SEUPD@CLEF: Team JIHUMING on Enhancing Search Engine Performance with Character N-Grams, Query Expansion, and Named Entity Recognition

Notebook for the LongEval Lab at CLEF 2023

Isil Atabek[1], Huimin Chen[1], Jesús Moncada-Ramírez[1], Nicolò Santini[1], Giovanni Zago[1] and Nicola Ferro[1]

[1]*University of Padua, Italy*

Abstract

Our group will propose a novel search engine for the Longitudinal Evaluation of Model Performance (LongEval) task at CLEF 2023 [1]; it will also be the final work of the subject Search Engines at the University of Padova. Our system focuses on the short-term and long-term temporal persistence of the systems' performance, for a collection of both English and French documents. Our approach involves considering both English and French versions of the documents using whitespace tokenization, stopword removal and stemming. We generate character N-grams to identify recurring word structures (as prefixes or suffixes) repeated over documents. We also use query expansion with synonyms (in English) and some Natural Language Processing (NLP) techniques as Named Entity Recognition (NER) to further refine our system. The similarity function utilized in our approach is BM25. Our system was developed in Java and primarily utilized the Lucene library. After extensive experiments on these techniques, we came up with five systems that have produced the best results in terms of MAP and NDCG scores. We analyzed these five selected systems by examining their MAP, NDCG, and Rprec scores on the test data. Additionally, we performed a Two-Way ANOVA to assess the AP of these systems. To compare our systems with each other, we will utilize the Tukey Honestly Significant Difference (HSD) test. In summary, our analysis indicates that incorporating French queries enhances search results, larger N-gram sizes contribute to improved effectiveness, while our NER approach negatively affects the scores.

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 the University of Padua, Italy. As a group in this subject, we are participating in 2023 CLEF LongEval: Longitudinal Evaluation of Model Performance [1].

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
isil.atabek@studenti.unipd.it (I. Atabek); huimin.chen@studenti.unipd.it (H. Chen); jesus.moncadaramirez@studenti.unipd.it (J. Moncada-Ramírez); nicolo.santini.1@studenti.unipd.it (N. Santini); giovanni.zago.3@studenti.unipd.it (G. Zago); ferro@dei.unipd.it (N. Ferro)
http://www.dei.unipd.it/~ferro/ (N. Ferro)
0000-0001-9219-6239 (N. Ferro)

This annual evaluation campaign focuses on evaluating the temporal persistence of information retrieval (IR) systems and text classifiers.

The LongEval collection relies on a large set of data provided by Qwant (a commercial privacy-focused search engine that was launched in France in 2013). Their idea regarding the dataset was to reflect changes of the Web across time, providing evolving document and query sets. The train collection [2] consists of 1,570,734 documents, 672 queries, 98 held-out queries, and 9656 evaluation assessments. The documents were chosen based on queries using the Qwant click model, in addition to random selection from the Qwant index. The queries are categorized into twenty topics, such as: car-related, antivirus-related, employment-related, energy-related, recipe-related, etc. In addition to the original French version, the collection also includes English translations of the documents and queries using the CUBBITT [3] system. The test collection [4] for the short-term persistence sub-task was gathered during July 2022, comprising 1,593,376 documents and 882 queries. The test collection for the long-term persistence sub-task was collected in September 2022, containing 1,081,334 documents and 923 queries. We will also use the test collection evaluation assessments [5] provided for the short-term, long-term and held-out queries.

The paper is organized as follows: Section 2 introduces related works; Section 3 briefly describes our approach; Section 4 describes our code in detail; Section 5 explains our experimental setup; Section 6 discusses how we selected and which are the five systems submitted to the competition, in addition to an analysis on the test collections and our main findings; finally, Section 7 draws some conclusions and outlooks for future work.

2. Related Work

There are many search engines using different techniques to enhance retrieval effectiveness from which we have taken inspiration from.

The BM25 [6] similarity function is widely used information retrieval as it considers term frequencies and document length. This function has demonstrated effectiveness in balancing precision and recall in search results, even if it doesn't consider meta-data document information as other approaches [7] do. Whitespace tokenization has not been the primary focus of research in any study, anyway, it seems to provide a useful baseline for tokenization. In contrast, E. Gow-Smith et al. [8] suggests that allowing tokens to include spaces causes problems, especially in architectures including transformers. Token lowercasing is also a recurring method in information retrieval [9], mainly because it reduces the vocabulary size. In order to implement BM25 similarity function and tokenization, both for English documents and for French documents, we used whitespace tokenization and lowercasing using libraries provided by Lucene. We implemented the tokenization in order to get advantages in the vocabulary size, using it in each class which implement the analyzer for that specific documents set. While our system adopts whitespace tokenization, we acknowledge these concerns and are open to exploring alternative tokenization methods in future iterations.

