A Fusion Approach for Web Search Result Diversification Using Machine Learning Algorithms

Lekshmi Kalinathan[1], Prabavathy Balsundaram[2], Yogesh Munees E[3], Siddharth S[4], Shrijith Mr[5], Shrikeshavinee Ramachandran[6], Shruti Sriram[7] and Ramdhanush Venkatakrishnan[8]

Abstract

Result diversification provides a broader view of a search topic while also increasing the possibilities of finding relevant information. It has shown to increase user satisfaction in recommender systems and web search. Many different approaches have been proposed in the related literature for the diversification problem. Since the web search result is huge in size, it is essential to have an efficient fusion approach. Hence, objective of this paper is to propose the implementation of fusion model based on KNN, CART and SVR regressors. This fusion model aims to improve the accuracy and reduce the error value of the result that is generated.

Keywords

Voting Regressor, Ensembling approach, CART, Fusion methods

1. Introduction

Currently, the clients of search engines are interested in retrieving pieces of information that cover many aspects of their information needs. Consequently, result diversification has attracted considerable attention in order to improve user satisfaction. For example, if a user wants to search for car of a type, a diverse result containing various brands, models with different horsepower and other technical characteristics is intuitively more informative than a result that contains a homogeneous result containing only the cars with similar features.

Diversification is also helpful to offset the influence of personalization. Personalization aims at tailoring results to meet the preferences of each specific individual. However, this can leads to excessive restriction of search results. Diversification can complement preferences and provide personalization systems with the means to retrieve more satisfying results.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

lekshmik@ssn.edu.in (L. Kalinathan); prabavathyb@ssn.edu.in (P. Balsundaram); yogesh2010695@ssn.edu.in (Y. M. E); siddharth2010622@ssn.edu.in (S. S); shrijith2010542@ssn.edu.in (S. Mr); shrikeshavinee2010573@ssn.edu.in (S. Ramachandran); shruti2010197@ssn.edu.in (S. Sriram); ramdhanush2010105@ssn.edu.in (R. Venkatakrishnan)

2. Related Work

Jiang et al [1] have proposed a learning framework for explicit result diversification in which subtopics are explicitly modeled. Recurrent Neural Networks, attention mechanism and Max Pooling are used to instantiate the framework. The framework is flexible to model query intent in a flat list or a hierarchy. Regardless of diversity modeling and optimization methods, all these solutions inherit the spirit of MMR, which is an implicit approach and disregards intent.

Göynük et al [2] have proposed R-LTR, a supervised learning approach for textual data and modified to allow tuning the weights of visual and textual features separately. The experiments using a benchmark dataset v with 153 queries and 45K images reveal that the proposed method significantly outperforms various ad hoc diversification approaches in terms of the sub-topic recall metric. Furthermore, certain variants of R-LTR are superior to the original method and provide additional (relative) gains of up to 2.2%.

Lu et al [3] proposed a multidimensional evaluation mechanism for analysing the results of diversification in image retrieval. Experiments were carried out to evaluate three different semantic distance algorithms (WordNet, Google Distance, and ESA) combined with three re-ranking algorithms (MMR, xQuAD, and Score Difference) on image diversification retrieval based on a subset of the NUS-WIDE image dataset. It allowed a comparison of these algorithms on social tags and visual tags to understand their strengths and weaknesses, and a comparison of visual distance algorithms to prove the effectiveness of semantic information in result diversification.

MacAvaney et al [4] proposed a paper which introduces a new distributional causal language modelling objective and a representation replacement strategy to better handle ambiguous queries.It is found that IntenT5 excels at handling faceted queries and improves ColBERT's performance for fully specified queries. Given that ambiguous queries appear to be difficult to handle, it investigates Distributional Causal Language Modeling for overcoming this problem.

Maxwell et al [5] proposed that, the impact of diversification is explored where searchers undertake complex search tasks using Interactive Information Retrieval (IIR). The goal of the system is to help the searcher learn about a topic and the number of aspects that the searcher finds indicates how much they learned during the process. This finding suggested that the participants seemed largely ambivalent to the difference in the performance of the systems.

