ISSN No:-2456-2165

A Detailed Comparative Analysis of Document Ranking Approaches

Hashmy Hassan
Assistant Professor, Department of Statistics, CUSAT
Research Scholar, Division of CSE, SOE, CUSAT
Kerala, India

Sudheep Elayidom
Professor, Division of CSE, SOE,
CUSAT
Kerala, India

Abstract:- This paper is a detailed comparative analysis of different document ranking algorithms, focusing on retrieval systems to generate a relevance score for each document on the user query. We are trying to bring out knowledge in document ranking through this analysis. In this paper, we divided the document ranking algorithms from pieces of literature. We categorized them into three main categories based on the algorithm used – a score based, machine learning and neural ranking approach.

Keywords:- *Information Retrieval, Document Ranking, Learning to Rank, Neural Ranking.*

I. INTRODUCTION

In the last decade, rapid growth in unstructured data sources has led to the advent of popular NoSQL databases. Searching these data has been identified as a challenge concerning scale and format. These unstructured data include documents in the image, scanned, printed, handwritten and electronic document exchange formats like jpeg, png, pdf etc. Various OCR tools are available to automatically convert these documents to a machine-understandable text format. But still, searching for a particular document upon a query has been a problem to address with mere digitalization of historical data. Electronically representing these digitalized data will fasten many operations and decision making, bringing transparency and ease in managing and reliability of the source.

Search is a crucial function used in any application. Searching can be on anything from the position of a keyword to searching for documents. The limited usage of exact keyword search has driven the emergence of semantic search and thereby bought a new horizon in many applications. Modern applications rely heavily on search, the quickest way to sift through massive data for important information.

A fundamental problem with document searching is ranking, which is significant in document retrieval and recommender systems. A document ranker ranks documents according to their relevance for a given query.

The document ranking approaches can be broadly divided into three categories. Section II explains these three models and algorithmic methods.

II. DOCUMENT RANKING

The feature vectors of each documentD_i have to be prepared before querying the documents to retrieve top k documents from candidate documents D that matches the query q. The q can contain multiple terms. Document ranking can be defined as identifying top k documents that best match the q from the D.

The document ranking approaches can be broadly classified into three – Score based models, machine learning-based modes and neural ranking models.

A. Score Based Models

In this approach, documents are ranked by combining independent Scores calculated for each query- the query term independence assumption. Like the Boolean models, the early models used the number of occurrences of the query terms in the documents to identify their relevance. Still, they failed to rank the documents based on relevance.

Score-based models use classic IR models like Vector Space Models(VSM) and Probabilistic models. The traditional Information Retrieval systems, such as query likelihood[1] , and BM25[2] are based on exact keyword match of document and query words. The above is built various smoothing, normalization and weighting techniques.

Score based models used the term frequencies (TF) to term importance in a document. The VSM ranked the document based on the (Term frequency-inverse Document Frequency) TF-IDF score of queries and documents represented in vector space. Whereas the probabilistic ranking models like BM25, the graph-based approach is built on top of VSM by ranking the documents by log odds of their relevance. TextRank[3] is an alternative to TF is used mainly for web search applications. The documents needed to be assessed per query calculated using various methods like *inverted index*[4] and other organization strategies like *impact ordering* [5].

Lucene is a full-featured text search engine. Lucene ranking function, which is based on a mix of the Vector Space Model (VSM) and the Boolean model of information retrieval, used to assess how relevant a document is to a given query . The primary premise underlying the Lucene technique is that the more times a query term appears in a document in comparison to how many times it appears in the entire

collection, the more relevant that document is to the query. Lucene additionally uses the Boolean model to filter down the pages that need to be scored depending on the query specification's use of boolean logic.

Some of the other tools, like [6] are built using the VSM, BM25 and IR Language models.

B. Machine Learning-Based Models

The document ranking problem has been solved using ML technologies to predict the ranking of documents using the "learning to rank" methods. Learning to rank is a machine learning approach to solve the document ranking problem. The training set will contain the set of documents with their relevance against each query. For a given query and set of documents, the model will give each document a score, and ordering the score of documents in ascending order can be used to obtain the rank of documents. Three significant methods in learning to rank are *pointwise, pairwise and listwise* methods.

➢ *Pointwise approach*

The pointwise method is a straightforward approach of using existing machine learning approaches to build models for document ranking. The three sub-categories of the pointwise approach include regression-based algorithms, classification-based algorithms, and ordinal regression-based algorithms.

