TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 6592 - 6599

Exploring the use of Machine Learning for Highly Accurate Text-based Information Retrieval System

Chandrashekhar Himmatrao Sawarkar[1]* Dr. Pramod N. Mulkalwar [2]

1Assistant Professor, Smt. Narsamma Arts, Commerce and Science College, Amravati,Maharashtra, India
e-mail: chsawarkar@gmail.com
[2]Associate Professor, Amolakchand Mahavidyalaya, Yavatmal, Maharashtra, India
e-mail: pnm_amv@rediffmail.com

Article Info
Volume 81
Page Number: 6592 - 6599
Publication Issue:
November-December 2019

Article History
Article Received:*5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *31 December 2019*

Abstract:
Information retrieval has been a pillar-stone for today's digital age, wherein every online entity expects correct and fast information. More than 80% of all searches made on search engines are text-based and thus having an accurate text-based information retrieval system is a must for today's corporations. Text-based retrieval systems range from simple query processing, to complex elastic search-based systems. The decision of algorithm selection for retrieval systems depends on the application for which the system is designed. While systems like chat-bots require highly-complex machine learning-based retrieval systems, some systems like intranet-based searches give high accuracy with simple query-processing. In this work, we propose the design of a machine learning-based hybrid information retrieval system which adapts itself as per the application, and provides solutions that result in highly accurate information retrieval. We tested the proposed algorithm under different datasets, and found it to be accurate with lesser response time as compared to some of the state-of-the-art systems.

Keywords: *Information, retrieval, machine, learning, adaptive, text.*

I. INTRODUCTION

An efficient information retrieval system is able to crawl data from different sources in order to provide query-relevant results with utmost accuracy and minimum delay. Information retrieval systems have matured in the past decade, from applications ranging from simple intranet-based searches, to complex auto-response systems. The selection of algorithms for clustering the data, feature extraction from the data, and classification of the data define the performance of these systems. Any information retrieval system can be designed with the help of the following steps,

- Data collection

Any retrieval system requires a huge-collection of application-specific datasets in order to effectively retrieve results. For example, bio-medical text-based retrieval systems require collection of bio-medical research papers, case-studies of patients, patient reports, etc. in order to effectively produce these results whenever a relevant query is given by the user. This step defines the depth of information that can be provided by the information retrieval. An initial optimum size estimation for the dataset is a must, and can be increased or reduced as per the performance of the retrieval engine.

- Data pre-processing

The input dataset is per-processed via filtering approaches. This filtering can be used to reduce the redundancy in data, fill-in missing values in the data, or any application specific process that is needed to convert the data into processable format by the later stages.

- Indexing and data-storage

Generally pre-processed data is given to an indexing engine, where single, double or multiple-indexing is done. This indexing speeds up the process of retrieval, and adds to the accuracy of retrieval. Efficient storage of the indexed data decides the read access time of the data, and thus must be done with effective approaches like graph-based storage.

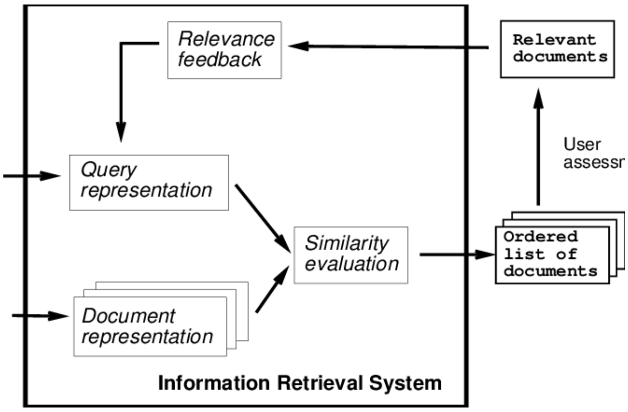

Figure 1. A typical document recommender system

- Query processing

The user provides an input query, which is processed with the help of language processing techniques like parts-of-speech tagging, chunking, lemmatization, synsets, antonym-sets, etc. The query processing engine deciphers the sense of the input query, and provides it for relevance matching. Query processing can also use ontology-based approaches for a better retrieval performance.

