Document Performance Prediction for Automatic Text Classification

Gustavo Penha[1], Raphael Campos[2], Sérgio Canuto[2], Marcos André Gonçalves[2], and Rodrygo L. T. Santos[2]

[1] Delft University of Technology
g.penha-1@tudelft.nl
[2] Computer Science Department, Universidade Federal de Minas Gerais
{rcampos,sergiodaniel,mgoncalv,rodrygo}@dcc.ufmg.br

Abstract. Query performance prediction (QPP) is a fundamental task in information retrieval, which concerns predicting the effectiveness of a ranking model for a given query in the absence of relevance information. Despite being an active research area, this task has not yet been explored in the context of automatic text classification. In this paper, we study the task of predicting the effectiveness of a classifier for a given document, which we refer to as document performance prediction (DPP). Our experiments on several text classification datasets for both categorization and sentiment analysis attest the effectiveness and complementarity of several DPP inspired by related QPP approaches. Finally, we also explore the usefulness of DPP for improving the classification itself, by using them as additional features in a classification ensemble.

Keywords: Performance prediction, automatic text classification

1 Introduction

Query performance prediction (QPP) is a challenging and fundamental problem in information retrieval. It concerns predicting the effectiveness of a ranking model when there is no relevance information available. Applications for QPP include selecting the best model depending on query features, combining multiple ranking models and requesting more information for potentially poorly formulated queries. QPP approaches have been divided into pre-retrieval [9, 13, 15, 16, 21, 35] and post-retrieval [6, 18, 23, 24, 30, 31, 33], depending on the information used by the method. Inspired by QPP, in this paper, we derive and adapt methods for document performance prediction (DPP), which aim at predicting the performance of automatic text classifiers.

2 Document Performance Prediction

The performance of automatic text classifiers is usually measured by their average effectiveness over test documents. However, this performance can vary

depending on the specific document in question. Inspired by query performance prediction, we define the task of document performance prediction as predicting the effectiveness of a text classifier for a given document, when labeled data is not available. Formally, a document performance predictor π can be defined as a function $\pi : D \times M \rightarrow Y$, where D and M denote the space of all documents and classifiers probability outputs, respectively, and Y denotes the space of possible effectiveness assessments given a pair $\langle d, m \rangle$, $d \in D$, $m \in M$. An effective predictor $\pi(d, m)$ approximates the true effectiveness $\Delta(\hat{y}_{dm}, y_{dm})$ as accurately as possible, where Δ is any classification effectiveness metric defined over the classification output $\hat{y}_{dm} = m(d)$ and the label y_{dm}. In our experiments in Section 3, we use cross-entropy as a representative evaluation metric Δ.

Depending on the information used by a document performance predictor π, it may fall into one of two categories: pre-classification and post-classification. In particular, a *pre-classification* DPP relies solely on the contents of document d to make its performance prediction. In contrast, inspired by post-retrieval query performance predictors, which leverage the ranked list produced by a target ranking model, a *post-classification* DPP uses the classification output $\hat{y}_{dm}$ in addition to the contents of document d. In the remainder of this section, we propose several pre- and post-classification approaches for DPP.

2.1 Pre-Classification DPP

Inspired by prior work on ad-hoc retrieval, we adapted pre-retrieval query performance predictors for DPP. Instead of applying these methods on the query q, we apply them to document d. Some of our proposed DPP also require statistics from a corpus T, comprising documents used for training the classifier.

dT-Stats. Our first category of pre-classification DPP, denoted dT-Stats, includes predictors that rely only on the document d and the training corpus T. These predictors are independent of the classifier and were inspired by several pre-retrieval QPP methods [13]. In particular, *tokenCount* and *termCount* are the total number of tokens and the number of unique tokens in the document, respectively. *AvQL* is the average character size of the tokens in the document. {*Av,Max,Dev*}*-IDF* are the average, maximum and standard deviation of the inverse document frequency of the document terms. *AvICTF* is the average inverse collection term frequency of the document terms, defined as $AvICTF = \frac{1}{n}\sum_{i=1}^{n}[\log_2(cf_i) - \log_2(tf_{i,d})]$, where n is the number of terms in the document, cf_i is the collection frequency of the i-th term and $tf_{i,d}$ is its term frequency in d. *SCS* is the simplified clarity score of document terms, i.e., $SCS \approx \log_2 \frac{1}{n} + \frac{1}{n}\sum_{i=1}^{n}[\log_2(cf_i) - \log_2(tf_{i,d}]$. {*Av,Sum,Max*}*-SCQ* are the average, maximum and standard deviation of the collection document similarity. *AvP* is the average number of senses for document terms, using WordNet function `wordnet.synsets`, and *AvNP* is the average number of noun senses among these. *Av-*{*Path,LCH,WUP*} are the relatedness of a sample of 50 terms from the document by calculating all their pairwise similarities, using three similarity functions provided by WordNet: Path, Leacock-Chodorow, and Wu-Palmer.