The Terrier [10] stopword list has been used in plenty of search engines because of the good results if offers working with web documents as blogs [11] or even recommender systems. For French documents we could not use Terrier stopword list, so we had to find another one[12]. Stopowords list are stored as .txt files wich we read and use in analyze classes. Each class obviously will read French stopword list if the document set is the French one, otherwise it use the Terrier stopword list.

Another basic information retrieval technique used in our search engine has been stemming. We have relied on the work of A. G. Jivani et al.[13] to get an overview of the most adequate stemming techniques for our documents. For the English documents we have chosen a minimal stemmer developed by D. K. Harman [14]. For the French documents we have also used a minimal stemmer developed by J. Savoy [15]. Also stemmers are implemented in analyzer classes and are used the minimal stemmers classes provided by Lucene library.

We have used query expansion [16] in order to broaden the search scope by including synonyms related to the original query. These synonyms come from WordNet [17], a popular lexical database that provides semantic relationships between words.

We have also included character N-grams of the English and French versions of the documents. Our experiments on character N-grams have been focus on comparing how the value of N can affect to the retrieval effectiveness. Our motivation for this study stemmed from the works of T. Wilson et al. [18], and J. Goodman [19], which also explored the impact of different N-gram models (among others) on performance. We tried to refine our results including Named Entity Recognition [20]. This technique has proven useful in other information retrieval systems addressing for example the food [21] or the archaeology [22] domain. We implemented character N-gram and NER in separated classes, used only for these scopes. In order to appreciate the results of doing different running and calculate scores for each one, we can choose in the class builder the number N of character. Doing this it is possible to appreciate the performance differences between different values of N. Synonyms are instead implemented only for the English analyzer every time an EnglishAnalyzer is instantiated it try to read and map a synonyms list.

The work from F. Cai et al. [23] remarks the importance of understanding query temporal dynamics for search result ranking.By considering the temporal patterns of queries and incorporating query temporal dynamics into the ranking process, search engines can deliver more relevant and timely results.An example of this is giving more weight to recent queries or adjusting the ranking based on a certain popularity during specific time periods. In the context of our task, with changing datasets, learning and estimating query temporal dynamics can be highly relevant.

The work from K. Hofmann et al. [24] presents valuable insights into evaluation methodologies for temporal aspects in web search systems. Specifically, the paper explores metrics for evaluating retrieval effectiveness over time, such as precision, recall, F-1 score, and mean average precision (MAP). The paper provides insights into various setups, including time-sliced evaluation, incremental evaluation, and evaluation with simulated temporal queries. These setups can serve as a basis for designing experimental setups that align with specific task requirements. With this motivation we can incorporate evaluation methodologies, metrics, and experimental setups specifically tailored for temporal information retrieval.

To compare, the work by F. Cai et al. [23] focuses on learning to estimate query temporal dynamics for web search. Their study aims to understand the temporal patterns of queries and incorporate this knowledge into the ranking process, thereby delivering more relevant and timely search results. This research addresses the importance of considering the temporal aspect of queries to improve retrieval effectiveness.

On the other hand, the work of K. Hofmann et al. [24] concentrates on evaluating web search systems while considering the dimension of time. Their paper explores evaluation methodologies specifically tailored to temporal aspects, such as time-sliced evaluation, incremental evaluation, and evaluation with simulated temporal queries. By proposing and examining these evaluation setups, the authors provide insights into assessing the performance of search engines in a temporal context.

When comparing the work of F. Cai et al. [23] and K. Hofmann et al. [24], it is evident that they address different aspects of temporal information retrieval. While F. Cai et al. focus on learning and estimating query temporal dynamics, K. Hofmann et al. concentrate on evaluating the effectiveness of search systems over time. Both papers contribute to the field of temporal information retrieval by providing novel insights and methodologies.

In the context of our search engine, we acknowledge the importance of understanding query temporal dynamics, as emphasized by F. Cai et al. [23]. By considering the temporal patterns of queries and incorporating them into our ranking process, we strive to deliver more relevant and timely search results. Furthermore, the evaluation methodologies presented by K. Hofmann et al. [23] serve as a valuable reference for assessing the performance of our search engine in a temporal context. While our specific implementations and techniques may differ, the underlying principles and motivations align with the contributions of these referenced papers in the field of temporal information retrieval.

3. Methodology

Our search engine can be divided into the following parts: parsing of the documents and queries, indexing, text processing (analyzers), and run generation (effective search).

Document **parsing** was performed using the JSON version of the documents. On the other hand, query parsing was based on an XML parser devloped by us.

