

# SEUPD@CLEF Task 1: Information retrieval for English and French documents: Team JIHUMING

Jes s Moncada Ram rez<sup>1</sup>, Isil Atabek<sup>1</sup>, Huimin Chen<sup>1</sup>, Michele Canale<sup>1</sup>, Nicol  Santini<sup>1</sup> and Giovanni Zago<sup>1</sup>

## Abstract

Our group will propose an original and efficient information retrieval system for Longitudinal Evaluation of Model Performance (LongEval) by CLEF2023[1]. Focus is on short term and long term temporal persistence of the systems' performance, for both English and French documents. The aim is to find a model giving good results for longitudinal evolving benchmarks, for the subject Search Engines, University of Padova.

## Keywords

CLEF 2023, Information retrieval, LongEval, English, French, Search Engines

## 1. Introduction

This report aims at providing a brief explanation of the Information Retrieval system built as a team project during the Search Engine course 22/23 of the master's degree in Computer Engineering and Data Science at University of Padua, Italy. Task chosen by the group is CLEF LongEval: Longitudinal Evaluation of Model Performance.

Longeval Websearch collection[2] relies on a large set of data provided by a commercial Qwant (a commercial search engine). The idea is to reflect changes of the Web across time, providing evolving document and query sets.

Our approach uses on N-grams, trying to avoid the problem of the sparsity of the data. Our focus was not to produce a high scoring specialized system, but to try developing general ideas in order to approach sentences selection and query expansion.

The paper is organized as follows: Section 2 describes our approach; Section 3 explains our experimental setup; Section 4 discusses our main findings; finally, Section 5 draws some conclusions and outlooks for future work.


---

*"Search Engines", course at the master degree in "Computer Engineering", Department of Information Engineering, and at the master degree in "Data Science", Department of Mathematics "Tullio Levi-Civita", University of Padua, Italy. Academic Year 2022/2023*

✉jesus.moncadaramirez@studenti.unipd.it (J. M. Ram rez); isil.atabek@studenti.unipd.it (I. Atabek); huimin.chen@studenti.unipd.it (H. Chen); michele.canale.1@studenti.unipd.it (M. Canale); nicolo.santini.1@studenti.unipd.it (N. Santini); giovanni.zago.3@studenti.unipd.it (G. Zago)



  2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

## 2. Methodology

In this section we address how the system was developed starting from the source code of HelloTipster[3] repository developed by Professor Nicola Ferro and presented to us during the Search Engine course. Furthermore, we will present main methodology and approaches using the same structure of the repository[4].

### 2.1. Parsing

We had a huge collection of documents in English and French. First thing was to manually examine document to understand how to read them. As described in his [GitHub repository](#), documents use a particular format, which includes a DOCNO and a DOCID. The DOCNO is the id of the document and the DOCID is the id of the collection.

JSON, on the other side, follows the much more standard following structure.

The whole parser is made by:

- DocumentParser: parses trec document
- JsonDocument: create a class for json doc
- LongEvalParser: counts the document and print out
- ParsedDocument: document parsed has FIELDS including ID, ENGLISH\_BODY, and FRENCH\_BODY

### 2.2. Analyzer

As basis for the analyzer we used the TokenStream class from Lucene, which Consumes a TokenStream for the given text by using the provided Analyzer and prints diagnostic information about all the generated tokens and their Attributes.

Streams are analyzed by applying these filters:

- WhitespaceTokenizer: splits on and discards only whitespace characters
- PatternReplaceFilter: it's applied twice. First time, using RegExpr patter deletes punctuations marks at the beginning of tokens, second time it does the same but at the end of them
- WordDelimiterGraphFilter: it splits words into subwords and performs optional transformations on subword groups. In our case, we decided to use these filters:

```
WordDelimiterGraphFilter.GENERATE_WORD_PARTS
    // Ex: "PowerShot" => "Power" "Shot"
| WordDelimiterGraphFilter.GENERATE_NUMBER_PARTS
    // Ex: "500-42" => "500" "42"
| WordDelimiterGraphFilter.CATENATE_NUMBERS
    // Ex: "500-42" => "50042"
| WordDelimiterGraphFilter.PRESERVE_ORIGINAL
    // Ex: "500-42" => "500" "42" "500-42"
| WordDelimiterGraphFilter.SPLIT_ON_CASE_CHANGE
    // Causes lowercase -> uppercase transition to start a new subword.
| WordDelimiterGraphFilter.STEM_ENGLISH_POSSESSIVE
    // "O'Neil's" => "O", "Neil"
```

- `LowerCaseFilter`: converts all characters to lowercase
- `StopFilter`: removes stop words applying terrier list[5]
- `SynonymTokenFilter`: only for English analyzer, it uses WordNet lexical database[6] to group words interlinked by means of conceptual-semantic and lexical relations
- `MinimalStemFilter`: implementing S-Stemmer from Harman article[7] for English language and Savoy stemming procedure[8], it divides words into stems
- `EmptyTokenFilter`: removes tokens with length 0

Lastly, the `NGramAnalyzer` is applied, trying tokens of one, two and three words. We decided to set the max at three, but this number is heuristic and can be changed if results shows better numbers.

### 2.3. Index

We used the standard Lucene Indexer with the BM25[9] similarity.

We decide to use `NGram` analyzer with a multilingual indexer, in order to index two version of the same document, creating documents with an unique ID and bodies from each language. Ngrams are represented as characters from both English and French version of the documents. N parameter is set at the analyzer level and class is a field, not stored, in order to minimize space occupation.

### 2.4. Search

### 2.5. Topic

## 3. Experimental Setup

Describe the experimental setup, i.e.

- used collections
- evaluation measures
- url to git repository and its organization
- hardware used for experiments
- ...

## 4. Results and Discussion

Provide a summary of the performance on the previous year dataset.

Discuss the results and any relevant issues.

## 5. Conclusions and Future Work

Provide a summary of what are the main achievements and findings.

Discuss future work, e.g. what you may try next and/or how your approach could be further developed.

## References

- [1] CLEF2023, LongEval CLEF 2023 Lab, <https://clef-longeval.github.io/>, 2023.
- [2] CLEF2023, LongEval Train Collection, <https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-5010>, 2023.
- [3] Nicola Ferro, hello-tipser, <https://bitbucket.org/frncl/se-unipd/src/master/hello-tipster/>, 2023.
- [4] Canale, Chen, Ramirez, Santini and Zago, jihuming, <https://bitbucket.org/upd-dei-stud-prj/seupd2223-jihuming/src/master/>, 2023.
- [5] I. Brigadir, Default english stop words from different sources, <https://github.com/igorbrigadir/stopwords>, 2023.
- [6] Princeton University, About WordNet, <https://wordnet.princeton.edu/download/current-version>, 2005.
- [7] D. Harman, How effective is suffixing?, *Journal of the American Society for Information Science* 42 (1991) 7–15.
- [8] J. Savoy, A stemming procedure and stopword list for general french corpora, *J. Am. Soc. Inf. Sci.* 50 (1999) 944.
- [9] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: *Text Retrieval Conference*, 1994.