d-Latent. Our second category of pre-classification DPP, denoted d-Latent, includes two predictors that are based on a latent representation of document d. In particular, $\{Max,Avg\}$-*PoolingGlove* denote the maximum and average of each of 50 Glove dimensions from "glove.6B.zip"[3] for the document terms.

2.2 Post-Classification DPP

Recent work has shown that post-retrieval query performance predictors are state-of-the-art in ad-hoc retrieval [18, 26, 27, 33]. Unlike QPP, we do not have access to a list of documents retrieved for a query. Instead, we have a probability distribution $\hat{y}_{dm}$ of the classes predicted by a classifier m for document d.

DistBased. Predictors from this category assign the relevance of a document d to each class by calculating distances between a document d and each class centroid or between d and its k nearest neighbors (10 in our experiments) from each class. Here, we use the distance scores (Cosine, Euclidean and Manhattan) themselves as predictors that exploit the combination of global and local information about the distribution of documents in each class, as described in prior works on document classification with distance-based features [12, 22].

BaggBased. Predictors from this category relate to the approach of Roitman et al. [27] and other approaches that estimate the variance of the retrieved lists [10, 23, 30, 32]. Here we bootstrap the estimators from the bagging-based models and use the variance of their predictions for document classes instead of the scores of top-retrieved documents. *BaggCVariance* is the standard deviation of each class predicted probability for n (20 in our experiments) random base estimators sampled j (50 in our experiments) times for each classification bagging model m from {RF [3], Bert [5], Broof [29]} and $n\_estimators = 200$ (which is the number of base models included in the bagging model). *BaggQ*$\{25,50,75\}C$ is similar to *BaggCVariance*, but instead of the standard deviation, we calculate the 25, 50 and 75 quantiles from the class prediction probabilities. *PredEntropy* is a vector containing the entropy of the base estimators predictions probability distribution for each bagging classification model m from {RF,Bert,Broof}. *NumPredC* is a vector containing the number of distinct classes (estimated probability not zero) in the base estimators predictions for each bagging classification model m from {RF,Bert,Broof}.

ProbPBased. DPP in this category use the prediction of any classifier, being agnostic to their inductive biases. *ProbPred* are the probability predictions $\hat{y}_{dm}$ of each class for each classification model m, resulting in a vector of dimensionality $|M|\times|C|$, where M are all the classification models the performances of which are being predicted and C is the target set of classes. *ProbPredVar* is the standard deviation of probability predictions of each class for each classification model, resulting in a vector of dimensionality $|M|\times|C|$. *ProbPBased* encompasses the 25, 50 and 75 quantiles of probability predictions of each class for each classification model, resulting in a vector of dimensionality $|M| \times |C| \times 3$.

[3] http://nlp.stanford.edu/data/glove.6B.zip

3 Evaluation

In this section, we aim to answer the following research questions:

Q1. How effective are the proposed DPP?
Q2. How complementary are the proposed DPP?
Q3. How effective are DPP for enhancing a classification ensemble?

3.1 Experimental Setup

We explored two categorization datasets, 20Newsgroups (20NG) and 4Universities (4UNI, aka WEBKB) with about 20,000 and 8,200 documents respectively. We also evaluate our approaches for the sentiment analysis task. We considered four data sets of messages labeled as positive or negative from distinct domains: Amazon, BBC, NYT, YouTube [4]. Inspired by prior work on QPP [9, 27, 30, 33], we compute the correlation between the predicted performance $\pi(d, m)$ and the actual performance $\Delta(\hat{y}_{dm}, y_{dm})$ for document d and classifier m. In particular, we measure the actual performance of m as the cross-entropy between the predicted class distribution $\hat{y}_{dm}$ and the true distribution y_{dm}. The higher the cross-entropy Δ, the more distant the two distributions.