In the **index**, we decided to include four fields: (1) the (processed) English version of the documents, (2) the (processed) French version, (3) character N-grams of both versions concatenated, and (4) some NER information extracted from the French (original) version. As similarity function we have used BM25 [6] as it takes into account both term frequency and document length.

The text in the fields of our indexes must first be processed, for this we have developed four different **analyzers**. The English analyzer is based on whitespace tokenization, breaking of words and numbers based on special characters, lowercasing, applying the Terrier stopword list, query expansion with synonyms based on the WordNet synonym map [25], and stemming. The French analyzer is based on whitespace tokenization, breaking of words and numbers based on special characters, lowercasing, applying a French stopword list [26] and stemming. To generate the character N-grams we consider only the letters of the documents (i.e. we discard numbers and punctuation). To perform NER we apply NLP techniques based on Apache OpenNLP [27], specifically, we used NER applied to locations, person names and organizations.

We conducted some experiments to generate the runs, i.e., we have tried different combinations of the explained techniques. Thus, our searcher will always use BM25 [6], but the rest of characteristics will depend on the run it is generating. See Section 5 for more details.

4. System Architecture

In this section, we address the technical aspects of how our system was developed following the structure (in packages) of the repository [28].

4.1. Parsing

To generate an index from the provided documents, we parse them by extracting their text into Java data structures. Our parser package is based on the JSON version of the documents, allowing easy manipulation and querying using various tools and libraries. We implemented a streaming parser using the `Gson` library in Java.

The whole parser is made up of the following classes:

- `DocumentParser`: An abstract class that represents a streaming parser and implements `Iterator` and Iterable.
- `JsonDocument`: a Java POJO for the deserialization of JSON documents.
- `ParsedDocument`: Represents a parsed document, containing an identifier and a body.
- `LongEvalParser`: Implements the `DocumentParser` class and handles the streaming logic. Objects of this class can be used as iterators to yield parsed documents

4.2. Analyzer

To process the parsed document text, we developed our own Lucene analyzers. Each analyzer follows a typical workflow involving a `Tokenizer` and a list of `TokenFilter` for a `TokenStream` To process the already parsed documents' text, we have implemented our own Lucene analyzers. All of them follow the typical workflow: use a `Tokenizer` and a list of `TokenFilter` to a `TokenStream`.

The project's final version creates an index with four fields for each document, requiring four different analyzers. The `AnalyzerUtil`described below utilize functionalities from the AnalyzerUtil helper class developed by Nicola Ferro.

4.2.1. English body field

The processing applied to the English version of the documents (using the `EnglishAnalyzer` class) includes:

1. Tokenize based on whitespaces.
2. Eliminate some strange characters found in the documents. It is unlikely that a user would perform a query including these characters.
3. Removal of punctuation marks at the beginning and end of words since whitespace tokenization is used.
4. Application of the `WordDelimiterGraphFilter` Lucene filter to split words into subwords based on case, divide numbers, concatenate numbers with special characters, and remove English possessive trailing "s.".
5. Lowercase all the tokens.
6. Apply the Terrier [10] stopword list.
7. Apply query expansion with synonyms using the `SynonymTokenFilter` from Lucene, based on the WordNet synonym map [25].
8. Apply minimal stemming using the `EnglishMinimalStemFilter`from Lucene.
9. Removal of empty tokens left by previous filters using a custom `EmptyTokenFilter`

4.2.2. French body field

The processing of French documents (in the class `FrenchAnalyzer` is identical to the processing of English documents in the first 5 points (excluding the English possessives' removal in 4.d). For this point on, we apply:

6. Apply a French stopword list [26].

7. Apply a minimal stemming process (in French) using `FrenchMinimalStemFilter` from Lucene.

8. Removal of empty tokens (`EmptyTokenFilter`).

4.2.3. Character N-grams

Character N-grams are created using the `NGramAnalyzer` class, which performs the following operations:

1. Tokenize based on whitespaces.

2. Lowercase all the tokens.

3. Removal of all characters except letters (including French accent letters).

4. Removal of empty tokens `EmptyTokenFilter`.

5. Generate character N-grams using `NGramTokenFilter` from Lucene.

The value of N has not been fixed in order to allow for the generation of different experiments. See Section 5 for more details.

4.2.4. NER extracted information

The NER information has been extracted using the Apache OpenNLP [27] library. As Lucene does not include these functionalities directly, we have used a modified version of a token filter developed by Nicola Ferro based on the mentioned library, (`OpenNLPNERFilter`).

