gloss e lemmi definiti su wordnet per ogni senso

# Thesaurus based

Esempi

## Query: dna

transmiss
inform
helix
polymer
long
cell

nucleotid
doubl
biochemistri
linear
nucleu
genet
deoxyribonucleic
acid

# Thesaurus based

Esempi

## Query: apple

malu

tree

nativ

mani

edibl

orchard

appl

firm

varieti

fruit

pumila

I termini non potrebbero essere affini a quello che vuole l'utente, sopratutto con una query molto ambigua come Apple

# Thesaurus based

## Considerazioni

Anche il metodo thesaurus based ha problemi di query drift. Ai termini espansi viene assegnato un peso per non stressare i termini dell'utente.

Termini specifici (es.DNA) vengono espansi bene invece termini polisemici con un contesto troppo generico potrebbero degradare le performance

Potrebbe risultare più efficente utilizzare un corpus specifico basato su wikipedia costruito a index time anzichè wordnet sia per query expansion che WSD

Textual

# Ranking

## Okapi BM25

Ranking function used by search engines to estimate the relevance of documents to a given search query. It is based on the probabilistic retrieval framework

## Terrier PL2

# Link analysis

# Ranking

Obiettivo implementare un modello di scoring che combinasse i valori del ranking testuale a quelli ottenuti tramite link analysis

## Pagerank

I risultati ottenuti dal modello di ranking testuale vengono **combinati** al valore di pagerank ottenuto a index time

## Hits

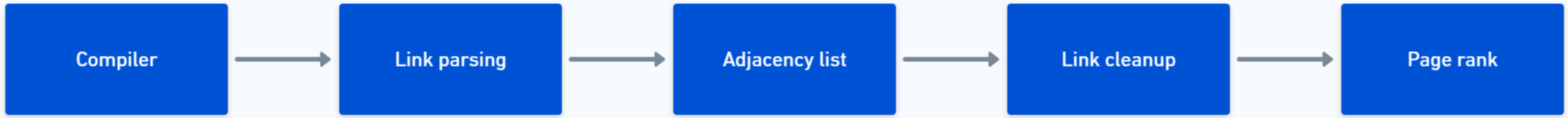
I risultati ottenuti dal modelli di ranking testuale vengono combinati ai valori di hub e authority calcolati sulla base del sottografo dei risultati più rilevanti.

RANKING

# Link pipelines

**Link analysis**

Dopo aver analizzato tutti gli articoli viene generata una lista delle adiacenze con i link tra le varie pagine di wikipedia



RANKING

# Link Analysis

## Link pipelines

Vengono scritti su file i collegamenti tra le varie pagine man mano che gli articoli vengono analizzati

### Adjacency list

La lista delle adiacenze viene salvata su file per motivi di robustezza e ottimizzazione della memoria con un risparmio notevole.

~ 173 Mb

### Link cleanup

Per ottenere valori più affidabili e un grafo più leggero è necessario ripulire la lista delle adiacenze ai soli articoli presenti nel dump

~ 43 Mb

# Link Analysis

## Link pipelines

Viene trasformata la lista delle adiacenze in formato graphml per poter operare in modo efficiente sul grafo

### NetworkX

*NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.*

### python-igraph

*igraph is a collection of network analysis tools with the emphasis on efficiency, portability and ease of use*

# Link Analysis

## Modello di ranking

Come andare a integrare uno score query-independent con uno score query-dependent per ottenere risultati sensati?

## Okapi BM25 + Pagerank

Estensione del modello di ranking BM25 mediante combinazione lineare al valore di pagerank relativo all'articolo

$$R(p, Q) = \alpha BM25(p, Q) + (1 - \alpha) PageRank(p)$$

Modern information retrieval - R.Baeza Yates, B.Riberio Neto

**Link Analysis**

# Okapi BM25 + Pagerank

## Problema

I valori di pagerank non sono uniformati allo score di BM25. Nemmeno normalizzando i valori è possibile raggiungere un risultato ottimale dato che sono molto distanti tra loro.

Vengono privilegiati gli articoli con un valore di pagerank altro e di conseguenza vengono visualizzati risultati non rilevanti.

# Pagerank sorting

Viene ordinata una finestra M = 30 di risultati rilevanti determinati attraverso il modello di ranking testuale predefinito sulla base del relativo score e il valore di pagerank. In alcune query è stato possibile ottenere una precisione migliore mentre in altri casi i valori sono peggiorati.

Si è concluso che potrebbe avere senso un modello di ranking che prenda in considerazione una serie di parametri come il valore di pagerank, la similarità del documento e del topic con la query (document clustering).

> *The authors used a Bayesian network to combine the different signals and showed that the combination leads to far better results than those produced by any of the combining ranking function in isolation*
>
> Modern information retrieval - R.Baeza Yates, B.Riberio Neto

# Link Analysis

## Esempio pagerank

list_of_portuguese_monarchs 1.1719406346447165e-05

portugal 0.00033106614157568534

kings_of_naples 2.8733564451697862e-06

list_of_monarchs_of_brazil 3.720489975538177e-06

manuel_i_of_portugal 2.6250022072998636e-05

john_iv_of_portugal 8.084313578111862e-06

list_of_counties_in_vermont 5.654867206179848 5e-06

county_(united_states) 0.0004142097455776016

# Link Analysis

## Modello di ranking

Come andare a calcolare e integrare lo score hits query-dependent allo score del modello di ranking per ottenere risultati sensati?