- Relevance matching

The processed query is given to a relevance matching engine, wherein the score of the query is calculated w.r.t. the indexed datasets which are stored in the system. The relevance matching can be done using techniques like term-frequency (TF), inverse document frequency (IDF), singular value decomposition (SVD), principal component analysis (PCA), etc. The selection of an effective relevance matcher decides the final retrieval accuracy of the system.

- Relevance ranking

Once the relevance processor provides scores for the input query, the ranking engine assists in arranging the positively scored documents in an order which is most useful for the user. The ranking engine can also take into consideration the user-feedback and re-score the documents. This re-scoring will change the document's rank at the output, and might add the accuracy of retrieval.

- User-feedback for improving relevancy

If the user is not satisfied by the ranked results, then the user-feedback engine asks the users to provide feedback as-to which results are most suited as per the user's requirements. Based on this feedback, the system is able to learn and re-tune it's internal scoring and ranking processing order to improve result relevancy. Algorithms like q-learning and incentive-based learning can be used for this purpose.

- Post-processing

Results obtained from the input query are sometimes post-processed in order to extract information from them. For example, if the input query is "what is the temperature today?", then apart from retrieving the data for today's temperature, the post-processing engine will also find out similarly-relevant parameters like precipitation, humidity, air direction, etc. This helps the users to get a better experience while searching.

Using these steps many systems have been proposed by researchers over-the-years. In the next section, some of the recent techniques are reviewed, followed by the design of the proposed technique. Finally, the result analysis of the given technique is performed, and some interesting observations about the results are mentioned. We conclude this text with some recommended gaps on which researchers can work-on to further improve this research.

II. LITERATURE SURVEY

To recover a positioned rundown of significant outcomes, IR specialists create recovery models and assess their adequacy (as depicted underneath). A recovery model speaks to the coordinating procedure of question and record. It creates a positioning of reports that match the inquiry. In early IR frameworks, this coordinating procedure was set-based, utilizing boolean questions that enabled a client to express consistent provisions for coordinating. For instance, a question could be defined to coordinate data AND recovery OR looking for. An early augmentation to the boolean model was to enable clients to weight inquiry terms [1]. Along these lines, the recovery model positions records by their importance to the inquiry, beginning with the most applicable report. A formalization of this is known as the vector space model [2]. In this model, the question and records are spoken to as vectors in a space, where each term is a measurement. Coordinating is finished by looking at the separation among record and question in this space, e.g., by estimating cosine comparability. Allotting a load to each term in this vector space has the impact of gauging the term in the archive positioning. Usually utilized weighting plans utilize the recurrence of a term in the report (term recurrence or TF) joined with the quantity of records that contain the term (archive recurrence or DF). The instinct behind the previous is that if a question term happens as often as possible in an archive, that record is probably going to all the more likely match the inquiry. The instinct behind the last is that words that happen in numerous records are probably going to be extremely conventional and along these lines less helpful for separating among significant and non-important archives. The idea of report positioning was additionally formalized in [3] as the likelihood positioning standard, which expresses that an ideal positioning is one that positions archives in diminishing request of their likelihood of pertinence to the inquiry. The two most conspicuous recovery models that have pursued from this guideline are: BM25 and the language displaying structure. The BM25 recovery model [4] heuristically consolidates the TF and DF insights portrayed above and utilizes record length standardization. In the language demonstrating system [5] reports are displayed as a sack of words drawn from a measurable language appropriation. Registering a score for a report comes down to processing the likelihood that both the question and the record where drawn from a similar dispersion. An elective way to deal with positioning archives is to apply AI in an alleged figuring out how to rank setting [6]. Here, positioning capacities, for example, BM25 and insights like as TF and DF are utilized as highlights in an AI way to deal with ideally rank records. A record assortment with significance appraisals for each question is utilized to get familiar with a model that ideally consolidates these highlights into a solitary positioning score. By and by, a wide range of sorts of highlights are considered, including for instance: record length, comprehensibility, or highlights got from interface structure in pages. A notable case of this last kind is Pagerank [7], a calculation that iteratively registers the impact of pages dependent on the pages that connect to it. In a web-based setting, positioning is improved legitimately from client criticism [8] rather than from comments in the disconnected setting. This web-based setting maps legitimately to comes nearer from reinforcement machine learning (RL) [9]. RL interlaces the ideas of ideal control and learning by experimentation. Focal is the idea of a "specialist" advancing its activities by connecting with nature. This is accomplished by learning an approach that maps states to activities. In displaying the IR setting, the work in [10] maps the recovery framework to job of operator, making the move of recovering archives given the condition of a question based on an approach. To formalize the assignment of discovering video content identified with a live transmission analyst model it as a Markov choice procedure (MCP) [11]. An MCP is a particular sort of fortification learning issue that was proposed before the field was known as support learning. In a MCP, we settle on the ideal activity in a Markov procedure [12]. The Markov property holds when the approach for a state is free on the past states. A Markov state hence needs to speak to the whole history of past states the extent that this is significant for the estimation of the arrangement. MDPs have been utilized to demonstrate different IR issues, for instance, Work in [13] use support learning and MDPs to improve positioning over different query item pages. In [14] decline loads of new terms dependent on past remunerations, while in [15] model session search as a double operator stochastic game. Researching the structure decisions for MDPs, they locate that express input and specifically empowering or debilitating explicit recovery innovation are best in session search [16]. In our spilling setting, new data continues coming in, subsequently we are managing a non-stationary MDP. All the more explicitly, in light of the fact that the choice of what activity to pick doesn't impact the states that rise up out of nature, this is viewed as an acquainted hunt task. Past record recovery. The majority of what we have depicted above identifies with the center IR undertaking of specially appointed hunt, where a client represents any conceivable question and an IR framework reacts with a positioned rundown of records. Numerous other IR undertakings exists. We spread two that are especially important to this theory: content characterization and archive separating. In a book arrangement task, the point is to relegate a specific name to a record, or saw in an unexpected way, to allot an archive to a specific class or classification