- *Regression-based algorithm*

This approach is used when the output of the training model is a real-valued relevance score of documents against the query. Polynomial Regression Function and Subset Ranking with regression are two powerful algorithms using this approach. The polynomial Regression method is used on the least square regression method to learn the scoring function.

Subset ranking with the regression method is an approach to solving ranking by reducing it to a regression problem. Where, for a group of documents associated with a given query, the original truth labels of these documents are categorized in a multiple ordered category where a function is used to rank these documents. [7]

- *Classification based algorithms*

Here the document ranking is considered a classification problem. The discriminative IR model and Multi-class Classification (McRank) are two ranking algorithms using the classification method. The discriminative IR and discriminative classification models are used for relevance ranking. In Machine Learning works of literature, discriminative methods are widely used to combine different kinds of features without the necessity of defining a probabilistic framework to represent the objects and the correctness of prediction. Some of the works using this approach using the classification methods like Maximum Entropy and Support Vector Machine are explained in [8]–[10].

McRank explained in[11] is based on the Discounted Cumulative Gain(DCG), where a perfect classifications result in perfect DCG scores, and the DCG errors are bounded by classification errors.

- *Ordinal Regression-based Algorithms*

Ordinal regression takes the ordinal relationship among the ground truth labels in the data to learn for document ranking. [12]–[16] is based on this approach.

➢ *Pairwise approach*

In this approach, the relative order between pairs of documents. The goal of learning is to maximize the number of correctly ordered document pairs. Here, the ranking problem is transformed into a task of pairwise classification, with an assumption that the ranking of documents can be achieved if all the pairs of documents are correctly ordered.

The input space of the pairwise approach contains a couple of documents, both represented as feature vectors. The output space includes pairwise preferences from {1, -1} between each pair of documents. However, the loss function merely considers the relative order between two documents rather than the total order relationship among all the documents associated with the same query. In this regard, the adopted loss functions are not per the evaluation measures.

The number of document pairs per query may differ from query to query; thus, the result can be biased in favour of queries with more documents in the training data[17]. Some of the algorithms using pairwise approaches are [18]–[25]

➢ *Listwise approach*

The listwise techniques use all of the documents in the training set that are connected with the same query as input. When performing listwise learning, there are two sorts of loss functions to consider. The loss function for the first type is linked to a specific evaluation metric(nDCG [26], ERR[27]). So because typically employed metrics are non-differentiable and non-decomposable, these strategies either strive to optimize the upper bounds as surrogate objective functions.[28]–[30] or approximate the target metric using some smooth functions [31]–[33]. However, several difficulties with the initial type methods remain unresolved.

On the one hand, some surrogate functions or approximated metrics are not convex, making optimization difficult. But the relationship between the surrogate function and the adopted metric has not been sufficiently investigated in most ranking algorithms, making it unclear whether optimizing the surrogate functions can optimize the target metric. The loss function in the second kind is not explicitly linked to a specific evaluation metric. The difference between the anticipated and ground-truth rankings is reflected in the loss function (E.g. [34]–[36]. Despite the fact that no specific assessment criteria are explicitly involved or optimized in this study, it is feasible that the learnt ranking function will perform well in terms of evaluation metrics.

In [37], the query-level ranking loss is quantified based on the smoothed Wasserstein distance between the predicted ranking and the ranking derived from ground truth labels, with a new ranking-specific cost matrix.

C. Neural Network Based Models

The ability of neural networks to work with raw queries and documents, compared with the other learning frameworks, has created a breakthrough success and is widely applied in neural networks. There are mainly two approaches in neural ranking models – *representation based* and *interaction-based* neural ranking models.

➢ *Representation Based Neural Ranking Models*

Vector representations of queries and documents through a sequence of neural computations, and ranking is calculated based on their similarity in representations. They learn good representations and match them in the learned representation space of query and documents. DSSM[38] and its convolutional version CDSSM[39] get representations by hashing letter-tri-grams to a low dimensional vector.[40] uses pseudo labelling as a weak supervised signal to trait h representation based ranking model. Some of the works are in [41]–[44].

➢ *Interaction Based Neural Ranking Models*

The vector representations of query and documents are created, and the word level similarity of both document and query is checked before applying an additional sequence of neural computations for ranking. Some of the works are [45]–[49]

The DRMM model[47] considers more factors, such as query term importance, exact matching signals, and diverse matching requirement. DRMM[47] and KNRM[48]consider only interaction between unigrams in the query and unigrams in the document. Each matrix M_{ij} element will be cosine similarity between the *i*-th query term *j*-th document term vectors. CONV-KNRM [49] is extended KNRM incorporated n-grams in interaction matrices. Some works like[46], [50], [51] first look at the local interaction between two texts, then design different network architectures for learning more about two texts, then design different network architectures for learning more complicated interaction patterns for relevance matching.