## Hits sorting

Vengono ordinati i risultati rilevanti determinati attraverso il modello di ranking predefinito sulla base del relativo score e i valori di hub, authority.

$$h,\ a\ =\ Hits(p,\ S)$$
$$R(p,\ Q)\ =\ BM25(p,\ Q) * (max(h,\ a))$$

# Link Analysis

## Esempio hits

| | | | | |
|---|---|---|---|---|
| ↑ | steve_wozniak | (0.61) | 7.90 | 4.83 |
| | apple_i | (0.33) | 7.87 | 2.62 |
| ↑ | next | (0.77) | 7.66 | 5.94 |
| | next_computer | (0.29) | 7.57 | 2.24 |
| | john_sculley | (0.52) | 7.54 | 3.93 |
| ↓ | fruitarianism | (0.06) | 7.49 | 0.47 |
| ↑ | apple_ii | (1.0) | 7.14 | 7.14 |
| ↓ | danny_boyle | (0.07) | 7.11 | 0.54 |
| ↓ | kate_winslet | (0.01) | 7.10 | 0.10 |
| | jef_raskin | (0.33) | 7.07 | 2.36 |
| | apple_lisa | (0.46) | 7.04 | 3.29 |
| ↑ | apple_inc. | (1.0) | 6.85 | 6.85 |
| ↓ | pixar | (0.17) | 6.77 | 1.21 |
| × | bloom_county | (0) | 6.40 | 0.0 |
| × | steve_martin | (0) | 6.40 | 0.0 |

# User interface

1. Schermata home
2. Risultati ricerca
3. Impostazioni

L'interfaccia proposta è molto semplice, accessibile dall'utente finale attraverso un browser e molto simile a quella dei moderni search engine come Google

Settings

# Wikoogle

Search

https://en.wikipedia.org/wiki/DNA

## 1. DNA

Deoxyribonucleic acid (; **DNA** does not usually exist as a single strand, but instead as a pair of strands that are held tightly together.

https://en.wikipedia.org/wiki/Genetics

## 2. Genetics

Diploid organisms form haploids by dividing, without replicating their **DNA**, to create daughter cells that randomly inherit one of each pair of chromosomes.Some bacteria can undergo Bacterial conjugation, transferring a small circular piece of **DNA** to another bacterium.

https://en.wikipedia.org/wiki/Cell_(biology)

## 3. Cell (biology)

The **DNA** of a prokaryotic cell consists of a single Circular prokaryote chromosome that is in direct contact with the cytoplasm.an organelle that houses the cell & #39; s **DNA**.

https://en.wikipedia.org/wiki/Biochemistry

## 4. Biochemistry

The **DNA** polymerase of the thermophile bacteria Thermus aquaticus, extracted in 1968 and named Taq polymerase, is a biochemical **DNA** replicator resistant to relative high temperatures (50-80 °C), which has allowed molecular biologists to ease complications in the Polymerase chain reaction method.

https://en.wikipedia.
1. DNA
Deoxyribonucleic a
strands that are he

https://en.wikipedia.
2. Genetics
Diploid organisms
that randomly inhe
conjugation, transf

https://en.wikipedia.
3. Cell (biolog
The **DNA** of a proka
contact with the cy

https://en.wikipedia.
4. Biochemist
The **DNA** polymera
Taq polymerase, is
which has allowed
method.

https://en.wikipedia.
5. Acid
Nucleic acids are i

# Settings

## Results limit

10 20 30

## Query expansion

### Local Context Analysis query expansion

Query expansion based on local context analysis algorithm. The main idea is to expand a query by selecting key "concepts" from the first 10 documents. These concepts are ranked accordingly with the query and the top ones are added in the query vector space with a specific weight. Based on the following paper by Jinxi Xu and W. Bruce Croft

### Thesaurus based query expansion

Query expansion based on a thesaurus(Wordnet). The query is expanded with the gloss of each disambiguated synset. Wordnet is generic corpus that might not be suited to expand queries for wikipedia(a wikipedia corpus could be better) so query drift might be behind the corner with polysemic and very general queries.

### No query expansion

Disable query expansion

INFRASTRUCTURE

# Architettura

## Flask Web Server

*Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.*

## + Jinja Templates

*Jinja is a modern and designer-friendly templating language for Python, modelled after Django’s templates. It is fast, widely used and secure with the optional sandboxed template execution.*

# Live server

L'applicazione è stata eseguita su una macchina cloud VPS configurata ad-hoc per superare limitazioni di memoria e potenza di calcolo.

In questo modo è stato possibile indicizzare più parti del wikidump

| S.O. | CPU | RAM | SSD |
|---|---|---|---|
| Debian | quad core | 12 GB | 60GB |

# Live server

## Sessione utente

Ogni utente detiene una propria sessione sulla quale vengono memorizzate le impostazioni di ricerca

## Oggetti condivisi

Per motivi di performance e utilizzo della memoria vengono mantenuti oggetti condivisi quali IndexReader e Searcher

**~ 4 Gb** di RAM utilizzata con il dump di riferimento

# Ottimizzazioni

## Preprocessing

### out-of-memory exceptions nella fase di indicizzazione

Durante la fase di indicizzazione si sono verificati problemi causati da un utilizzo eccessivo della memoria disponibile. In particolare questo era dovuto al fatto che Whoosh mantiene in memoria i documenti da indicizzare fino a quando l'indice non viene scritto.

Ottimizzazioni

# Preprocessing

## out-of-memory exceptions nella fase di indicizzazione

Dopo diversi tentativi è stato possibile determinare una configurazione ottimale per garantire velocità nella fase di indicizzazione e un corretto utilizzo della memoria.

L'indice viene costruito mediante scritture multiple ogni 20k articoli processati che consentono di liberare la memoria centrale con un utilizzo minimo di 500 Mb fino a un picco massimo di 4 Gb.

Ottimizzazioni

# Preprocessing

## bottleneck nella fase di compilazione degli articoli

Si è notata la presenza di un collo di bottiglia nella fase di compilazione degli articoli che rallenta notevolmente il processo di indicizzazione. Questa fase viene eseguita in modo sequenziale e pertanto non consente di sfruttare al meglio architetture a multiprocessore.

Per via delle caratteristiche architetturali del progetto non è stato possibile applicare migliorie che consentano di ovviare al problema in tempi rapidi.

Ottimizzazioni

# Link Analysis

## fault-tolerance nella fase di link analysis

Per garantire una maggiore tolleranza ai rischi nella fase di link analysis si è deciso di scrivere su file la lista delle adiacenze man mano che gli articoli vengono processati. Questa modifica ha inoltre consentito un notevole risparmio in termini di utilizzo della memoria.