[17]. The objective of this errand is to compose an assortment of records and permit better understanding and elucidation of the assortment. An early case of content grouping is bookkeepers appointing topical names to books (returning to the primary library around 300 BC). One of the most unmistakable ebb and flow models is that of spam location, e.g., in email or in web search [18]. The calculations for this undertaking intently look like those utilized a recovery setting. We utilize a methodology like the figuring out how to rank methodology portrayed above, in which we use highlights got from literary substance to get familiar with a model that doles out marks as precisely as could reasonably be expected. Another normal IR task is record separating. In a regular hunt task the inquiry changes and the assortment stays static. In a run of the mill archive sifting task, a standing question is utilized to channel a surge of records [19]. No positioning of reports is important. Different instances of such assignments incorporate condensing online networking continuously [20] and discovering replications of news stories while they show up. Fleeting IR. Transient IR [21] manages demonstrating fleeting examples to improve data recovery and spreads subjects, for example, report freshness and worldly significance. Work in [22] survey ebb and flow inquire about patterns in worldly IR and recognize open issues and applications. Open issues incorporate how to figure fleeting similitude, how to consolidate scores for printed and transient questions, and introducing worldly data in an intelligent setting. Work in [21] utilizes straight time arrangement models for weighting terms, where the time arrangement is registered on the objective report assortments. Also, work in [19] utilize transient rushes in a microblog assortment to reweight question terms for improved recovery. Our work contrasts in that we model transient elements in the surge of captions from which we create questions. Thereby, we have utilized machine learning to build a highly effective retrieval model. The details of the same are given in the next section, followed by its performance analysis

III. The proposed machine learning-based information retrieval model

The proposed machine learning algorithm used to perform information retrieval, works in the following phases,

- Pre-execution or intensive learning phase
- Execution or incremental learning phase or AI phase