Omer Sagi et al [6] have studied various ensembling techniques like AdaBoost, Bagging, Random Forest, Gradient Boosting Machines, Rotation Forest, Extremely Randomized Trees and Deep Neural Networks. They suggested that these existing ensemble methods certainly improve predictive performance by avoiding overfitting, decreasing the risk of obtaining local minima and expanding the search space. The research direction in this survey includes the refinement of popular algorithms suitable for big data and distribution of ensembling algorithms across multiple machines.

DONG et al[7] grouped ensemble learning into four categories: classification based supervised and semi-supervised ensemble methods, clustering based supervised and semi-supervised ensemble methods. It presented challenges and possible research directions for each mainstream approach of ensemble learning. Further, it also suggests that the performance of ensemble method can be improvised by fusing it with the deep learning and reinforcement learning.

3. Task and Dataset Description

The dataset used for this task is from ImageCLEF 2022 [8]. Result diversification fusion task [9] focuses on identifying relevant outputs and optimizing them, given a query. An inducer is a model which predicts images related to a query. The similarity scores and ranks are calculated and written into a text file during run time. However, a single inducer cannot be worked with, as it might have low precision or performance. Hence an extended version containing many inducers are used to make the predictions more precise. This is called ensembling technique. The performance given by the ensembling technique even tops the performance of the highest single inducer.
The data for this task is obtained from the Retrieving Diverse Social Images Task dataset [Ionescu2020]. The outputs of 56 inducers, each stored in a separate text file, representing a total of 123 queries (topics). Each entry or row in these files is of the format as given below in the Table 1.

Table 1
Attributes of inducer file

Fields	Representation
query_id	Unique id of the query
inter	Ignored value
photo_id	Unique photo id
rank	Photo rank
sim	Similarity score
run_name	Name of inducer

4. Methodologies used

In order to understand the evaluation of the similarity scores by the inducers, three predictors namely, KNN Regressor, Classification and Regression Tree and Support Vector Regressor were explored.

4.1. KNN Regressor

KNN (K - Nearest Neighbors) is a supervised machine learning model which classifies a data point based on feature similarity. It is called a lazy learner as not much training is required. KNN is applied to both classification and regression problems. It gives accurate results when the dataset is small and when it is properly labeled. First, the Euclidean distances of the new data point are calculated from all the training points and K of its closest neighbors are considered. Then, out of the K neighbors, the number of points in each class are counted. The new data point is classified into the class in which a majority of its neighbors lie. KNN regressor uses the same distance functions as KNN classification. After having the distances calculated, the samples of k smaller distances were found and the target values are averaged to obtain the predicted value.

4.2. Classification and Regression Trees

While classification attempts to predict a class label, regression predicts a probabilistic value corresponding to a class label. The CART algorithm [10] works by searching for the best homogeneity for the subnodes in the decision tree with the help of the Gini Index criterion. The root node is taken as the training set and is split into two by considering the best attribute and threshold value. Further, the subsets are also split using the same logic. This continues until the last pure subset is found or the maximum number of leaves possible in that growing tree. Each leaf node represents a class label where decisions are taken after computing all features.

4.3. Support Vector Machine Regressor

Support Vector Regression is a regression algorithm [11] that supports both linear and non-linear regressions. This method works on the principle of the Support Vector Machine, i.e., it finds a hyperplane in an N-dimensional space that distinctly classifies the data points. SVR differs from SVM in the way that SVM is a classifier that is used for predicting discrete categorical labels while SVR is a regressor that is used for predicting continuous ordered variables. In simple regression, the idea is to minimize the error rate while in SVR the idea is to fit the error inside a certain threshold which means, work of SVR is to approximate the best value within a given margin called - tube. The objective of SVR is to fit as many data points as possible without violating the margin.