Senza questa miglioria in caso di errore nella fase di indicizzazione sarebbe stato necessario ri-processare tutti gli articoli per determinare nuovamente la matrice delle adiacenze.

# Ottimizzazioni

## Link Analysis

### poor-performance nella fase di link analysis cleanup

La fase di pulizia della lista delle adiacenze si è rivelata molto onerosa in termini di tempo e computazione, per questo motivo si è deciso di parallelizzare tale parte di codice.

Per ridurre ulteriormente il tempo computazionale dedicato a questa fase si è pensato di effettuare la ricerca degli articoli mediante tabella di hash piuttosto che all'interno di un array con una conseguente riduzione del costo da O(n) a O(1).

Ottimizzazioni

# Architettura

## microservizi e job code

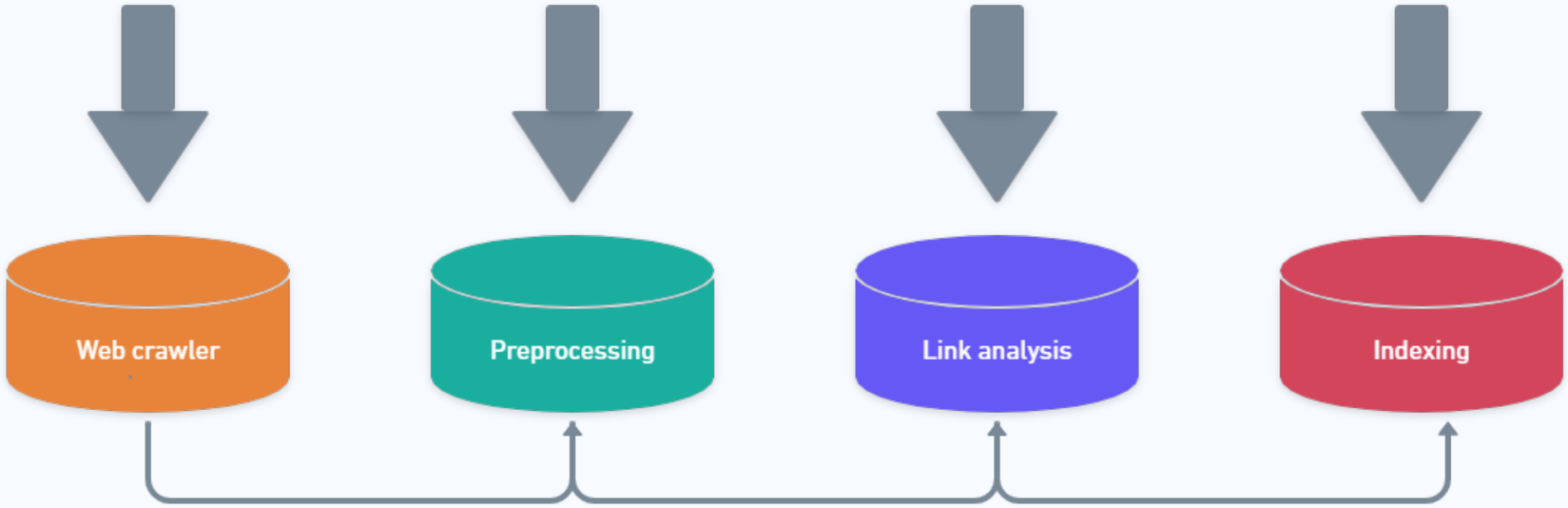
Sulla base delle criticità evidenziate nelle slide precedenti si è pensato ad una architettura orientata a microservizi con i job distribuiti su diverse code.

Questo tipo di architettura avrebbe consentito di ridurre notevolmente la complessità del sistema con conseguenti vantaggi di scalabilità e manutenibilità.

# Architettura

## microservizi e job code

**Ottimizzazioni**




I componenti del sistema sono tra loro indipendenti e non dipendono da implementazione specifiche