The pre-execution or intensive learning phase works in the following steps,

i. Initialize the learning parameters, such as,

Number of learning rounds = Nr

Number of learning solutions = Ns

Learning rate = Lr

Max number of features per solution = Fmax

Max algorithms per solution = Amax

Max models per solution = Mmax

ii. For each round, for each solution which has to be changed in this round, perform the following to find a new solution,

   a. Select random features from the input document. Make sure that the number of features is exactly Fmax

   b. Select the Mmax models from the set of models available for processing

   c. Evaluate the Fmax features using the Mmax models for the input query

   d. Select Amax number of random algorithms, from the following list of algorithms,

      i. Query expansion algorithm

      ii. Pragmatic algorithm based on NLP

      iii. Interconnectivity algorithm based on NLP

   e. Apply algorithm learning for all the Amax classifiers using these Fmax features on each of the Mmax models

f. Evaluate the accuracy of the classifier system, and mark it as the learning convergence for this solution, using the following formula,

$$Lc = \frac{\sum_{i=1}^{Imax} Ai/Ndi}{Imax} \dots (1)$$

$$Ai = Accuracy\ for\ i^{th}\ model$$

$$Ndi = Normalized\ Delay\ needed\ for$$

$$processing\ the\ i^{th}\ model$$

g. The normalized delay is evaluated using the following formula,

$$Ndi = \frac{di}{\sum di} \dots (2)$$

$$where, di = delay\ needed\ to\ process\ the\ i^{th}\ model$$

h. Observe this solution, and keep it for ready reference

iii. Evaluate the learning convergence for each of the solutions, and then evaluate the learning threshold

$$Lth = \frac{\sum_{i=1}^{Ns} LCi}{Ns} * Lr \dots (3)$$

$$where, Lr\ is\ the\ learning\ rate$$

iv. For each solution which satisfies equation 4, pass it onto the next round, else, discard the solution and replace it with the help of step (ii)

$$LCi > Lth \dots (4)$$

v. Repeat steps (ii) to (iv) for Nr rounds, and prepare the following table at the end of the Nr round,

Sol. Num	Sel. feats.	Sel. Models	Sel. algos	LC val	Accuracy

Table 1. The intensive learning-table

vi. From the table 1, select the solution with highest value of LC and highest value of accuracy and use it for the execution phase.

Due to the intensive learning phase, we get a large number of solutions, which are kept for further evaluation in the actual execution phase. The following steps are performed in the actual execution phase,

i. Select the best accuracy entry from the learning table 1

ii. For each of the input text, apply the feature selection as mentioned in the 2nd column

iii. Apply the classifiers as mentioned in the 3rd and 4th column of table 1, and evaluate the output information retrieval

iv. Inject random training set entries for evaluation, and repeat steps (i) to (iii) for these random entries

v. Evaluate the accuracy for these random entries, and evaluate the value of Lc for each of these entry sets

vi. If the value of Lc for a given set is lower than the one selected from table 1, then update table 1 with this value

vii. Select the next best entry from table 1, and repeat the process for each of the queries

viii. In case more than half of the entries of table 1 are replaced, then retrain the algorithm with the help of the pre-execution step, and re-create table 1 with better entries of Lc

Due to the continuous learning process which takes place in this algorithm, the overall system's accuracy improves, and we get a better information retrieval accuracy than any of the individual algorithms. The proposed technique works exceptionally well when compared to the existing state-of-the art methods, this comparison is done in the next section.

IV. Result and Analysis

In order to evaluate the performance of the proposed algorithm, we compared the obtained results with the following algorithms,

- Vector space
- Probabilistic
- Boolean-model

The results were evaluated for the common products dataset, which was taken from the Shopify website. Shopify is a subsidiary of Facebook, and helps common people to setup web-stores with minimum effort. People provide data about their products, and that data is stored inside the products file in a Java Simple Object Notation (JSON) format. The collected dataset contains 100k records, and these records are processed in real-time using indexing technique. The obtained data is then given to an indexer for indexing and obtaining the results in a dictionary format. These results are then processed through the given algorithms, and results like precision, recall, f-measure and accuracy are evaluated.