III. CONCLUSION

In this paper, the different document ranking algorithms are compared. We have investigated different approaches toward the solution to the problems. We divided the document ranking problem solutions into three main classes based on the approaches toward the problem. They are score-based, Machine Learning based, and Neural ranking approaches. Future works in this area can be bringing out a ranking method including contents of documents mentioned as references within the same set of documents, which can in turn help in multi-document summarization for question answering applications.

REFERENCES

[1]. J. M. Ponte and W. B. Croft, “A Language Modeling Approach to Information Retrieval.”
[2]. S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.
[3]. R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Texts.”
[4]. [4] J. Zobel, “Inverted Files for Text Search Engines”, doi: 10.1145/ACM.
[5]. V. Ngoc Anh and A. Moffat, “Pruned Query Evaluation Using Pre-Computed Impacts,” 2006.
[6]. D. Roy, S. Saha, M. Mitra, B. Sen, and D. Ganguly, “I-REX: A lucene plugin for explainable IR,” in International Conference on Information and Knowledge Management, Proceedings, Nov. 2019, pp. 2949–2952. doi: 10.1145/3357384.3357859.
[7]. T. Y. Liu, “Learning to rank for Information Retrieval,” Foundations and Trends in Information Retrieval, vol. 3, no. 3, pp. 225–231, 2009, doi: 10.1561/1500000016.
[8]. F. C. Gey, “Inferring Probability of Relevance Using the Method of Logistic Regression.”
[9]. R. R. Larson, “A Fusion Approach to XML Structured Document Retrieval,” 2005.
[10]. R. Nallapati, “Discriminative Models for Information Retrieval,” 2004.
[11]. P. Li, C. J. C. Burges, and Q. Wu, “McRank: Learning to Rank Using Multiple Classification and Gradient Boosting.”
[12]. A. Shashua and A. Levin, “Ranking with Large Margin Principle: Two Approaches*.”
[13]. W. Chu, Z. U. A. Uk, and C. K. I. Williams, “Gaussian Processes for Ordinal Regression Zoubin Ghahramani,” 2005.
[14]. W. Chu and S. S. Keerthi, “New Approaches to Support Vector Ordinal Regression”.
[15]. K. Crammer and Y. Singer, “Pranking with Ranking.”
[16]. W. Chu and Z. Ghahramani, “Preference Learning with Gaussian Processes.”
[17]. T. Qin, X. D. Zhang, M. F. Tsai, D. S. Wang, T. Y. Liu, and H. Li, “Query-level loss functions for information retrieval,” Information Processing and Management, vol. 44, no. 2, pp. 838–855, Mar. 2008, doi: 10.1016/j.ipm.2007.07.016.
[18]. T. Y. Liu, “Learning to rank for Information Retrieval,” Foundations and Trends in Information Retrieval, vol. 3, no. 3, pp. 225–231, 2009, doi: 10.1561/1500000016.
[19]. G. W. Cottrell, B. Bartell, and R. Belew, “Learning to Retrieve Information Salience Using Natural Statistics View project Deep Composition View project Learning to Retrieve Information,” 1994. [Online]. Available: https://www.researchgate.net/publication/2750191
[20]. Y. Freund, R. Iyer, R. E. Schapire, Y. Singer, and T. G. Dietterich, “An Efficient Boosting Algorithm for Combining Preferences,” 2003.
[21]. Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, “Adapting Ranking SVM to Document Retrieval,” 2006.