# Test suite

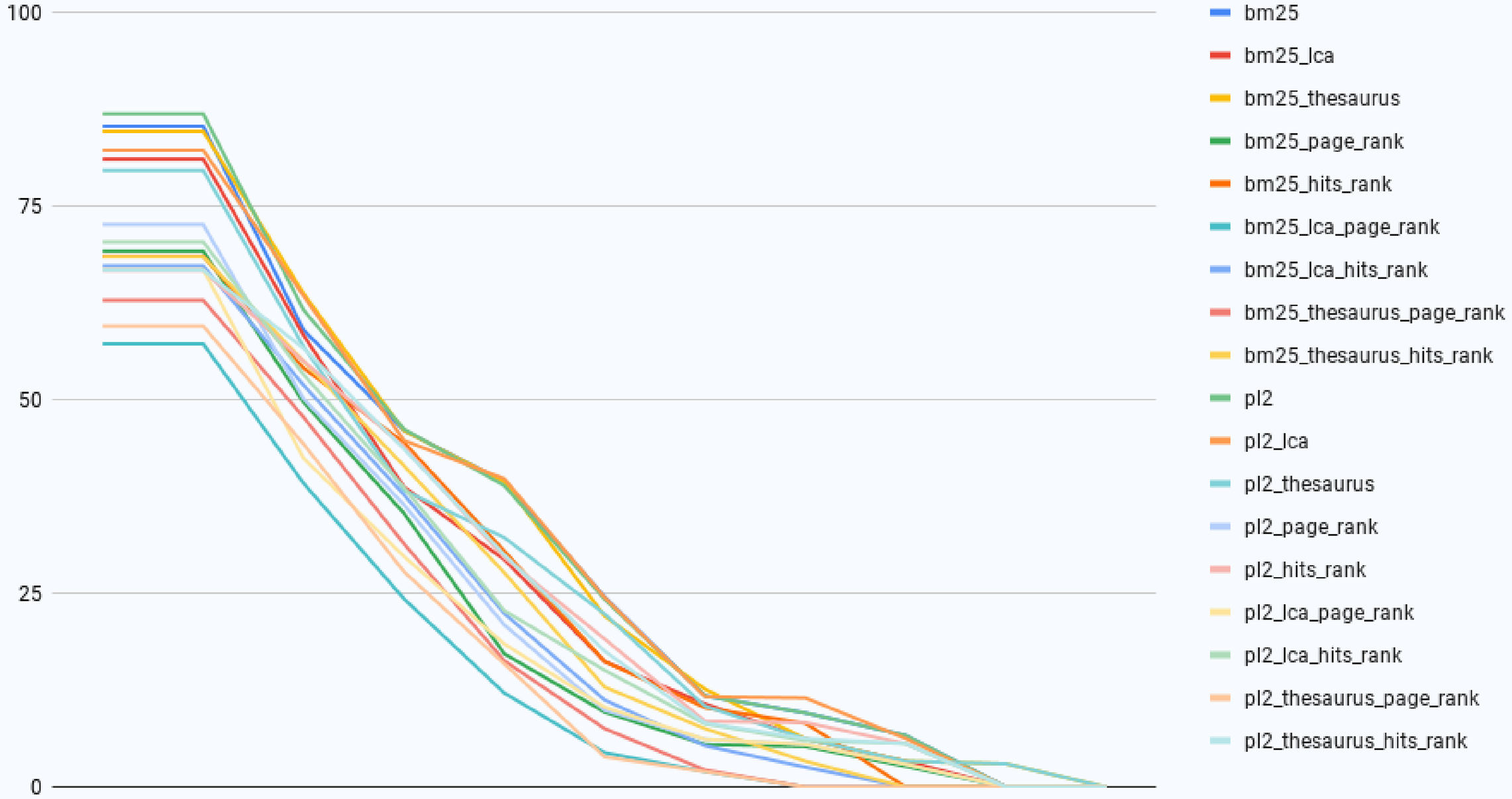
**Scraper**

La creazione del test suite è stata effettuata tramite uno scraper che effettua query su google selezionando solamente gli articoli presenti nel dump.

Una volta trovati 30 articoli per ogni query viene calcolata la MAP

Google è molto dinamico (i risultati dipendono dall'utente, history, posizione, ecc..), quindi la costruzione del test set è stata fatta attraverso un ambiente isolato per evitare che i risultati venissero influenzati da parametri estern

# EVALUATION


100
75
50
25
0
bm25
bm25_lca
bm25_thesaurus
bm25_page_rank
bm25_hits_rank
bm25_lca_page_rank
bm25_lca_hits_rank
bm25_thesaurus_page_rank
bm25_thesaurus_hits_rank
pl2
pl2_lca
pl2_thesaurus
pl2_page_rank
pl2_hits_rank
pl2_lca_page_rank
pl2_lca_hits_rank
pl2_thesaurus_page_rank
pl2_thesaurus_hits_rank

EVALUATION