Accuracy = (TP+TN)/(TP+FP+FN+TN)

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

F1 Score = 2(Recall * Precision) / (Recall + Precision)*

where, TP = Total number of results which must be retrieved and are present at the output

TN = Total number of results which must not be retrieved and are not present at the output

FP = Total number of results which must not be retrieved and are present at the output

FN = Total number of results which must be retrieved and are not present at the output

Using these equations, we evaluated the parameters and obtained the following results for precision of the algorithms,

Query Length (in words)	Boolean-model	Probabilistic model	Vector space-model	Proposed model
1	97.8	98.5	98.9	99.1
2	97.82	98.54	98.93	99.16
3	97.83	98.56	98.96	99.2
5	97.85	98.59	98.99	99.25
6	97.86	98.62	99.02	99.3
8	97.88	98.65	99.05	99.35
10	97.89	98.68	99.08	99.4
15	97.91	98.71	99.11	99.45
20	97.92	98.74	99.14	99.5

Table 1. Results for precision

Similarly, the results for recall can be presented via the following table,

Query Length (in words)	Boolean-model	Probabilistic model	Vector space-model	Proposed model
1	84.6	88.23	92.5	96.5
2	84.62	88.33	92.54	96.6
3	84.63	88.38	92.6	96.75
5	84.65	88.4	92.7	96.82
6	84.67	88.48	92.75	96.95
8	84.68	88.53	92.82	97.06
10	84.7	88.59	92.88	97.17
15	84.71	88.64	92.95	97.28
20	84.73	88.7	93.01	97.39

Table 2. Results for recall of different algorithms

The accuracy was also evaluated, and the following results shown in table 3 were obtained. We observe that the proposed machine learning based algorithm optimizes the overall performance of the information retrieval process. It does this by finding all possible combinations of algorithms which can be applied to the system in order to obtain the best retrieval output. From the results we can observe that the proposed algorithm improves the precision by more than 5%, while the recall rate is improved by more than 10%, while the accuracy is improved by more than 2%.

Query Length (in words)	Boolean-model	Probabilistic model	Vector space-model	Proposed model
1	98.13	98.73	99.06	99.29
2	98.14	98.73	99.07	99.31
3	98.15	98.74	99.09	99.35
5	98.15	98.75	99.1	99.38
6	98.16	98.76	99.12	99.41
8	98.17	98.76	99.13	99.44
10	98.18	98.77	99.15	99.47
15	98.18	98.77	99.16	99.5
20	98.19	98.78	99.18	99.53

Table 3. Comparison of algorithmic accuracy for the system

While 2% might seem a small improvement, but consider the case for 100k records. A 2% improvement in accuracy results in 2000 records being correctly processed, which makes a big difference when it comes to real-time information retrieval process.

V. CONCLUSION AND FUTURE WORK

The proposed machine learning algorithm combines the advantages of most of the previously used state-of-the-art information retrieval systems like vector space model, Boolean model and probabilistic model, etc. These models combined with different classifiers help the proposed algorithm to achieve more than 99% accuracy in terms of retrieval, this means that the user is able to obtain data records which almost completely match the input query, thereby improving the overall user experience while using the system. Moreover, the performance in terms of precision and recall values is also very inspiring, and it makes the proposed system capable for usage in all kinds of real-time information retrieval scenarios. Due to the incorporation of machine learning, the overall system QoS is improved. It can further be optimized with the help of more advanced computation techniques like deep-learning and incentive-based learning. These techniques when combined with the existing machine learning algorithm will further boost the QoS and help is making a better recommendation system.

VI. REFERENCES

[1] M. Koubarakis, S. Skiadopoulos and C. Tryfonopoulos, "Logic and Computational Complexity for Boolean Information Retrieval," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1659-1666, Dec. 2006.

[2] P. Castells, M. Fernandez and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 2, pp. 261-272, Feb. 2007.