We predict the performance of several classifiers, $M = \{$XGBoost [7], KNN [1], NaiveBayes [34], Bert [5], Broof [29], RandomForest [3], SVM [17], MLP [11]$\}$. Except for Bert and Broof,[4] we used scikit-learn v0.18 implementations and their default hyperparameters, with TF-IDF document representations as input. To evaluate our proposed DPP, we perform a 5-fold cross-validation. In each round, four folds serve as the training corpus T and the remaining fold is used to calculate the correlation between the predicted and actual performance of each classifier m, averaged across all models in M. Accordingly, we report the mean of the average correlation obtained by each DPP across the five test folds.

3.2 DPP Effectiveness

To address $Q1$, Table 1 shows the mean average correlation coefficient (Pearson's ρ and Kendall's τ) attained by the best-performing DPP in each of the categories described in Sections 2.1 and 2.2. The most successful predictors are from the categories *BaggBased* and *ProbPBased*, which comprise post-classification predictors. This result is somewhat expected given that post-retrieval QPP are state-of-the art. *dT-Stats* and *d-Latent* predictors are not effective for the sentiment analysis datasets. However, they achieve higher correlations in the categorization datasets (4UNI and 20NG). We believe this happens because sentiment analysis datasets are smaller in number of documents as well as in document length, hurting statistics taken on the document and corpus. Finally, *DistBased* predictors were ineffective in our experiments. We attribute this to the fact that neighbor information is used only by one of the eight classifiers whose performance we are predicting (KNN). For the other seven classifiers, this inductive bias does not hold, hence it is not a good predictor of their performance.

[4] https://github.com/raphaelcampos/stacking-bagged-boosted-forests

Table 1. Effectiveness of document performance prediction strategies in terms of Pearson's ρ and Kendall's τ correlation with the cross-entropy loss.

Method	4UNI		20NG		Amazon		BBC		NYT		YouTube	
	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ
BaggBased	.129	**.195**	.205	**.302**	.163	.203	.207	**.488**	0.09	.114	.241	.303
ProbPBased	**.250**	.187	**.408**	.157	**.240**	**.274**	**0.39**	.397	**.113**	**.123**	**.421**	**.448**
DistBased	.017	.024	.030	.026	.019	.026	.040	.059	.020	.030	.017	.026
d-Latent	.121	.149	.081	.092	.020	.029	.046	.076	.020	.030	.039	.047
dT-Stats	.175	.159	.265	.263	.019	.032	.038	.058	.017	.027	.016	.028

Table 2. Effectiveness of the combination of document performance prediction using different groups of methods as input space, in terms of Pearson's ρ and Kendall's τ correlation with the cross-entropy loss. Superscripts $^{\dagger}/^{\ddagger}$ denote statistically significant improvements over the best raw DPP at 95%/99% confidence intervals.

Input space	4UNI		20NG		Amazon		BBC		NYT		YouTube	
	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ
Best raw DPP	.250	.195	.408	.302	.240	.274	.390	.488	.113	.123	.421	.448
BaggBased	.347‡	.471†	.386	.516†	.304†	.400†	.257	.536†	.145†	.203†	.401	.512†
ProbPBased	.439†	.556†	.554‡	**.720**†	.250‡	.325†	.553†	.613†	.112	.148†	**.444**†	.468†
TF-IDF	.230	.296‡	.334	.276	.243	.314†	.242	.260	.103	.158†	.325	.363
d-Latent	.229	.291	.099	.120	.015	.022	.033	.045	.014	.022	.021	.032
DistBased	.030	.051	.029	.017	.010	.014	.030	.041	.015	.022	.029	.045
dT-Stats	.220	.258	.297	.360	.008	.012	.022	.038	.010	.012	.018	.026
All-pre-clf	.241	.301	.300	.367	.014	.019	.029	.041	.012	.018	.016	.026
All-post-clf	.478†	**.631**†	.538	.382	**.317**†	**.423**†	**.563**†	**.703**†	**.159**†	**.218**†	.440‡	**.531**†
All	**.479**†	.628†	**.582**†	.437	.288†	.381†	.464	.624†	.132‡	.183†	.396	.488†

3.3 DPP Complementarity

The combination of predictors through machine learning has been explored for improving query performance predictors, with the assumption that they capture complementary information [2, 8, 14, 20, 31, 36]. To address $Q2$, we assess the complementarity of the proposed DPP for a given classifier m when used as input features for a machine-learned DPP (ML-DPP) aimed to predict m's actual performance. Table 2 shows the effectiveness of several different groups of DPP used as input features for a ML-DPP based on a random forest regressor. The single best-performing DPP is included as a baseline. For all datasets, we can significantly improve upon the single best DPP by combining multiple DPP. Therefore, recalling $Q2$, we conclude that the proposed DPP have a degree of complementarity and capture different types of information.