# MAP

| | |
|---|---|
| bm25 | 33,41515152 |
| bm25_lca | 29,54848485 |
| bm25_thesaurus | 33,23939394 |
| bm25_page_rank | 23,96363636 |
| bm25_hits_rank | 27,33333333 |
| bm25_lca_page_rank | 17,86060606 |
| bm25_lca_hits_rank | 24,15151515 |
| bm25_thesaurus_page_rank | 20,98484848 |
| bm25_thesaurus_hits_rank | 25,89090909 |
| pl2 | 33,88181818 |
| pl2_lca | 33,29393939 |
| pl2_thesaurus | 30,16969697 |
| pl2_page_rank | 25,1969697 |
| pl2_hits_rank | 27,58181818 |
| pl2_lca_page_rank | 22,65151515 |
| pl2_lca_hits_rank | 26,38484848 |
| pl2_thesaurus_page_rank | 19,35151515 |
| pl2_thesaurus_hits_rank | 27,39393939 |

# Considerazioni

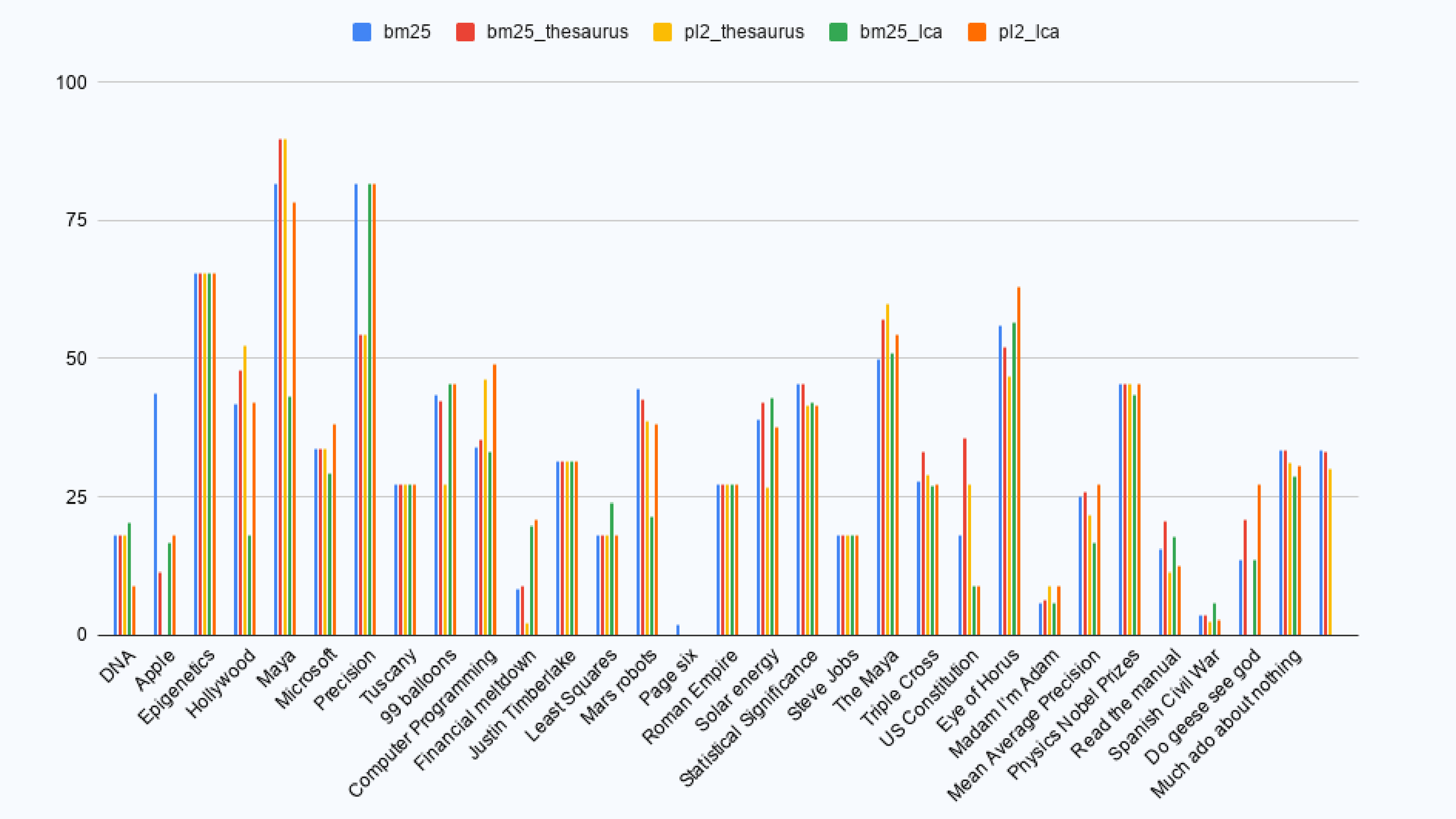
Dopo una analisi dei risultati e sulla base di come i valori di precision e recall cambiano  all'interno dei vari sistemi sono state tratte le seguenti considerazioni

