Information Retrieval
2023-2024
Fact Finder - Fact Search Engine

Nicolò Urbani - 856213
Mattia Piazzalunga - 851931

February 21, 2024

1 Introduction

The identification of what is real or artificially created is undoubtedly a crucial issue in contemporary society. The advent of social media has accelerated the dissemination of information, but at the same time, it has also led to the spread of unverified facts, artificially created content, and misinformation that do not correspond to reality, thus introducing artificial biases into society.
The task of fact-checking plays a fundamental role in ensuring the quality of information, which extends beyond the boundaries of social media. In a context where news can be distorted, manipulated, or even created ad hoc to influence public opinion, fact-checking emerges as a tool of paramount importance in keeping society informed and critical-thinking.
Although fact-checking is a necessity, according to a December 2021 Eurostat survey [1], only 23% of individuals aged 16 to 74 had verified the accuracy of information or content in the three months prior to the survey, despite 47% of them encountering false or dubious information from news websites or social media. Providing accurate information to support or refute a fact, therefore, becomes a fundamental challenge that heavily involves research and can mitigate their impact on society. Can an effective retrieval be part of the solution?
In this project, the focus is on retrieving verified facts based on a well-known benchmark dataset: FEVER (Fact Extraction and Verification), Thorne et al. 2018 [2]. It is based on 185,445 manually annotated claims, categorizing them into three categories: Supported, Refuted, or Not Enough Info. FEVER has also become a reference dataset for IR and supports our goal: building a fact retrieval system.

2 Test Collection - Analysis of Queries and Documents

The FEVER collection [2] consists of 185,445 statements annually annotated as ”SUPPORTED,” ”REFUTED,” or ”NOT ENOUGH INFO.” Associated with instances of the first two classes, the dataset provides the combination of sentences that constitute the evidence required to support or refute the statement. The annotation process was complex and supported by a team of experts, with certified quality, as reported in Thakur et al. paper from 2021 [**?**].
Although it is important to cite the origin of this significant dataset, in our task, we need the version supportive of IR. The ”new” FEVER dataset contains queries (query ID, text), qrels (query ID, document ID, relevance, iteration), and documents (document ID, text, title) and is constructed only on statements labeled SUPPORTED and REFUTED. We report, therefore, the process that led to the generation of the ”new dataset in figure 1.

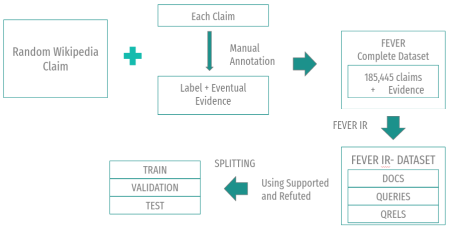


Figure 1: Dataset Generation Process for FEVER

The collection presented at the TREC conference consists of train, validation, and test sets, enabling fair evaluations of different IR models.

Table 1: Datasets Size.

Col1	Train Set	Validation Set	Test Set
Documents	5.4M	5.4M	5.4M
Qrels	140K	8.1K	7.9K
Queries	110K	6.7K	6.7K

As you can see, the FEVER document corpus consists of more than 5 million documents. Due to computational resource limitations, in training IR models, we will only use the text of the news, excluding the title.

Let’s proceed with a quick exploratory analysis.

As observable in Figure 2, the queries used for training, validation, and testing exhibit similar lengths, which helps limit biases in model training.

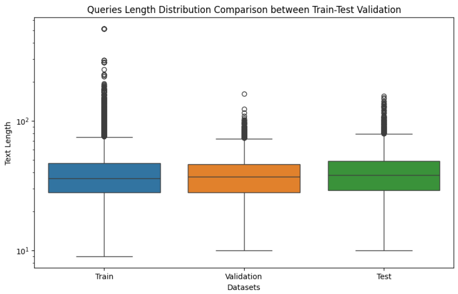


Figure 2: Distribution of Box Plot Length

Furthermore, examining the word cloud of queries in Figure 3, it's apparent that there are frequent words that do not significantly contribute to the semantics and discrimination of documents.