Table 3. Average macro F1 score for an ensemble of eight classifiers with DPP as additional meta-features. Superscripts $^{\dagger}/^{\ddagger}$ denote statistically significant improvements compared to not using additional meta-features at 95%/99% confidence intervals.

	20NG	4UNI	Amazon	BBC	NYT	YouTube
Stacking	.939	.779	.759	.769	.630	.759
+ BaggBased	**.949**† (1.1%)	**.851**† (9.2%)	**.836**† (10.1%)	**.878**† (14.2%)	**.761**† (20.8%)	**.858**† (13.0%)
+ ProbPBased	.939 (0.0%)	.779 (0.0%)	.755 (-0.5%)	.783 (1.8%)	.628 (-0.3%)	.757 (-0.3%)
+ DistBased	.946† (0.7%)	.772 (-0.9%)	.741 (-2.4%)	.770 (0.1%)	.635 (0.8%)	.765 (0.8%)
+ d-Latent	.941 (0.2%)	.779 (0.0%)	.733 (-3.4%)	.731 (-4.9%)	.635 (0.8%)	.752 (-0.9%)
+ dT-Stats	.939 (0.0%)	.778 (-0.1%)	.748 (-1.4%)	.751 (-2.3%)	.629 (-0.2%)	.766 (0.9%)
+ ml BaggBased	.939 (0.0%)	.785 (0.8%)	.750 (-1.2%)	.790 (2.7%)	.631 (0.2%)	.761 (0.3%)
+ ml ProbPBased	.940 (0.1%)	.774 (-0.6%)	.756 (-0.4%)	.787 (2.3%)	.639 (1.4%)	.770 (1.4%)
+ ml DistBased	.939 (0.0%)	.780 (0.1%)	.757 (-0.3%)	.777 (1.0%)	.628 (-0.3%)	.765 (0.8%)
+ ml d-Latent	.939 (0.0%)	.780 (0.1%)	.753 (-0.8%)	.788 (2.5%)	.641 (1.7%)	.761 (0.3%)
+ ml dT-Stats	.939 (0.0%)	.779 (0.0%)	.756 (-0.4%)	.766 (-0.4%)	.628 (-0.3%)	.763 (0.5%)

3.4 Application: Enhancing Classification Ensembles

Improved QPP does not automatically translate to improved retrieval [25]. Roitman et al. [28] demonstrated through simulations that a minimum correlation of $\rho > 0.35$ would be necessary for a QPP to be useful. Although this barrier has been surpassed by several QPP in the literature, their observed utility for ad-hoc retrieval has been marginal [19]. To address $Q3$, we assess the usefulness of DPP for improving text classification, by employing DPP as additional meta-features to a stacking layer, which combines the output of the eight classifiers in M.

Table 3 summarizes our results in terms of macro F1, comparing the addition of groups of DPP against the stacking strategy. We obtained significant improvements with only one strategy, *BaggBased*, which is one of our most accurate DPP. However, several other sets of features that are also accurate for the DPP task did not translate to improvements in classification. We hypothesize that having a high accuracy in the performance prediction task is not sufficient for a DPP to improve the classification ensemble, as our empirical results corroborate.

4 Conclusions

We proposed several document performance predictors (DPP) for automatic text classification. We demonstrated their effectiveness and complementarity by thorough experiments on both categorization and sentiment analysis datasets. Moreover, we showed an application for DPP in improving automatic text classification ensembles, with state-of-the-art results. As future work, we plan to investigate why predictors with high correlations on the document performance prediction task do not necessarily translate into improved text classification.

Acknowledgements. Work partially funded by project MASWeb (FAPEMIG APQ-01400-14) and by the authors' individual grants from CNPq and FAPEMIG.

Bibliography

[1] Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)

[2] Bashir, S.: Combining pre-retrieval query quality predictors using genetic programming. Appl. Intell. **40**(3), 525–535 (2014)

[3] Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)

[4] Campos, R., Canuto, S., Salles, T., de Sá, C.C., Gonçalves, M.A.: Stacking bagged and boosted forests for effective automated classification. In: Proc. of SIGIR, pp. 105–114 (2017)

[5] Campos, R., Canuto, S., Salles, T., de Sá, C.C., Gonçalves, M.A.: Stacking bagged and boosted forests for effective automated classification. In: Proc. of SIGIR, pp. 105–114 (2017)

[6] Carmel, D., Yom-Tov, E.: Estimating the query difficulty for information retrieval. Synthesis Lectures on Information Concepts, Retrieval, and Services **2**(1), 1–89 (2010)