## Query

Query con più parole hanno scarse performance perchè non viene tenuto in considerazione la proximity tra i termini. L'effetto si amplifica ancora di più con query expansion LCA o in generale con query expansion

## Expansion

Non ha senso espandere tutte le query. Query specifiche sono espanse bene e migliorano le performance. In altri casi degradano notevolmente, in particolare su query generiche.


bm25
bm25_thesaurus
pl2_thesaurus
bm25_lca
pl2_lca
100
75
50
25
0
DNA
Apple
Epigenetics
Hollywood
Maya
Microsoft
Precision
Tuscany
99 balloons
Computer Programming
Financial meltdown
Justin Timberlake
Least Squares
Mars robots
Page six
Roman Empire
Solar energy
Statistical Significance
Steve Jobs
The Maya
Triple Cross
US Constitution
Eye of Horus
Madam I'm Adam
Mean Average Precision
Physics Nobel Prizes
Read the manual
Spanish Civil War
Do geese see god
Much ado about nothing

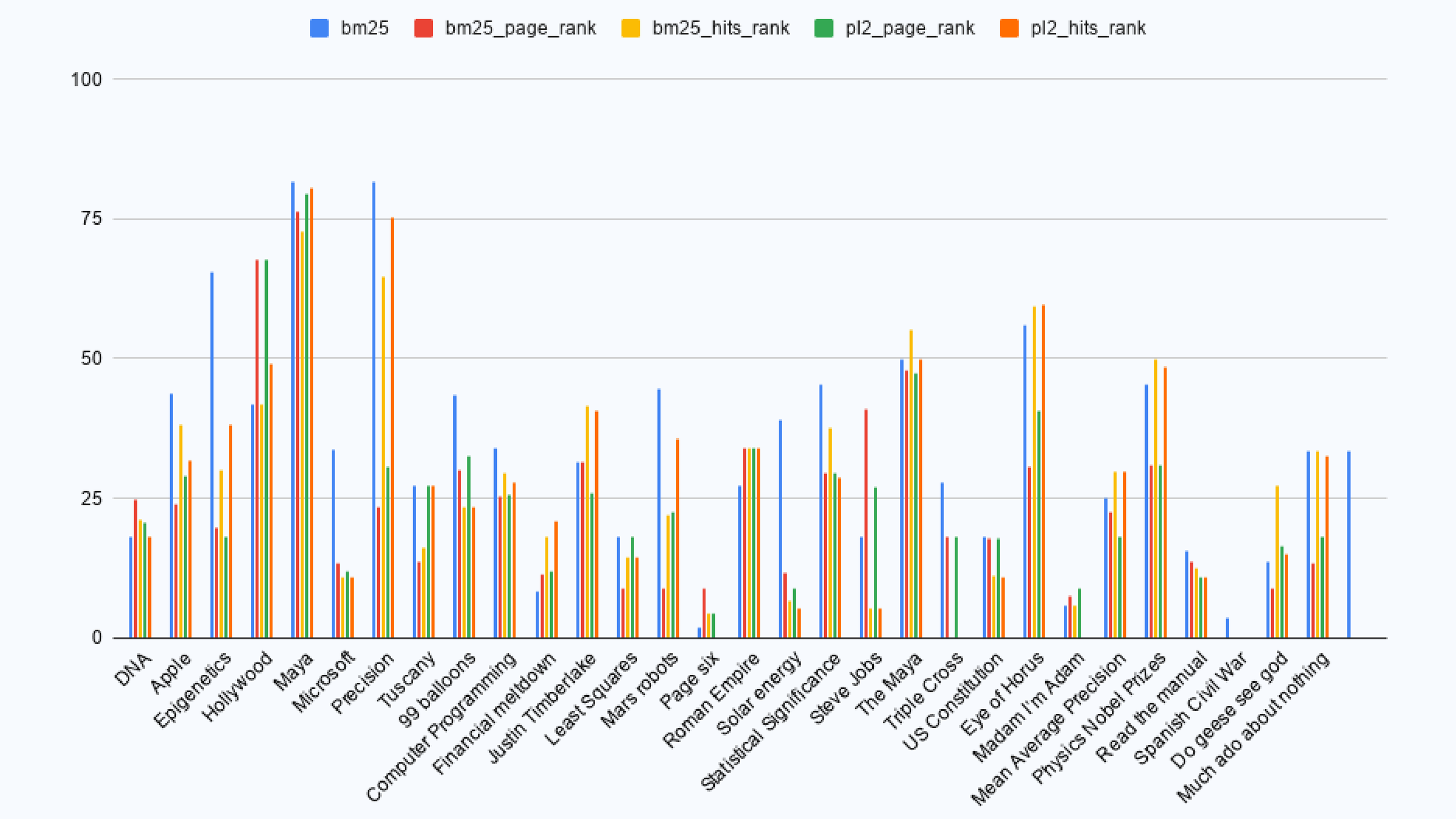
# Considerazioni

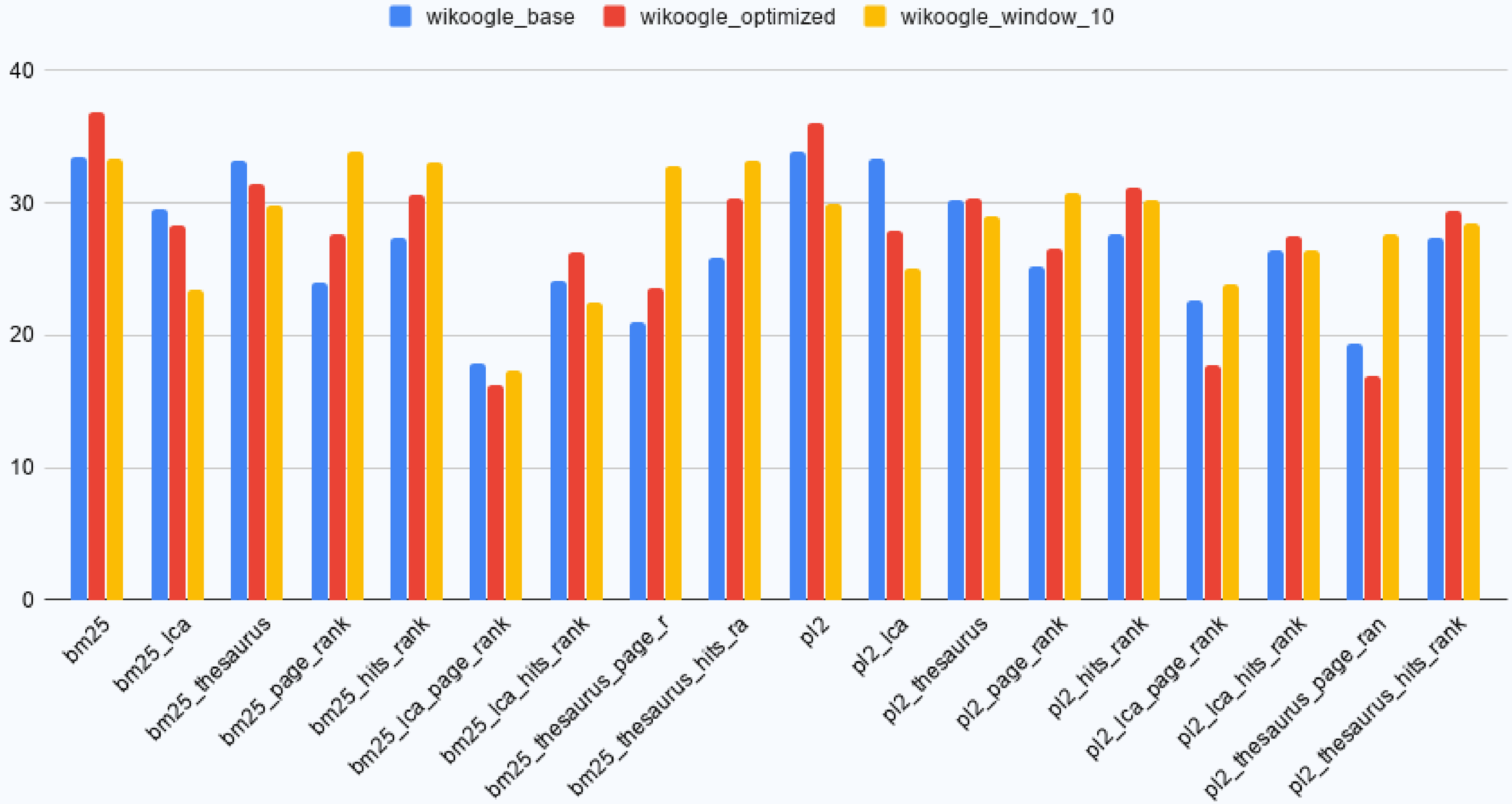
## Link analysis

Ordinare i risultati sulla base di una data finestra funziona bene per certe query ma peggiore in altri casi. Limitando la finestra a 10 risultati si ottiene un ranking più affidabile e affine al sistema di base. Si è notato come spesso alcune pagine, seppur non rilevanti, vengono messe in risalto e altre considerate rilevanti perdono la loro posizione.