The processing of the tokens in this analyzer (`NERAnalyzer`) is the following:

1. Tokenization using the `StandardTokenizer` from Lucene.

2. NER tagging using a model for locations.

3. NER tagging using a model for person names.

4. NER tagging using a model.

4.3. Index

Initially, we developed a `DirectoryIndexer` to handle single-language documents (English or French). However, when considering both versions, we deprecated it in favor of the final `MultilingualDirectoryIndexer`.

The `MultilingualDirectoryIndexer` is used for indexing multilingual documents and obtaining basic vocabulary statistics. To create an instance, parameters such as document directory paths, index directory path, expected document count, and custom analyzers (English-Analyzer, FrenchAnalyzer, NGramAnalyzer, and NERAnalyzer) are required. Additionally, the chosen similarity function (BM25) and the RAM buffer size for indexing must be specified.

During indexing, the `MultilingualDirectoryIndexer` reads documents from the English and French directories, processes them with the specified analyzers, and creates an inverted index. Both directories must contain the same number of files and documents with matching IDs. Each iteration combines the English and French versions of the same document into a single Lucene document in the index.

After indexing, we utilize a method to print vocabulary statistics, including unique terms, total terms, and frequency lists for English and French. This provides a useful overview for analysis and optimization of the search system. The indexer also estimates the remaining time required for indexing, addressing the time-consuming nature of the process.

4.4. Search

The `Searcher` class in the search package performs effective searches by applying the specified analyzers to the query title and matching it with the corresponding index fields. The `QueryParser` class from Lucene is utilized for this process. The search is conducted using the BM25 similarity function. Users can specify the index path, topics file path, number of expected topics, run descriptor, and the maximum number of documents to retrieve (1000). A user-friendly menu allows the selection of desired runs and distinguishes between train and test data.

4.5. Topic

To read queries (in TREC format) we developed our own LongEval topic reader (`LongEvalTopicReader`) in the topic package. It consists of the LongEvalTopic and LongEvalTopicReader classes. The `LongEvalTopic` represents each query with a number (<num>) and a title (`<title>`), serving as the equivalent of `QualityQuery` in `TrecTopicsReader`. The `LongEvalTopicReader` parses the query file as an XML file using the Java XML library.

5. Experimental Setup

In this section, we describe the experimental setup employed in our study.

5.1. Hardware and Software Environment

All the code and documentation related to the project were developed and stored in the group's repository (https://bitbucket.org/upd-dei-stud-prj/seupd2223-jihuming/src/master/) [28]. The repository, hosted on Bitbucket, provided a centralized location for accessing and managing the project's source code and documentation.

The development and experimentation phases of the project were conducted using personal computers. The specific software tools and versions used included Java JDK version 17, Apache version 2, Lucene version 9.5, and Maven. These tools and versions were employed to facilitate the implementation and execution of the experimental systems.

5.2. Evaluation Metrics

We computed the Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) scores for all our systems' runs on the training collection. These scores were used to select the top five systems to be submitted to CLEF. Additionally, MAP, NDCG, and Precision at the Recall base (Rprec) scores were computed for the submitted systems' runs on the test (short-term and long-term) and held-out collections. These metrics provided a reliable estimation of the final performance of our systems.

5.3. Statistical Analysis

We performed a statistical analysis using Two-Way ANOVA to assess the Average Precision (AP) of the five submitted systems across all the topics on the test and held-out collections. Furthermore, pairwise comparisons of the submitted systems were conducted using the Tukey Honestly Significant Difference (HSD) test.

5.4. Indexes

In order to do different run experiments, our team has created several indexes from each of the provided collections (train, short-term test, and long-term test). Put simply, certain indexes mentioned in this report incorporate only a few of the characteristics discussed, while others encompass all the characteristics outlined in the definitive version of the project.

All the created indexes are **multilingual**, which allows us to take full advantage of the (bilingual) data collection. Additionally, we did some experiments with character N-grams generating different versions of indexes with 3-grams, 4-grams and 5-grams; the motivation was to compare how this parameter affects to the effectiveness of our systems. 3-grams are able to collecting more local information in our documents, while 4-grams and 5-grams are more open to the context. An additional functionality of some indexes is query expansion, but as commented, this is only applied to the English body. One index includes Named Entity Recognition which provides not only the search for keywords but also identifying and extracting specific named entities. The subsequent indexes are:

- `multilingual_3gram`: both languages of documents, using character 3-grams.
- `multilingual_3gram_synonym`: both languages, character 3-grams, (English) query expansion with synonyms.
- `multilingual_4gram_synonym`: both languages, character 4-grams, (English) query expansion with synonyms.

- `multilingual_5gram_synonym`: both languages, character 5-grams, (English) query expansion with synonyms.
- `multilingual_4gram_synonym_ner`: both languages, character 4-grams, (English) query expansion with synonyms, NER techniques.