5. Implementation

The dataset is split into 90%-10% where the 90% is used for training and the remaining 10% is used for validation data. The similarity score is normalized and extracted to ensure the same data distribution. The training data is used as the input while the output is reflected on the validation data. Three predictor models P_1, P_2 and P_3 were built to study how similarity scores are assigned.

The predictor P_1 was implemented using KNeighborsRegressor function of sklearn[1].neighbors library. Regressor parameters n_neighbors, weights, metric as 5, uniform and l2_distance respectively are initialized. The regressor is fitted on the training dataset X and y is obtained as their responses. The K-neighbors of a data point are found. Further, the weighted graph for k neighbors is calculated for the data points in X. With the model obtained, target is predicted for the given test data.
The predictor P_2 was implemented using DecisionTreeRegressor function of sklearn.tree library. Regressor parameters criterion, splitter, min_samples_split and min_samples_leaf as squared_error, best, 2 and 1 respectively are initialized. These parameter values lead to fully grown and unpruned trees for this dataset. In addition, it also helps in the reduction of memory consumption. Decision tree regressor observes similarity scores of each inducer and trains a model to predict the data in future. The predictor P_3 was implemented using sklearn.svm with the following parameter values. SVR finds the curve for the given input similarity scores.

Parameter:Value	Kernel:rbf, degree:3, gamma:scale, Coef:0.0, epsilon:0.1, shrinking:True, Cache_size:200, max_iter:-1

However, instead of using the curve as a decision boundary, it uses the curve to find the match between the support vector and position of the curve. SVR acknowledges the presence of non-linearity in the input similarity score and provides a proficient prediction model. The predictors are trained with 90% of the training dataset and the remaining set is validated. The data is split into batches of 5, and each batch is used to predict the similarity score of the next input for each predictor. Then a comparison between the actual and predicted values is done and the deviation between the two models is calculated as error. Ranks have been assigned for the models P_1, P_2 and P_3 based on the error values. The ranks of the base regressors and the predictor models are used to construct the voting regressor. It is trained using the entire training data set. The output of each batch predicts the output for the next input. Error values are calculated by comparing the actual and the predicted values of the model. It is then tested on the test data and the similarity score predicted by the model replaces the original similarity score.

6. Results and Analysis

Voting regressor is used to predict the similarity scores of the models present in the validation dataset. The predicted and actual values of similarity scores were plotted in figures 1, 2, 3 and 4. From the figures 1, 2 and 3, it can be inferred that P_2 predicts better similarity scores as compared to P_1 and P_3. Furthermore on analyzing figures 2 and 4, it can be seen that the predictor P_2 predicts better values as compared to the voting regressor. Table 2 shows the MAE and rank values for the base and voting regressor models.

