HackerSearch

Information Retrieval

Ana Barros - up201806593
João Costa - up201806560
João Martins - up201806436

Frontend

- **Search bar** – queries the default search field (searches in all fields);
- **Advanced search** – allows the user to query different fields at the same time;
- The **number of results** for the query are shown below along with the results (uses **pagination**);
- Matching terms in the user query present in the documents are **highlighted**;
- Results can be **filtered by type**: **faceting**;
- **Sorting** functionality **based on date and score**;
- Progressive **word-completion and possible matching entry suggestion**.
- **Spellchecking** suggestions after querying (if any are found).

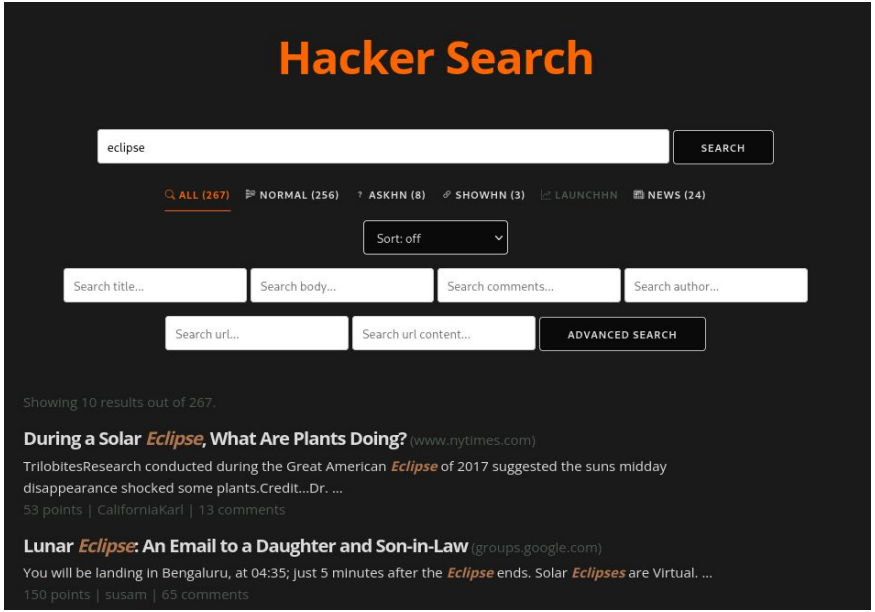

Figure 1: HackerSearch Frontend

Spellchecking (/spell)

- **New field-type:** *spellField*
  - Stream treatment with this field-type is the same during index and query time.
  - Preprocessing
    - Non-alphanumeric characters replaced with spaces.
  - Tokenizer
    - Splits on whitespace and punctuation.
  - Filters:
    - Non ASCII characters are converted to their ASCII equivalents.
    - Characters are converted to lower case.
    - English singular possessive cases disappear.
- *IndexBasedSpellChecker* was chosen over *DirectSolrSpellChecker*.
- Spellchecking is activated by default.
- *Collation* mechanism used in order to make it easier to work with the suggestions.

Query	Spellchecking
eclips	eclipse
luanr	lunar
vrsion cntrol	version control

Figure 2: Spellchecking result examples

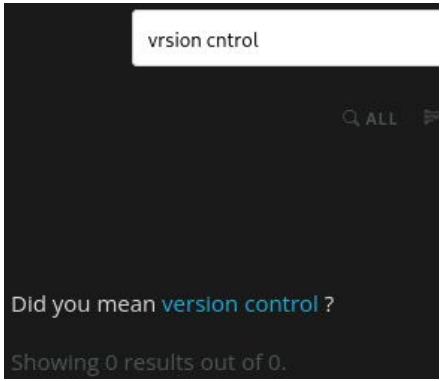

Figure 3: Collation feature example in frontend

Suggestions (/suggest) – Possible matching entry

Suggestions of possible hits for the current query:

- Based on the title of the documents.
- Suggests matches based on prefix matches to all tokens (*AnalyzingInfixLookupFactory)*.
- Stores index in a *DocumentDictionaryFactory* – light-weight and provides all features required.