The indexes also can be found in the following Google Drive folder.

5.5. Runs

After creating the indexes, we were able to conduct multiple runs to evaluate the effectiveness of our system. These runs not only experiment with some techniques specified here, but also consider different versions (English or French) of the queries. With them, we can compare and analyze different aspects of our system's performance, such as precision and recall. The runs are the following:

- `seupd2223-JIHUMING-01_en_en`: English topics; using English body field.
- `seupd2223-JIHUMING-02_en_en_3gram`: English topics; using English body field and 3-gram field.
- `seupd2223-JIHUMING-03_en_en_4gram`: English topics; using English body field and 4-gram field.
- `seupd2223-JIHUMING-04_en_en_5gram`: English topics; using English body field and 5-gram field.
- `seupd2223-JIHUMING-05_en_en_fr_5gram`: English topics; using English and French body fileds and 5-gram field.
- `seupd2223-JIHUMING-06_en_en_4gram_ner`: English topics; using English body field, 4-gram field and NER information field.
- `seupd2223-JIHUMING-07_fr_fr`: French topics; using French body field.
- `seupd2223-JIHUMING-08_fr_fr_3gram`: French topics; using French body field and 3-gram field.
- `seupd2223-JIHUMING-09_fr_fr_4gram`: French topics; using French body field and 4-gram field.
- `seupd2223-JIHUMING-10_fr_fr_5gram`: French topics; using French body field and 5-gram field.
- `seupd2223-JIHUMING-11_fr_en_fr_5gram`: French topics; using English and French body fields and 5-gram field.
- `seupd2223-JIHUMING-12_fr_fr_4gram_ner`: French topics; using French body field, 4-gram field and NER information field.

The process of creating the indexes typically took around 1 hour, except the indexes that included NER, which took approximately 16 hours. On the other hand, generating the runs was a much quicker process, taking consistently less than a minute and a half to complete.

The mentioned analysis of the runs on the training collection will take place in Section 6.1. The analysis of the runs on the test collection will take place in Section 6.

6. Results Analysis

In this section, we first present the runs that achieved the highest and lowest scores on the train and test data. Subsequently, we provide an interpretation and analysis of these score values (see Section 6.5).

6.1. Training Data and Selection of Runs to Submit

Table 1
MAP and NCDG scores for all the runs on the training collection

Run ID	Run	MAP Score	NCDG Score
01	en_en	0.0700	0.1614
02	en_en_3gram	0.0704	0.1661
03	en_en_4gram	0.0874	0.2025
04	en_en_5gram	0.1028	0.2288
05	en_en_fr_5gram	0.0669	0.1525
06	en_en_4gram_ner	0.0360	0.1098
07	fr_fr	0.1656	0.3135
08	fr_fr_3gram	0.1698	0.3208
09	fr_fr_4gram	0.1737	0.3269
10	fr_fr_5gram	0.1748	0.3285
11	fr_en_fr_5gram	0.1288	0.2797
12	fr_fr_4gram_ner	0.1362	0.2881

The analysis shows that, on training data, the highest MAP score (0.1748) is achieved by `10_fr_fr_5gram`, followed by `09_fr_fr_4gram` (0.1737), `08_fr_fr_3gram` (0.1698), `07_fr_fr` (0.1656), and `12_fr_fr_4gram_ner` (0.1362). The lowest MAP score (0.0360) is obtained by `en_en_4gram_ner`.

Similarly, the highest NDCG score (0.3285) belongs to `10_fr_fr_5gram`, followed by `09_fr_fr_4gram` (0.3269), `08_fr_fr_3gram` (0.3208), `07_fr_fr` (0.3135), and `12_fr_fr_4gram_ner` (0.2881). The lowest NDCG score (0.1098) corresponds to `en_en_4gram_ner`.

Because of this, the five system submitted to CLEF have been (in order of importance): `10_fr_fr_5gram`, `09_fr_fr_4gram`, `08_fr_fr_3gram`, `07_fr_fr`, and `12_fr_fr_4gram_ner`. Following the competition workflow, we created new indexes based on the test collection and re-executed this top five runs.

6.2. Short Term Test Data

Table 2 presents the mentioned scores computed for our five submitted systems on the (test) short-term collection.

In the Two-Way ANOVA analysis presented in Table 3 we observed a significant p-value ($p < 0.05$). Thus, we can conclude that there are significant differences among our systems. As ANOVA does not tell which systems are significantly different from each other, in Table 4 we can observe the Tukey's Honestly Significantly Differenced (HSD) test. It suggests that pairwise comparisons between systems 7–12, 8–12, 9–12 and 10–12 reject null hypothesis ($p < 0.05$) and indicate statistical significant differences.