## Pagerank e Hits

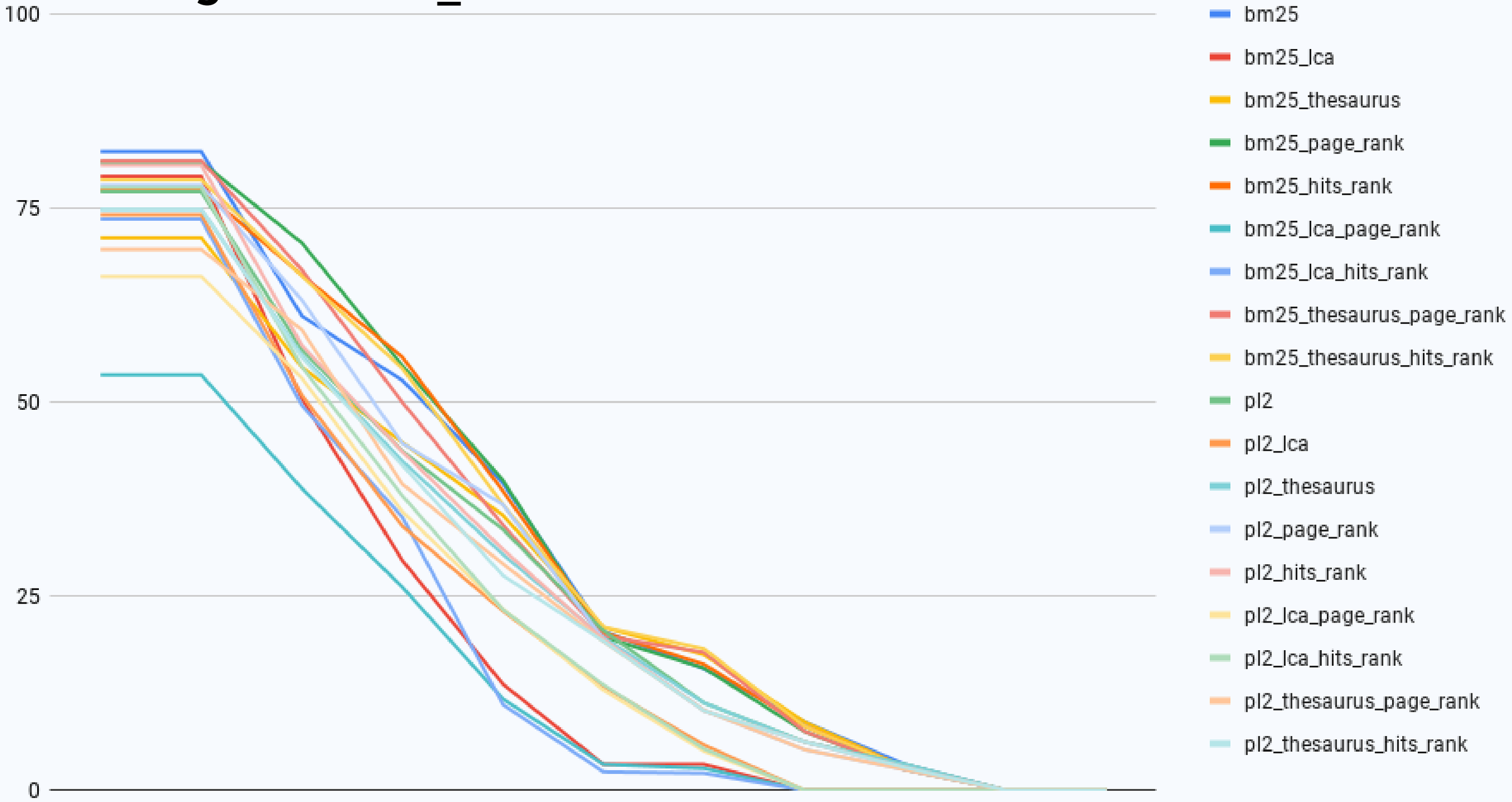
Utilizzare un modello di score dove il risultato finale è ottenuto attraverso una somma pesata di diverse ***features*** potrebbe portare a risultati migliori rispetto ad un semplice ordinamento. Google ad esempio utilizza più di 200 features per definire il ranking di una pagina.


bm25
bm25_page_rank
bm25_hits_rank
pl2_page_rank
pl2_hits_rank
100
75
50
25
0
DNA
Apple
Epigenetics
Hollywood
Maya
Microsoft
Precision
Tuscany
99 balloons
Computer Programming
Financial meltdown
Justin Timberlake
Least Squares
Mars robots
Page six
Roman Empire
Solar energy
Statistical Significance
Steve Jobs
The Maya
Triple Cross
US Constitution
Eye of Horus
Madam I'm Adam
Mean Average Precision
Physics Nobel Prizes
Read the manual
Spanish Civil War
Do geese see god
Much ado about nothing