- **New field-type**: *suggestion_type*.
  - Tokens split on whitespace and punctuation.
  - Characters are converted to lower case.
  - Non-alphanumeric characters are deleted.
- **New field**: *sugg*.
  - Sourced from the *story_title* field.
  - Uses the new field-type: *suggestion_type*.

Figure 4: Frontend matching entry suggestion

Query	Top suggestions
regain	…Autopilot automatically regained…
fin	Ask HN: How do you find roles as a solo developer?

Figure 5: Document suggestions examples

Suggestions (/suggest) – Single-word completion

Suggestions for current word completion:

- Uses the prefix of the word currently being typed;
- Is **context dependent** – looks at previous and **last 2 tokens** using **ngrams** (*FreeTextLookupFactory*);
- Stores index in a *DocumentDictionaryFactory* – light-weight and provides all features required;
- Uses the same field described in the previous slide;

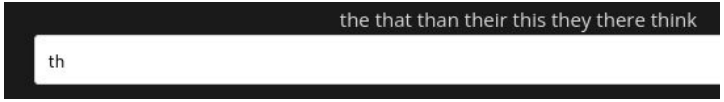

Figure 6: Word completion suggestions examples in frontend

Query	Suggestions
i miss y	you your years year yc
r	rust real run remote report
automatically r	automatically-regained rust real run remote

Figure 7: Word completion suggestions examples

Faceting

- Supported by Solr through the use of ***facet.query*** and ***facet.field***.
- Two features in the system:
  - Arrange results into buckets corresponding to different story types.
  - Separate stories that are news from those that aren't.
    - Query: *"facet.query":"newssite_filter:news"* and *"facet.field":"story_type"*.
- In the frontend:
  - Provides the count of documents for each category
  - Lets user filter results that fit on a given bucket.

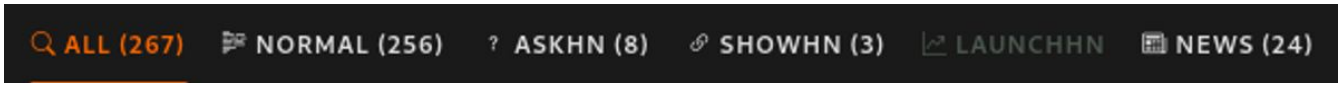

Figure 8: Frontend faceting usage example

Highlighting

- Solr provides a *HighlighterComponent*.
  - Stands out the matched content of a query.
- Highlighting method: *Unified Highlighter*
  - Displays the actual Lucene matches.
  - Produced the best results.
- In the frontend:
  - Highlighted contents in the query results are used to highlight all relevant matches to the user.