Table 2
MAP, NCDG and Rprec scores for the submitted runs on the (test) short-term collection

Run ID	Run	NCDG Score	MAP Score	RPREC Score
07	fr_fr	0.3367	0.1883	0.1561
08	fr_fr_3gram	0.3384	0.1893	0.1579
09	fr_fr_4gram	0.3423	0.1911	0.1581
10	fr_fr_5gram	0.3447	0.1926	0.1603
12	fr_fr_4gram_ner	0.2980	0.1468	0.1172

Table 3
Two-Way ANOVA table assessing AP for the submitted systems on the (test) short-term collection

Source	DF	SS	MS	F	PR(>F)
C(system)	4	1.348254	0.337063	6.250613	0.000052
Error	4405	237.539013	0.053925	–	–
Total	4409	238.887267	–	–	–

Table 4
Tukey Honestly Significant Difference test for the submitted systems on the (test) short-term collection

Run 1	Run 2	Diff	Lower	Upper	q-value	p-value
07_fr_fr	08_fr_fr_3gram	0.000986	-0.029190	0.031162	0.126122	0.900000
07_fr_fr	09_fr_fr_4gram	0.002808	-0.027368	0.032984	0.359124	0.900000
07_fr_fr	10_fr_fr_5gram	0.004282	-0.025894	0.034458	0.547611	0.900000
07_fr_fr	12_fr_fr_4gram_ner	0.041538	0.011362	0.071714	5.312288	0.001641
08_fr_fr_3gram	09_fr_fr_4gram	0.001822	-0.028354	0.031998	0.233002	0.900000
08_fr_fr_3gram	10_fr_fr_5gram	0.003296	-0.026880	0.033472	0.421489	0.900000
08_fr_fr_3gram	12_fr_fr_4gram_ner	0.042524	0.012348	0.072700	5.438410	0.001152
09_fr_fr_4gram	10_fr_fr_5gram	0.001474	-0.028702	0.031650	0.188487	0.900000
09_fr_fr_4gram	12_fr_fr_4gram_ner	0.044346	0.014170	0.074522	5.671412	0.001000
10_fr_fr_5gram	12_fr_fr_4gram_ner	0.045820	0.015644	0.075995	5.859899	0.001000

6.3. Long Term Test Data

Table 5 presents the mentioned scores computed for our five submitted systems on the (test) long-term collection.

In the Two-Way ANOVA analysis presented in Table 6 we observed a significant p-value ($p < 0.05$), so we can conclude that there are significant differences among our systems. The Tukey's Honestly Significantly Differenced (HSD) test in Table 7 again suggests that pairwise comparisons between systems 7–12, 8–12, 9–12 and 10–12 reject null hypothesis ($p < 0.05$).

Table 5
MAP, NCDG and Rprec scores for the submitted runs on the (test) long-term collection

Run ID	Run	NCDG Score	MAP Score	RPREC Score
07	fr_fr	0.3447	0.1880	0.1589
08	fr_fr_3gram	0.3454	0.1881	0.1600
09	fr_fr_4gram	0.3480	0.1888	0.1611
10	fr_fr_5gram	0.3533	0.1920	0.1642
12	fr_fr_4gram_ner	0.3046	0.1433	0.1192

Table 6
Two-Way ANOVA table assessing AP for the submitted systems on the (test) long-term collection

Source	DF	SS	MS	F	PR(>F)
C(system)	4	1.564386	0.391097	8.562506	7.01E-07
Error	4610	210.56392	0.045675	–	–
Total	4614	212.128306	–	–	–

Table 7
Tukey Honestly Significant Difference test for the submitted systems on the (test) long-term collection

Run 1	Run 2	Diff	Lower	Upper	q-value	p-value
07_fr_fr	08_fr_fr_3gram	0.000340	-0.026808	0.027488	0.048345	0.900000
07_fr_fr	09_fr_fr_4gram	0.001090	-0.026058	0.028237	0.154891	0.900000
07_fr_fr	10_fr_fr_5gram	0.004239	-0.022909	0.031387	0.602592	0.900000
07_fr_fr	12_fr_fr_4gram_ner	0.044458	0.017311	0.071606	6.319943	0.001000
08_fr_fr_3gram	09_fr_fr_4gram	0.000750	-0.026398	0.027897	0.106546	0.900000
08_fr_fr_3gram	10_fr_fr_5gram	0.003899	-0.023249	0.031047	0.554247	0.900000
08_fr_fr_3gram	12_fr_fr_4gram_ner	0.044798	0.017651	0.071946	6.368287	0.001000
09_fr_fr_4gram	10_fr_fr_5gram	0.003149	-0.023998	0.030297	0.447701	0.900000
09_fr_fr_4gram	12_fr_fr_4gram_ner	0.045548	0.018400	0.072696	6.474833	0.001000
10_fr_fr_5gram	12_fr_fr_4gram_ner	0.048697	0.021550	0.075845	6.922534	0.001000

6.4. Held Out Test Data

Table 8 presents mentioned scores computed for our five submitted systems on the (train) held-out collection.