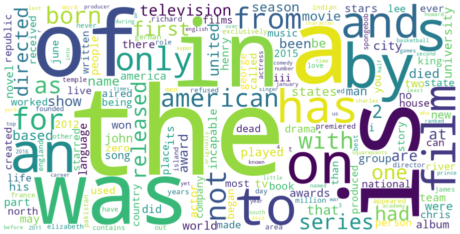


Figure 3: Distribution of Words in the Test Query Set

3 Search Engine - Basic Search

Let's describe, before focusing into details, the workflow and analysis applied in this paper in general terms:

- Preprocessing;
- Document Indexing;
- Utilization of basic retrieval models and analysis of results;
- Query Expansion and analysis of results;
- Neural reranking models and analysis of results.

Sentences are, by construction, more significant from a semantic point of view than words. However, IR models, as well as ML algorithms, need access to words to operate. It becomes fundamental to apply preprocessing steps to obtain indices, derived from words, that are significant and without semantic ambiguities. After performing the classic case folding, we analyze word frequencies to understand their importance. In particular, we mention two laws:

- Zipf's law, first expressed in 1949 by Zipf et al. in the book "Human Behaviour and the Principle of Least Effort" [3], is an empirical law that describes the distribution of word frequencies in a natural text. It is based on the observation that few words in a text are extremely common, while most words are very rare. To be more precise, the frequency of a word is inversely proportional to its rank in the list of most frequent words.

- From Zipf's law derives Luhn's analysis, described in 1958 by Luhn in the "IBM Journal of Research and Development" [4]. It is a technique aimed at identifying and extracting relevant information from large volumes of text or data. The main idea is as follows: the frequency with which some words appear in a text provides an important indication of their meaning. Additionally, their position in the sentence is another important parameter for semantic interpretation.

In summary, rare words are discardable because they are "rare," while common words are not informative. The insights from Luhn's analysis are utilized in this paper at various levels. For example, for basic retrieval models, we apply stop word removal on the most frequent words, but more generally, we employ models that incorporate considerations from this analysis in term weighting (e.g., BM25).
Entering into specifics, we can highlight the construction of two indices for the models in this report:

- First index. Built for basic retrieval models, it leverages PyTerrier's default imports (TerrierTokenizer, TerrierStemmer.porter, TerrierStopWords).

- Second index. Built for neural reranking models, it does not perform the stopword removal compared to the previous one to allow such models to understand the context of terms.

From the list just provided, we note that a preprocessing technique is included in the construction of both indices: stemming. Stemming is a technique that operates at a morphological level, bringing words back to their roots. It is less effective than lemmatization, which brings wordforms to the base form, but not so much as to disregard the computational aspect required. Stemming improves recall at the expense of precision and also reduces the size of the dictionary. This last characteristic guided the choice of stemming in this study: the size of the document corpus amounts to 5.4 million, a reduction in the dictionary size leads to lower computational and memory requirements. Additionally, as cited in Buckley et al. (1984) [5], stemming is generally effective in information retrieval tasks. The stemmer used is Porter, which has been experimentally proven to be at least as good as all other stemmers in circulation.

Entering into the detail of not removing stop words for neural models, we can highlight that their removal would alter the context of the words, not allowing to leverage it for reviews. When using a contextual model like BERT, in particular, context becomes central, enough to allow a different representation of polysemous words depending on it. Even though we did not remove them, it is interesting to mention a study by Quiao et al. from 2019 [6] that might raise the need for further investigation. Surprisingly, contrary to what was just highlighted and common belief, removing stopwords in transformer-based models did not affect their performance.

At this point, we report in Figure 4 the word cloud of the indexed terms of the first built index and in 5 the word cloud of the second index.