[22]. C. L. A. Clarke, N. Fuhr, N. Kando, W. Kraaij, and A. P. de Vries, SIGIR '07 : 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval : July 23-27, 2007, Amsterdam, the Netherlands.
[23]. C. Burges, T. Shaked, E. Renshaw, N. Hamilton, and G. Hullender, "Learning to Rank using Gradient Descent."
[24]. W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to Order Things."
[25]. T. Qin, X. D. Zhang, D. S. Wang, T. Y. Liu, W. Lai, and H. Li, "Ranking with multiple hyperplanes," Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07, pp. 279–286, 2007, doi: 10.1145/1277741.1277791.
[26]. J. Kek, "Cumulated Gain-Based Evaluation of IR Techniques."
[27]. Association for Computing Machinery. Special Interest Group on Information Retrieval., H. and Web. Association for Computing Machinery. Special Interest Group on Hypertext, and ACM Digital Library., Proceeding of the 18th ACM conference on Information and knowledge management : 2009, Hong Kong, China, November 02-06, 2009. Association for Computing Machinery, 2009.
[28]. C. L. A. Clarke, N. Fuhr, N. Kando, W. Kraaij, and A. P. de Vries, SIGIR '07 : 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval : July 23-27, 2007, Amsterdam, the Netherlands.
[29]. C. L. A. Clarke, N. Fuhr, N. Kando, W. Kraaij, and A. P. de Vries, SIGIR '07 : 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval : July 23-27, 2007, Amsterdam, the Netherlands.
[30]. O. Chapelle, Q. Le, and A. Smola, "Large margin optimization of ranking measures."
[31]. Association for Computing Machinery. Special Interest Group on Information Retrieval., SIGIR '08 : the 31st annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 20-24, 2008, Singapore. Association for Computing Machinery, 2008.
[32]. M. Taylor, J. Guiver, S. Robertson, and T. Minka, "SoftRank: Optimizing Non-Smooth Rank Metrics," 2008.
[33]. T. Qin, T.-Y. Liu, and H. Li, "A General Approximation Framework for Direct Optimization of Information Retrieval Measures," 2008.
[34]. F. Xia, W. Zhang, and H. Li, "Listwise Approach to Learning to Rank-Theory and Algorithm," 2008.
[35]. Z. Cao, T. Qin, , M.-F. Tsai, H. Li , "Learning to Rank: From Pairwise Approach to Listwise Approach."
[36]. C. J. Burges, R. Ragno, and Q. Viet Le, "Learning to Rank with Nonsmooth Cost Functions."
[37]. H. T. Yu, J. M. Jose, A. Jatowt, X. Yang, H. Joho, and L. Chen, "Wassrank: Listwise document ranking using optimal transport theory," in WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining, Jan. 2019, pp. 24–32. doi: 10.1145/3289600.3291006.
[38]. [Association for Computing Machinery. Special Interest Group on Information Retrieval, H. and W. Association for Computing Machinery. Special Interest Group on Hypertext, and Association for Computing Machinery, CIKM'13 : proceedings of the 22nd ACM International Conference on Information & Knowledge Management : Oct. 27- Nov. 1, 2013, San Francisco, CA, USA.
[39]. Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management, Nov. 2014, pp. 101–110. doi: 10.1145/2661829.2661935.
[40]. M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. Bruce Croft, "Neural ranking models withweak supervision," in SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Aug. 2017, pp. 65–74. doi: 10.1145/3077136.3080832.
[41]. Association for Computing Machinery. Special Interest Group on Information Retrieval, H. and W. Association for Computing Machinery. Special Interest Group on Hypertext, and Association for Computing Machinery, CIKM'13 : proceedings of the 22nd ACM International Conference on Information & Knowledge Management : Oct. 27- Nov. 1, 2013, San Francisco, CA, USA.
[42]. J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng, "Modeling Interestingness with Deep Neural Networks." [Online]. Available: http://dumps.wiki-
[43]. Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management, Nov. 2014, pp. 101–110. doi: 10.1145/2661829.2661935.
[44]. W. Uddin Ahmad, K.-W. Chang, and H. Wang, "MULTI-TASK LEARNING FOR DOCUMENT RANKING AND QUERY SUGGESTION." [Online]. Available: https://github.com/wasiahmad/mnsrf
[45]. J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for Ad-hoc retrieval," in International Conference on Information and Knowledge Management, Proceedings, Oct. 2016, vol. 24-28-October-2016, pp. 55–64. doi: 10.1145/2983323.2983769.
[46]. B. Hu, Z. Lu, H. Li, and † Qingcai Chen, "Convolutional Neural Network Architectures for Matching Natural Language Sentences." [Online]. Available: http://www.noahlab.com.hk/technology/Learning2Match.html
[47]. J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for Ad-hoc retrieval," in International Conference on Information and Knowledge Management, Proceedings, Oct. 2016, vol. 24-28-October-2016, pp. 55–64. doi: 10.1145/2983323.2983769.

[48]. C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-To-end neural ad-hoc ranking with kernel pooling," in SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Aug. 2017, pp. 55–64. doi: 10.1145/3077136.3080809.
[49]. Z. Dai, J. Callan, C. Xiong, and Z. Liu, "Convolutional neural networks for soft-matching N-grams in ad-hoc search," in WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Feb. 2018, vol. 2018-Febuary, pp. 126–134. doi: 10.1145/3159652.3159659.
[50]. S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN," Apr. 2016, [Online]. Available: http://arxiv.org/abs/1604.04378
[51]. L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text Matching as Image Recognition." [Online]. Available: www.aaai.org