In the Two-Way ANOVA analysis presented in Table 9 we didn't observe a significant p-value ($p >= 0.05$), so we can conclude that there are not significant differences among our systems.

Table 8
MAP, NCDG and Rprec scores for the submitted runs on the training collection, held-out evaluation

Run ID	Run	NCDG Score	MAP Score	RPREC Score
07	fr_fr	0.3271	0.1746	0.1397
08	fr_fr_3gram	0.3307	0.1725	0.1326
09	fr_fr_4gram	0.3364	0.1763	0.1397
10	fr_fr_5gram	0.3413	0.1788	0.1385
12	fr_fr_4gram_ner	0.2868	0.1369	0.1004

Table 9
Two-Way ANOVA table assessing AP on the training collection, held-out evaluation

Source	DF	SS	MS	F	PR(>F)
C(system)	4	0.11896	0.02974	0.689161	0.599712
Error	485	20.929668	0.043154	–	–
Total	489	21.048628	–	–	–

6.5. Discussion

Results suggest that French queries perform better than their English counterparts, possibly due to the training data's French origin and later translation into English. Moreover, the IR system's effectiveness generally increases with a larger N-gram size, as indicated by the higher scores of `04_en_en_5gram` and `10_fr_fr_5gram`. Conversely, the inclusion of NER in the indexing process seems to have a negative impact on the scores, as shown by the lower scores of `06_en_en_4gram_ner` and `12_fr_fr_4gram_ner`. The use of query expansion with synonyms in English does not seem to improve the search results to any great extent.

It's interesting to notice that the cross-language approaches (`05_en_en_fr_5gram` and `1_fr_en_fr_5gram`) are out of the five bests systems. It turns out that searching for English words in French documents and vice versa messes up the search, lowering the score. Another interesting aspect is that the worst-performing index is the one with named entity recognition in English (`06_en_en_4gram_ner`): it combines translated queries and NER, which appears to be the two worst-performing approaches.

Our findings indicate that the MAP and NDCG scores in the training data closely align with those in the test data. Interestingly, there are instances where the scores in the test data show improvement compared to the training data. Thus, according to the test data, the ranking of our best-performing systems is the same as in Section 6.1 (in order of importance): `10_fr_fr_5gram`, `09_fr_fr_4gram`, `08_fr_fr_3gram`, `07_fr_fr`, and `12_fr_fr_4gram_ner`.

Based on the ANOVA analysis, the null hypothesis is only not rejected in the runs on the held-out collection, suggesting no statistically significant difference. We attribute this finding to the limited number of queries in that particular collection. However, in cases where a statistical significant difference exists among the systems, it appears that system number 12 (`12_fr_fr_4gram_ner`) is the only one with a significant impact. This aligns with our expectations since system number 12 is the only submission utilizing NER among the others.

In general, we focused on trying multiple approaches, this is why our score has such a big space for improvement. As already said, French queries with bigger N-gram sizes perform better. In this system, instead of relying on single-word matches, the queries take place with more context, resulting in better search results.

7. Conclusions and Future Work

In summary, the IR systems developed in this study followed the Parsing-Analyzer-Index-Search-Topic paradigm and utilized different methodologies, among which the following stand out: processing of English documents based on whitespace tokenization, the TERRIER stopword list, query expansion, and stemming; processing of French documents based on whitespace tokenization, a stopword list and stemming; character N-grams of both versions concatenated; and NER information extraction using NLP techniques.

In terms of future work, there are several areas that could be explored to improve the effectiveness of the developed IR systems. Firstly, we could improve indexing methodologies, such as increasing the value of N in N-gram, as we have commented on in Section 6. Secondly, we could explore better NLP techniques to improve the accuracy of the IR systems, as NER turns out not to be very effective.

Overall results suggests we did not achieve the best possible results, since we preferred to explore multiple simple approaches rather than a single complex one. In most cases ANOVA results shows that we don't have a significant difference between the systems, so that we could start from any of them and improve.

One last possible future work could be a machine-learning based IR system. This would be a more dynamic approach to IR systems, as it would be able to adapt to different types of queries and documents.