[7] Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proc. of SIGKDD, pp. 785–794, ACM (2016)

[8] Chifu, A.G., Laporte, L., Mothe, J., Ullah, M.Z.: Query performance prediction focused on summarized LETOR features. In: Proc. of SIGIR, pp. 1177–1180 (2018)

[9] Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proc. of SIGIR, pp. 299–306 (2002)

[10] Cummins, R., Jose, J., O'Riordan, C.: Improved query performance prediction using standard deviation. In: Proc. of SIGIR, pp. 1089–1090 (2011)

[11] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

[12] Gopal, S., Yang, Y.: Multilabel classification with meta-level features. In: Proc. of SIGIR, pp. 315–322 (2010)

[13] Hauff, C.: Predicting the effectiveness of queries and retrieval systems. Ph.D. thesis, EEMCS (2010)

[14] Hauff, C., Azzopardi, L., Hiemstra, D.: The combination and evaluation of query performance prediction methods. In: Proc. of ECIR, pp. 301–312 (2009)

[15] Hauff, C., Hiemstra, D., de Jong, F.: A survey of pre-retrieval query performance predictors. In: Proc. of CIKM, pp. 1419–1420 (2008)

[16] He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Proc. of SPIRE, pp. 43–54 (2004)

[17] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. App. **13**(4), 18–28 (1998)

[18] Kurland, O., Shtok, A., Carmel, D., Hummel, S.: A unified framework for post-retrieval query-performance prediction. In: Proc. of ICTIR, pp. 15–26 (2011)

[19] Macdonald, C., Santos, R.L.T., Ounis, I.: On the usefulness of query features for learning to rank. In: Proc. of CIKM, pp. 2559–2562 (2012)

[20] Mizzaro, S., Mothe, J., Roitero, K., Ullah, M.Z.: Query performance prediction and effectiveness evaluation without relevance judgments: two sides of the same coin. In: Proc. of SIGIR, pp. 1233–1236 (2018)
[21] Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty. In: Proc. of QP Workshop at SIGIR, pp. 7–10 (2005)
[22] Pang, G., Jin, H., Jiang, S.: CenKNN: a scalable and effective text classifier. Data Min. Knowl. Discov **29**(3), 593–625 (2015)
[23] Pérez-Iglesias, J., Araujo, L.: Standard deviation as a query hardness estimator. In: Proc. of SPIRE, pp. 207–212 (2010)
[24] Raiber, F., Kurland, O.: Using document-quality measures to predict web-search effectiveness. In: Proc. of ECIR, pp. 134–145 (2013)
[25] Raiber, F., Kurland, O.: Query-performance prediction: setting the expectations straight. In: Proc. of SIGIR, pp. 13–22 (2014)
[26] Roitman, H.: Query performance prediction using passage information. In: Proc. of SIGIR, pp. 893–896, ACM (2018)
[27] Roitman, H., Erera, S., Weiner, B.: Robust standard deviation estimation for query performance prediction. In: Proc. of ICTIR, pp. 245–248 (2017)
[28] Roitman, H., Hummel, S., Kurland, O.: Using the cross-entropy method to re-rank search results. In: Proc. of SIGIR, pp. 839–842 (2014)
[29] Salles, T., Gonçalves, M., Rodrigues, V., Rocha, L.: BROOF: exploiting out-of-bag errors, boosting and random forests for effective automated classification. In: Proc. of SIGIR, pp. 353–362 (2015)
[30] Shtok, A., Kurland, O., Carmel, D.: Predicting query performance by query-drift estimation. In: Proc. of ICTIR, pp. 305–312 (2009)
[31] Shtok, A., Kurland, O., Carmel, D.: Using statistical decision theory and relevance models for query-performance prediction. In: Proc. of SIGIR, pp. 259–266 (2010)
[32] Tao, Y., Wu, S.: Query performance prediction by considering score magnitude and variance together. In: Proc. of CIKM, pp. 1891–1894 (2014)
[33] Zamani, H., Croft, W.B., Culpepper, J.S.: Neural query performance prediction using weak supervision from multiple signals. In: Proc. of SIGIR, pp. 105–114 (2018)
[34] Zhang, H.: The optimality of naive Bayes. AA **1**(2), 3 (2004)
[35] Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: Proc. of ECIR, pp. 52–64 (2008)
[36] Zhou, Y., Croft, W.B.: Query performance prediction in web search environments. In: Proc. of SIGIR, pp. 543–550 (2007)