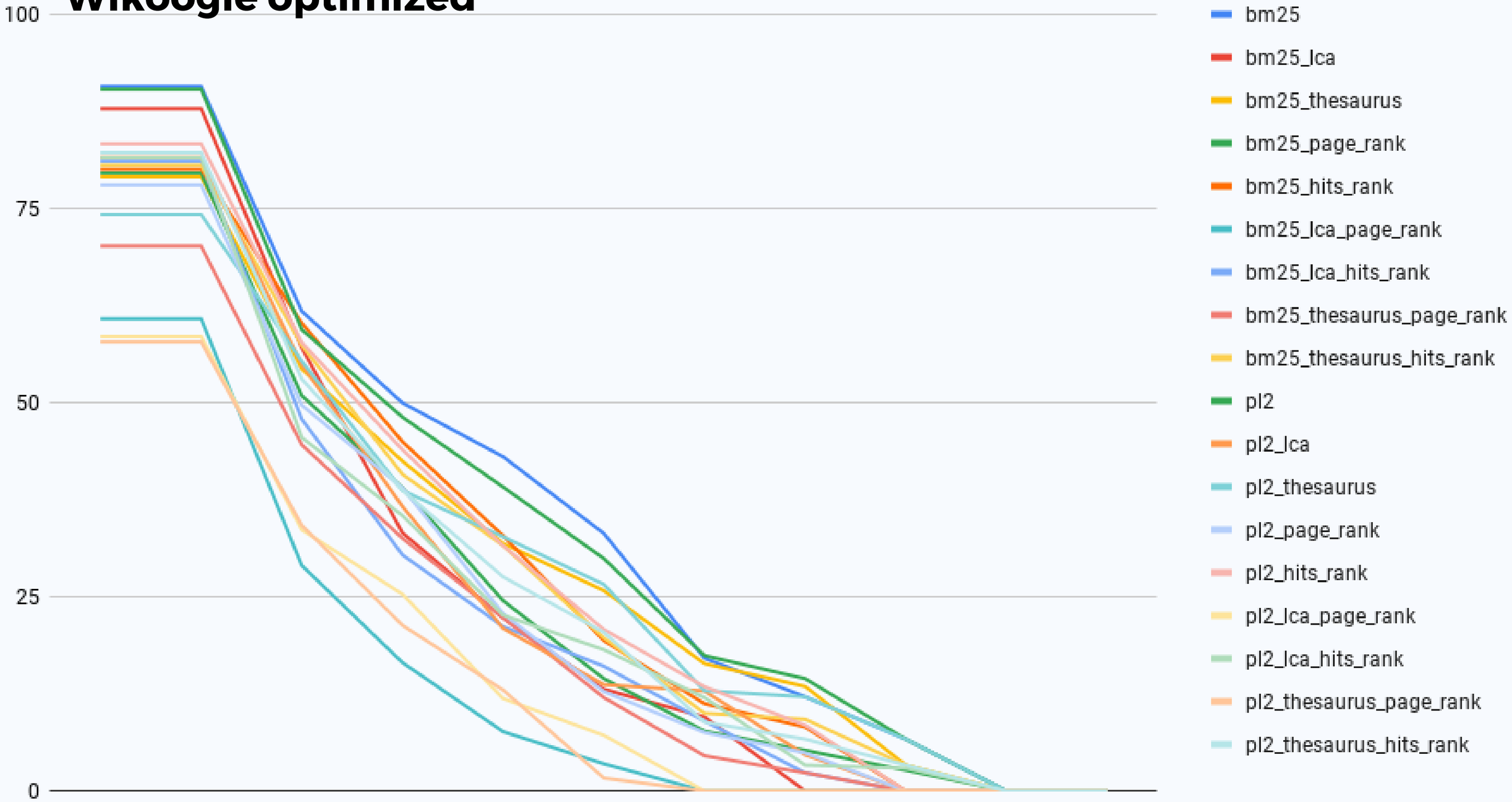

wikoogle_base
wikoogle_optimized
wikoogle_window_10
40
30
20
10
0
bm25
bm25_lca
bm25_thesaurus
bm25_page_rank
bm25_hits_rank
bm25_lca_page_rank
bm25_lca_hits_rank
bm25_thesaurus_page_r
bm25_thesaurus_hits_ra
pl2
pl2_lca
pl2_thesaurus
pl2_page_rank
pl2_hits_rank
pl2_lca_page_rank
pl2_lca_hits_rank
pl2_thesaurus_page_ran
pl2_thesaurus_hits_rank

| | wikoogle_base | wikoogle_optimized | wikoogle_window_10 |
|---|---|---|---|
| bm25 | 33,41515152 | 36,84545455 | 33,29393939 |
| bm25_lca | 29,54848485 | 28,26666667 | 23,5 |
| bm25_thesaurus | 33,23939394 | 31,44242424 | 29,75151515 |
| bm25_page_rank | 23,96363636 | 27,6030303 | 33,88181818 |
| bm25_hits_rank | 27,33333333 | 30,63939394 | 33,00909091 |
| bm25_lca_page_rank | 17,86060606 | 16,20909091 | 17,26969697 |
| bm25_lca_hits_rank | 24,15151515 | 26,26969697 | 22,5 |
| bm25_thesaurus_page_rank | 20,98484848 | 23,51515152 | 32,83333333 |
| bm25_thesaurus_hits_rank | 25,89090909 | 30,28787879 | 33,18787879 |
| pl2 | 33,88181818 | 35,99090909 | 29,96666667 |
| pl2_lca | 33,29393939 | 27,9 | 25,03939394 |
| pl2_thesaurus | 30,16969697 | 30,32121212 | 28,98787879 |
| pl2_page_rank | 25,1969697 | 26,62121212 | 30,73636364 |
| pl2_hits_rank | 27,58181818 | 31,14545455 | 30,16969697 |
| pl2_lca_page_rank | 22,65151515 | 17,73636364 | 23,87575758 |
| pl2_lca_hits_rank | 26,38484848 | 27,56666667 | 26,36060606 |
| pl2_thesaurus_page_rank | 19,35151515 | 16,8969697 | 27,69090909 |
| pl2_thesaurus_hits_rank | 27,39393939 | 29,36666667 | 28,51212121 |

# Wikoogle window_10



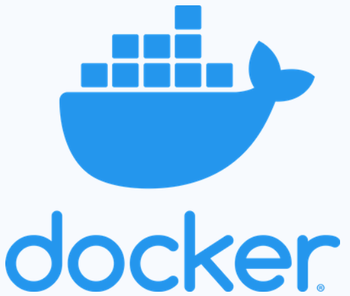
# Wikoogle optimized







# Tech stack

DEMO AT **http://212.237.42.43:8080/**

# Wikoogle

PREPROCESSING

# WIKIPEDIA PARSING

Fonti differenti hanno grammatiche incomplete, ambigue o inconsistenti tra di loro

Per questo motivo è stata definita una grammatica interna che fosse la meno ambigua, più completa e semplice possibile ai fini di indicizzazione e analisi

```
template = "{{", title, { "|", part }, "}}" ;
part     = [ name, "=" ], value ;
title    = text ;
```

```
text           :=  &
template       := '{{' text '}}'
```

PREPROCESSING

# WIKIPEDIA PARSING

> *So far progress has been made in both grammar definition and parser behaviour. Although none of the descriptions seems to be complete, some have achieved to describe a good part of the language.Current parser descriptions tried to do its best to follow the MediaWiki's parser behaviour.*

mediawiki.org/wiki/Markup_spec

EBNF

```
start link    = "[[";
end link      = "]]";
internal link = start link, full
pagename, ["|", label], end link, label
extension;
```

ABNF

```
link = "[[" wikitext-L3 "]]"
wikitext-L3     = literal / template /
tplarg / link / comment /
                  line-eating-comment /
unclosed-comment / xmlish-element /
                  *wikitext-L3
```