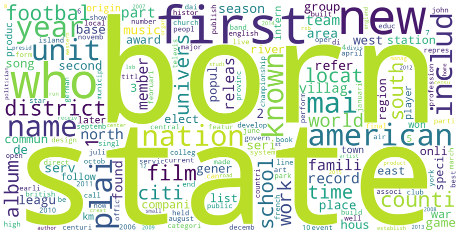


Figure 4: Index 1 - Basic Models

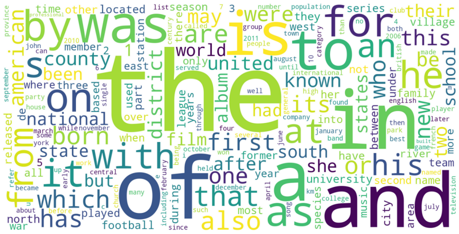


Figure 5: Index 2 - Neural Models

Considering that the collection is made up of 5.4 million documents, the first index contains 2,471,240 terms, 203,556,545 postings, and 269,889,695 tokens. In the second index, there are 2,471,392 terms, 284,943,850 postings, and 453,148,161 tokens. The second index contains more terms, stopwords, and related posting lists.

In this initial part, Basic Search Models are introduced, employing various pipelines to discern the most effective model performance.

- **TF-IDF (Term Frequency-Inverse Document Frequency)** is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents. Effective for small to medium-sized document collections. Does not consider term saturation or document length normalization, which can lead to skewed relevance scores for longer documents.

- **BM25** is an extension of the TF-IDF model and addresses some of its limitations, such as term saturation and document length normalization. BM25 is scalable and performs well on larger collections of documents compared to TF-IDF.

- **TUNED BM25** provides better tuning options, allowing for more precise control over the retrieval process. Solutions to mitigate the saturation problem include parameter tuning, such as adjusting the k1 and b parameters, or considering alternative weighting schemes. In this step the parameters are tuned through GridSearch, in particular b controls the amount of correction for the document lenght (0 no correction and 1 maximum correction), k1 controls the saturation and k3 (optional) scale the IDF. Cosidering the amount of necessary resources for the tuning it was runned on Kaggle. The result are not avaiable in the notebook, but the best configuration is with b=0.5, k1=0.3 and k3=20.

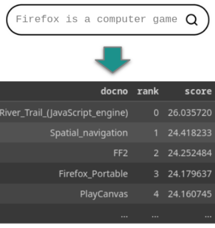
- **Query Expansion - B01- Query Expansion (QE)** is a method used for improving Information Retrieval (IR) by adding the terms that are almost selected from feedback documents, and similar to the user query terms, in order to improve the effectiveness of the search ranking. Bo1 (Bag of One) divergence from Randomness query expansion model to rewrite the query based on the occurrences of terms in the feedback documents provided for each query.The pipeline is mad of BM25-Tuned — B01 Query Expansion — BM25-Tuned.

The obtained results are as follows:

Table 2: Retrieval Model comparison.

		TF-IDF	**BM25**	**TUNED BM25**	**TUNED BM25 + BO1**
SIMPLE MODELS	P@5	0.11	0.11	0.14	0.14
	P@10	0.06	0.07	0.08	0.07
	R@5	0.50	0.49	0.63	0.64
	R@10	0.60	0.60	0.71	0.72
	MAP	0.37	0.37	0.50	0.50
	NDCG	0.47	0.46	0.60	0.60
	MRR	0.39	0.39	0.53	0.52

The results obtained allow us to notice that in the case of basic models, without using a neural approach, the model that achieves the best performance is BM25 (after tuning operations) and the BM25 and B01 pipeline. These models achieve good results, with a MAP (Mean Average Precision) equal to 0.50. It is important to note high recall values, indicating a good ability of the model to retrieve relevant documents. The low precision values indicate low relevance among the retrieved documents. This emphasizes the importance of considering both metrics. The obtained values of NDCG (Normalized Discounted Cumulative Gain) allow us to take into account the relevance of documents based on their position, considering the higher utility of documents in the first positions. In the analyzed case, the NDCG value is acceptable. The MRR (Mean Reciprocal Rank) allows us to conclude that in most cases, the first relevant document is found in the first positions. A further visual analysis of the results obtained (repeated in the slides) allows us to see that in all models, the first documents are relevant to the proposed query and satisfy the user's informational needs, as showed in the Figure 6.