[1] https://scikit-learn.org/stable/

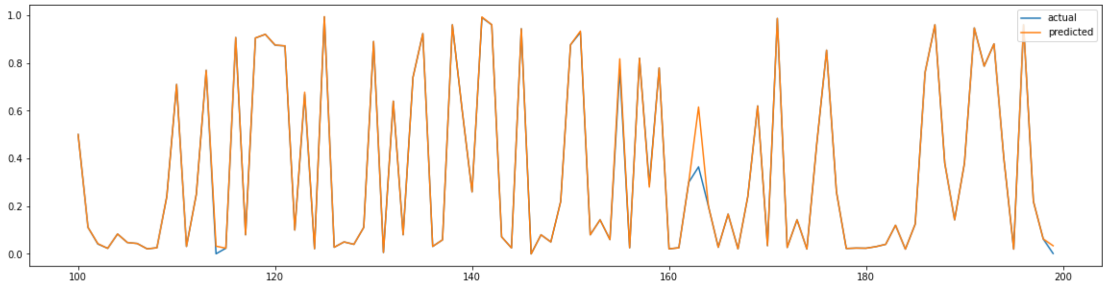

Figure 1: Actual versus Predicted similarity scores using KNN Regressor

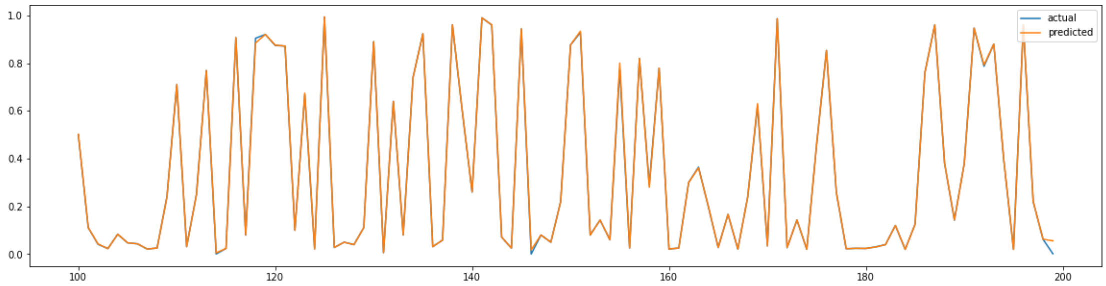

Figure 2: Actual versus Predicted similarity scores using CART Regressor

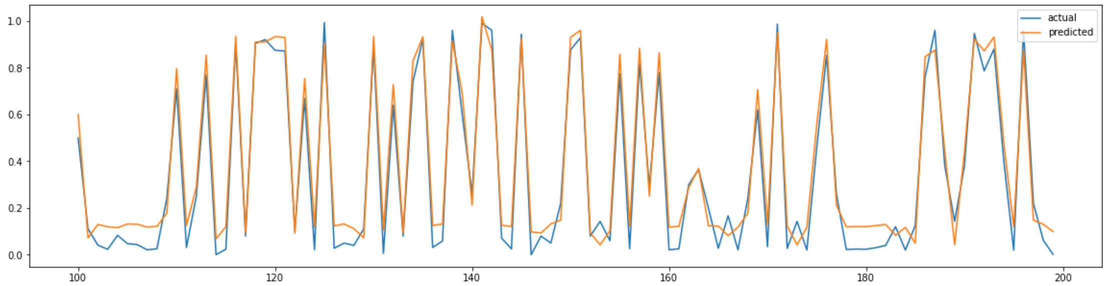

Figure 3: Actual versus Predicted similarity scores using SVR Regressor

Given the superior performance of the CART model, we chose it for creating our submission runs. It has been tested with CLEF test data using metrics F1 measure and cluster recall. Different clusters from many cluster labels are assessed under the cluster recall metric and the harmonic mean of cluster recall and precision equates to the F1 measure.

The CART model is used to predict the updated similarity score values for the test data. The test data contains 175,591 predicted values which are divided into 56 text files with around 3150 entries each. Ten different CART models were built by varying the paramater, iteration size.

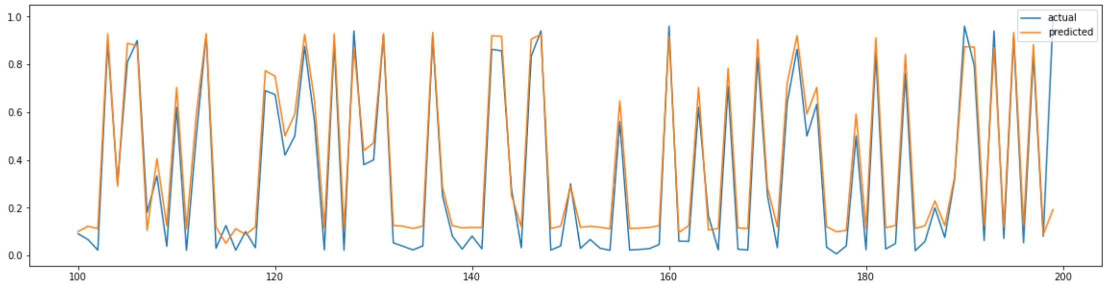

Figure 4: Actual versus Predicted similarity scores using Voting Regressor

Table 2
Performance of base and voting regressors

Models	**MAE**	**Rank**
P1 (K-Nearest Neighbors)	0.004	2
P2 (CART)	0.003	1
P3 (SVR)	0.085	3
Voting Regressor	0.017	-