References

[1] CLEF2023, LongEval CLEF 2023 Lab, https://clef-longeval.github.io/, 2023.

[2] P. Galuščáková, R. Devaud, G. Gonzalez-Saez, P. Mulhem, L. Goeuriot, F. Piroi, M. Popel, LongEval train collection, 2023. URL: http://hdl.handle.net/11234/1-5010, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[3] M. Popel, M. Tomkova, J. Tomek, Ł. Kaiser, J. Uszkoreit, O. Bojar, Z. Žabokrtský, Transforming machine translation: a deep learning system reaches news translation quality

comparable to human professionals, Nature Communications 11 (2020) 1–15. URL: https://www.nature.com/articles/s41467-020-18073-9. doi:10.1038/s41467-020-18073-9.
[4] P. Galuščáková, R. Devaud, G. Gonzalez-Saez, P. Mulhem, L. Goeuriot, F. Piroi, M. Popel, LongEval test collection, 2023. URL: http://hdl.handle.net/11234/1-5139, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
[5] P. Galuščáková, R. Devaud, G. Gonzalez-Saez, P. Mulhem, L. Goeuriot, F. Piroi, M. Popel, LongEval click-model relevance judgements (qrels), 2023. URL: http://hdl.handle.net/11234/1-5151, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
[6] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Text Retrieval Conference, 1994.
[7] S. Robertson, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. URL: http://scholar.google.de/scholar.bib?q=info:U4l9kCVIssAJ:scholar.google.com/&output=citation&hl=de&as_sdt=2000&as_vis=1&ct=citation&cd=1.
[8] E. Gow-Smith, H. T. Madabushi, C. Scarton, A. Villavicencio, Improving tokenisation by alternative treatment of spaces, 2022. arXiv:2204.04058.
[9] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK, 2008. URL: http://nlp.stanford.edu/IR-book/information-retrieval-book.html.
[10] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, C. Lioma, Terrier: A High Performance and Scalable Information Retrieval Platform, in: M. Beigbeder, W. Buntine, W. G. Yee (Eds.), Proc. of the ACM SIGIR 2006 Workshop on Open Source Information Retrieval (OSIR 2006), 2006.
[11] I. Soboroff, C. Macdonald, I. Ounis, Overview of the trec-2009 blog track, 2010. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=905208.
[12] genediazjr, Stopwords French (FR), https://github.com/stopwords-iso/stopwords-fr/blob/master/stopwords-fr.txt, 2023.
[13] A. G. Jivani, et al., A comparative study of stemming algorithms, Int. J. Comp. Tech. Appl 2 (2011) 1930–1938.
[14] D. K. Harman, How effective is suffixing?, J. Am. Soc. Inf. Sci. 42 (1991) 7–15.
[15] J. Savoy, A stemming procedure and stopword list for general french corpora, J. Am. Soc. Inf. Sci. 50 (1999) 944–952.
[16] E. N. Efthimiadis, Query expansion., Annual review of information science and technology (ARIST) 31 (1996) 121–87.
[17] C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, Language, Speech, and Communication, MIT Press, Cambridge, MA, 1998.
[18] T. Wilson, S. Raaijmakers, Comparing word, character, and phoneme n-grams for subjective utterance recognition, 2008, pp. 1614–1617. doi:10.21437/Interspeech.2008-270.
[19] J. Goodman, A bit of progress in language modeling, 2001. arXiv:cs/0108005.
[20] B. Mohit, Named entity recognition, Natural language processing of semitic languages (2014) 221–245.
[21] G. Popovski, B. K. Seljak, T. Eftimov, A survey of named-entity recognition methods for

food information extraction, IEEE Access 8 (2020) 31586–31594.
[22] A. Brandsen, S. Verberne, K. Lambers, M. Wansleeben, Can bert dig it? named entity recognition for information retrieval in the archaeology domain, Journal on Computing and Cultural Heritage (JOCCH) 15 (2022) 1–18.
[23] F. Cai, H. Wang, C. Zhai, Learning to estimate query temporal dynamics for web search, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 373–382.
[24] K. Hofmann, S. Whiteson, M. de Rijke, Evaluating web search systems considering time, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 321–330.
[25] Princeton University, About WordNet, https://wordnet.princeton.edu/download/current-version, 2005.
[26] G. Diaz, A. Suriyawongkul, Stopword list in French, https://github.com/stopwords-iso/stopwords-fr/tree/master, 2023.
[27] Apache, Apache OpenNLP, https://opennlp.apache.org/, 2023.
[28] Atabek, Canale, Chen, Moncada-Ramirez, Santini and Zago, JIHUMING, https://bitbucket.org/upd-dei-stud-prj/seupd2223-jihuming/src/master/, 2023.