Firefox is a computer game

	qid	docid	docno	rank	score	query
0	1	4212581	River_Trail_(JavaScript_engine)	0	26.035720	Firefox is a computer game
1	1	4536369	Spatial_navigation	1	24.418233	Firefox is a computer game
2	1	1696167	FF2	2	24.252484	Firefox is a computer game
3	1	1661512	Firefox_Portable	3	24.179637	Firefox is a computer game
4	1	3816930	PlayCanvas	4	24.160745	Firefox is a computer game
...	...	...	...	...	...	...

Figure 6: BM25 Search Engine Result

4 Search Engine - Advanced Search (Neural Models)

Considering the results obtained from the first proposed models, certain neural models were introduced with the intention of improving the effectiveness of the previous models. However, when these introduced models are used without pretraining, their performance tends to be suboptimal. In this scenario, the index for neural models is employed, which is built while considering stopwords as well.

The proposed neural models are:

- KNRM - This re-ranker provided by OpenNIR, consist of ranker which define the neural architecture and the vocab in this case 'wordvec_hash' which define how text are encoded in the model. The model is not trained, so KNRM will use random weights to combine scores. It's designed to improve the relevance of search results by re-ranking them based on a neural network model that captures semantic similarities between queries and documents. The applied piepline is based on: BM25-TUNED % 50 — Text Extraction & None cleaning — KNRM. The re-ranking is applied on the first 50 retrived documents, considering the limited resources.

The obtained results can be summarized in this table:

Table 3: Neural Models.

		KNRM	**BERT- MSMARCO**	**SCI-BERT**	**BI-ENCODER**
NURAL MODELS	P@5	0.006	0	0	0
	P@10	0.006	0	0	0
	R@5	0.03	0	0	0
	R@10	0.06	0	0	0
	MAP	0.04	0	0	0
	NDCG	0.17	0	0	0
	MRR	0.04	0	0	0

The obtained results suggest that the KNRM re-ranker does not improve performance, potentially due to the absence of training of the model itself. Thus far, the best-performing model appears to be the tuned version of BM25.

Considering another project of Fact Detection, we implemented a neural model, such as BERT fine-tuned to achieve 96% accuracy in classifying certain facts. It's important to note that Fact Detection and Fact Retrieval represent distinct tasks within the realms of information retrieval and natural language processing, each possessing unique objectives and methodologies. Fact Detection involves the assessment of the truthfulness of specific claims or statements, whereas Fact Retrieval focuses on retrieving relevant factual information from extensive collections of documents or knowledge bases.

The integration of both approaches could potentially enhance the effectiveness of fact-checking. This integration could be a feature development of the project, bridging the gap between identifying the veracity of claims and accessing relevant factual information. By combining Fact Detection and Fact Retrieval methodologies, we may create a more comprehensive and robust system for fact-checking.

In conclusion, the proposed project permits to analyze one of the most important problems in our digital society today: fact-checking. Thanks to the development of search engines, it is possible to collect correct information about general facts in our world. In particular, it is important to train each Information Retrieval Model on reliable sources that provide correct and trustworthy results.

References

[1] Eurostat, "How many people verified online information in 2021?" December 2021. [Online]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20211216-3

[2] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 809–819. [Online]. Available: https://www.aclweb.org/anthology/N18-1074

[3] G. K. Zipf, *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.

[4] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[5] C. Buckley and G. Salton, "Efficiency of stemming for information retrieval: An experimental study," *ACM Transactions on Information Systems (TOIS)*, vol. 2, no. 2, pp. 107–106, 1984.

[6] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of bert in ranking," 04 2019.