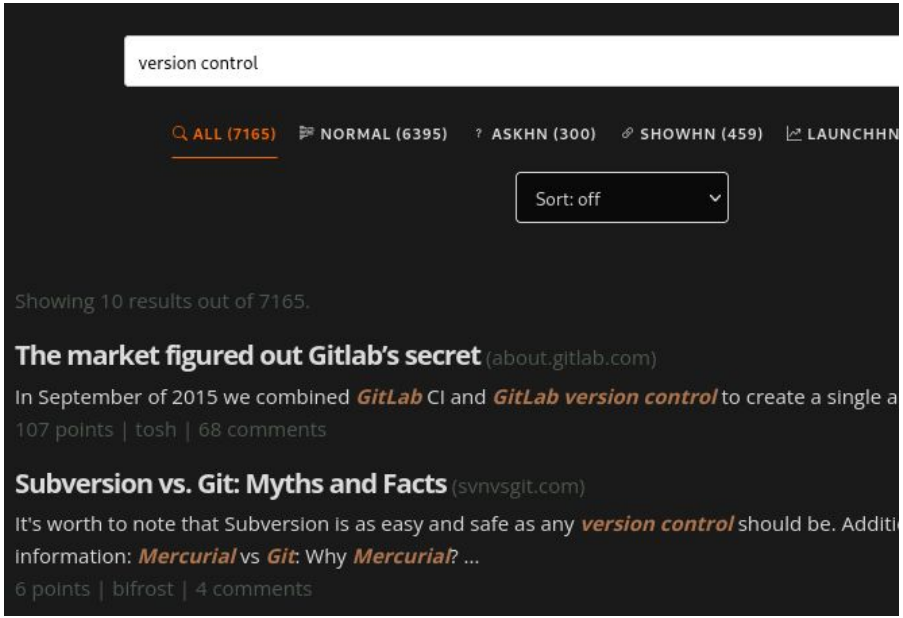

Figure 9: Frontend highlighting example

Synonym system revisited

- Synonym lists introduced a problem:
  - Original keyword search could be considered less relevant because synonyms were more popular.
- Solution: **Special weight system** for the synonyms.
  - Boost original keyword by 2.0.
- Example for "*git*" search:
  - Synonyms aren't boosted: *version control, GitHub, GitLab…*
  - *git* keyword is boosted by 2.0.
  - Original keyword is considered to be twice as important as its synonyms.
- Synonym system expanded:
  - More synonyms added.
  - Split *version control* synonyms into multiple synonym associations to apply boosting

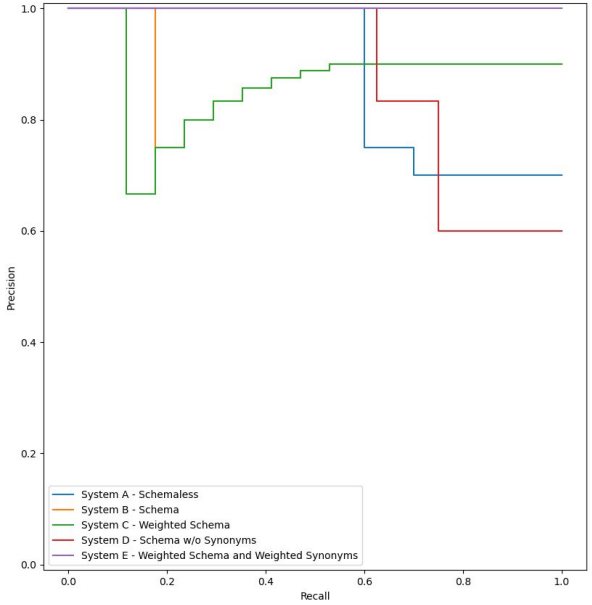

Figure 10: New PR-curve for *version control* query

More like this (/mlt)

- New request handler:
  - Takes a query that matches a single document.
  - Return similar documents.
- Stories comparison uses 4 fields: title, content, type, and URL's domain.
  - Term vectors generated for these fields ⇒ higher efficiency and accuracy.
- Problem: Too many documents being returned as similar when they are not.
- Cause: Stop words are kept on the title field, but they shouldn't be considered when calculating similarity.
- Solution:
  - Discard words with less than 3 characters ⇒ eliminates smaller stop words.
  - Discard words that appear in more than 10% of the documents ⇒ eliminates the most common stop words.

Stop word	Frequency (%)
a	17%
and	10%
but	1%
how	5%
or	1%
what	4%
will	1%
the	21%
i	3%
if	1%

Figure 11: Stop word frequency.

Story	Most similar
During a Solar **Eclipse**…	…yesterday's total lunar **eclipse**
Firefox … Sideloaded Extensions	Blocking cryptominers … Firefox

Figure 12: Similar stories example.

Future Work

- Improve dataset.
  - Collect all types of posts from all time.
  - Collect all comments (instead of just the top two).
  - Find the real descendants count of comments
  - Improve web scrapping:
    - Handle all websites (e.g.: *youtube.com*).
    - Process binary data (e.g.: convert **PDF files to text**).
- Identify *niche* stories.
  - Allow users to find results about newer/unknown/unpopular tools and topics.

Thanks for your attention