[3] H. Yang, W. Shin and J. Lee, "Private Information Retrieval for Secure Distributed Storage Systems," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2953-2964, Dec. 2018.

[4] Q. Chen, Q. Hu, J. X. Huang and L. He, "TAKer: Fine-Grained Time-Aware Microblog Search with Kernel Density Estimation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1602-1615, 1 Aug. 2018.

[5] Mathias Ellmann. 2018. Natural language processing (NLP) applied on issue trackers. In Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering (NL4SE 2018). Association for Computing Machinery, New York, NY, USA, 38–41. DOI:https://doi.org/10.1145/3283812.3283825

[6] J. Liu *et al.*, "Artificial Intelligence in the 21st Century," in *IEEE Access*, vol. 6, pp. 34403-34421, 2018.

[7] Z. Zhu, Q. Peng, Z. Li, X. Guan and O. Muhammad, "Fast PageRank Computation Based on Network Decomposition and DAG Structure," in *IEEE Access*, vol. 6, pp. 41760-41770, 2018.

[8] S. C. H. Hoi, R. Jin and M. R. Lyu, "Batch Mode Active Learning with Applications to Text Categorization and Image Retrieval," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1233-1248, Sept. 2009.

[9] G. Ioannakis, A. Koutsoudis, I. Pratikakis and C. Chamzas, "RETRIEVAL—An Online Performance Evaluation Tool for Information Retrieval Methods," in *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 119-127, Jan. 2018.

[10]Y. Pan, H. Lee and L. Lee, "Interactive Spoken Document Retrieval With Suggested Key Terms Ranked by a Markov Decision Process," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 632-645, Feb. 2012.

[11] Wolfgang Büschel, Annett Mitschick, and Raimund Dachselt. 2018. Here and Now: Reality-Based Information Retrieval: Perspective Paper. In Proceedings of the 2018 Conference on Human Information Interaction & Retrieval (CHIIR '18). Association for Computing Machinery, New York, NY, USA, 171–180. DOI:https://doi.org/10.1145/3176349.3176384

[12]P. A. Mishra and B. Roy, "A novel technique for inferring user search using feedback sessions," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 856-860.

[13]Onal, K.D., Zhang, Y., Altingovde, I.S. *et al.* Neural information retrieval: at the end of the early years. *Inf Retrieval J* **21,** 111–182 (2018) doi:10.1007/s10791-017-9321-y

[14]N. T. Wai Khin and N. N. Yee, "Query Classification based Information Retrieval System," *2018 International Conference on*

Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, 2018, pp. 151-156.

[15]N. T. Alhindawi, "Information Retrieval - Based Solution for Software Requirements Classification and Mapping," *2018 5th International Conference on Mathematics and Computers in Sciences and Industry (MCSI)*, Corfu, Greece, 2018, pp. 147-154.

[16] Sarvar Patel, Giuseppe Persiano, and Kevin Yeo. 2018. Private Stateful Information Retrieval. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS ’18). Association for Computing Machinery, New York, NY, USA, 1002–1019. DOI:https://doi.org/10.1145/3243734.3243821

[17]J. Luo and X. Xue, "Research on Information Retrieval System Based on Semantic Web and Multi-Agent," *2010 International Conference on Intelligent Computing and Cognitive Informatics*, Kuala Lumpur, 2010, pp. 207-209.

[18]Xiao Qiuhui, "Design of Web-based teaching system for information retrieval," *2009 IEEE International Symposium on IT in Medicine & Education*, Jinan, 2009, pp. 462-465.

[19]A. Gudivada and N. Tabrizi, "A Literature Review on Machine Learning Based Medical Information Retrieval Systems," *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Bangalore, India, 2018, pp. 250-257.

[20]C. S. S. Kumar, M. Mohanapriya and C. Kalaiarasan, "A new approach for information retrieval in semantic web mining involving weighted relationship," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, 2017, pp. 1-4.

[21]Srinaganya.G. and J. G. R. Sathiaseelan, "A technical study on Information Retrieval using web mining techniques," *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, 2015, pp. 1-5.