Table 3
F1 and Cluster Recall scores for 10 runs of CART model

Run No.	**F1 score**	**CR score**
1	0.5029	0.3925
2	0.5223	0.4087
3	0.5352	0.4036
4	0.5521	0.4295
5	0.5628	0.4407
6	0.5489	0.4297
7	0.5634	0.4414
8	0.5489	0.4297
9	0.4983	0.3778
10	0..4948	0.3734

Each variation of the CART model was tested with a given test data and the results obtained were submitted. For each submission run, we have obtained the F1 and CR scores and it has been tabulated in the Table 3. A cluster recall of 0.4414 and a F1 measure of 0.5634 has been obtained for the 10 runs.

7. Conclusion

To improve the accuracy of the results of the inducers, three regressors were implemented in the voting regressor. The model was trained on data from 56 different inducers, containing 167,139 training values and tested on data from 56 inducers, containing 175,591 testing values. The base regressors obtained MAE values of 0.004 for KNN, 0.003 for CART and 0.085 for SVR. The voting regressor yielded a MAE of 0.017. Among the implemented regressors, CART provided the optimised result. Of the 10 best submissions, the best F1 score and CR score are 0.5634 and 0.4414 respectively.

References

[1] Jiang Z, Dou Z, Zhao WX, Nie JY, Yue M, Wen JR, Supervised search result diversification via subtopic attention, IEEE Transactions on Knowledge and Data Engineering. 10(2018) 1971-84.

[2] Göynük B. Supervised learning for image search result diversification. (Middle East Technical University,2019)

[3] Lu W, Luo M, Zhang Z, Zhang G, Ding H, Chen H, Chen J. Result diversification in image retrieval based on semantic distance. Information Sciences. 2019 Oct 1;502:59-75.

[4] McAvaney S, Macdonald C, Murray-Smith R, Ounis I. IntenT5: Search Result Diversification using Causal Language Models. arXiv preprint arXiv:2108.04026. 2021 Aug 9.

[5] Maxwell D, Azzopardi L, Moshfeghi Y. The impact of result diversification on search behaviour and performance. Information Retrieval Journal. 2019 Oct;22(5):422-46.

[6] Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining And Knowledge Discovery.* **8**, e1249 (2018)

[7] Dong, X., Yu, Z., Cao, W., Shi, Y. & Ma, Q. A survey on ensemble learning. *Frontiers Of Computer Science.* **14**, 241-258 (2020)

[8] Ionescu, B., Müller, H., Peteri, R., Rückert, J., Ben Abacha, A., Herrera, A., Friedrich, C., Bloch, L., Brüngel, R., Idrissi-Yaghir, A., Schäfer, H., Kozlovski, S., Cid, Y., Kovalev, V., Ştefan, L., Constantin, M., Dogariu, M., Popescu, A., Deshayes-Chossart, J., Schindler, H., Chamberlain, J., Campello, A. & Clark, A. Overview of the ImageCLEF 2022: Multimedia Retrieval in Medical, Social Media and Nature Applications. *Experimental IR Meets Multilinguality, Multimodality, And Interaction.* (2022,9,5)

[9] Ştefan, L., Constantin, M., Dogariu, M. & Ionescu, B. Overview of ImageCLEFfusion 2022 Task - Ensembling Methods for Media Interestingness Prediction and Result Diversification. *CLEF2022 Working Notes.* (2022,9,5)

[10] Breiman, L., Friedman, J., Olshen, R. & Stone, C. Classification and regression trees. Belmont, CA: Wadsworth. *International Group.* **432**, 9 (1984)

[11] Drucker, H., Burges, C., Kaufman, L., Smola, A. & Vapnik, V. Support vector regression machines. *Advances In Neural Information Processing Systems.* **9** (1996)