Towards Privacy-Preserving Evaluation for Information Retrieval Models over Industry Data Sets

Peilin Yang[1], Mianwei Zhou[2], Yi Chang[3], Chengxiang Zhai[4], and Hui Fang[1]

[1] University of Delaware, USA
franklyn,hfang@udel.edu
[2] PlusAI, USA
mianwei@plus.ai
[3] Huawei Research America, USA
yichang@acm.org
[4] University of Illinois at Urbana-Champaign, USA
czhai@cs.uiuc.edu

Abstract. The development of Information Retrieval (IR) techniques heavily depends on empirical studies over real world data collections. Unfortunately, those real world data sets are often unavailable to researchers due to privacy concerns. In fact, the lack of publicly available industry data sets has become a serious bottleneck hindering IR research. To address this problem, we propose to bridge the gap between academic research and industry data sets through a privacy-preserving evaluation platform. The novelty of the platform lies in its "data-centric" mechanism, where the data sit on a secure server and IR algorithms to be evaluated would be uploaded to the server. The platform will run the codes of the algorithms and return the evaluation results. Preliminary experiments with retrieval models reveal interesting new observations and insights about state of the art retrieval models, demonstrating the value of an industry data set.

Keywords: test collections, privacy, evaluation

1 Introduction

Evaluation is essential in the field of Information Retrieval (IR). Whenever a new IR technique is proposed and developed, it needs to be evaluated and analyzed using multiple representative data collections. Since the beginning of the field, there have been a few community-based efforts on constructing evaluation collections for the IR research, such as TREC, NTCIR and CLEF. These collections are available for researchers to download, and the researchers can then conduct experiments on these data collections using their own computers. Such an evaluation methodology has been used by many researchers in thousands of publications.

Although TREC collections can provide valuable insights on how well an IR method performs, they are not the same data collections used by the search engine industry. Unfortunately, privacy is one of the reasons that prevent industry from sharing their data sets [10]. As a result, it remains unclear how well the observations we draw about an IR method based on the TREC collections can be generalized to the real world data sets used in the search engine industry.

One possible solution is to anonymize the data to protect privacy [1]. However, the data anonymization would lead to the loss of some useful information, and it would also pose constraints on the developed methods. Recently, a data-centric evaluation methodology, i.e., privacy-preserving evaluation (PPE), has been proposed [5, 3, 13]. This evaluation methodology does not require the sharing of the industry data set, which protects the privacy of the data. Instead, it advocates the industry to host an online evaluation system so that the researchers could upload their codes to evaluate their effectiveness over the industry data sets. The proposed PPE framework is also related to the studies on Evaluation-as-a-Service (EaaS) [6–8], where users can leverage the APIs provided by the system to fetch documents or to submit a ranking request.

This paper follows the idea of the privacy-preserving evaluation (PPE) framework[5], and presents a specific implementation of the framework, i.e., the *PPE-M* system. With the implemented system, we evaluate a few representative basic IR models over an industry data set and compare the results with those obtained on the standard TREC collections. Our study demonstrates that the PPE framework enables researchers to evaluate their methods using industry data collections, which essentially closes the gap between the IR researchers in academia and the industry data. Moreover, evaluation over the industry data set makes it possible to gain new insights about existing retrieval models. We focus on the evaluation of basic IR models in this paper, but the framework can be easily generalized for other tasks.

2 A General Framework of Privacy-Preserving Evaluation

Traditional IR evaluation methodology often requires a data collection to be downloaded to a local computer that also stores the code of an IR algorithm. After downloading the data, we can then run the code, get the results on the data collection and conduct further analysis. Clearly, this methodology would not work well for industry data sets which are not publicly available.

To address this limitation, a data-centric based privacy- preserving evaluation framework has been recently proposed [5]. The basic idea is to keep a data collection securely stored on its own server while allowing researchers to upload their codes to the server. The codes can access the statistics of the data collection through some pre-defined strategies, and then will be executed on the host server. The results will be returned to the researchers for further analysis.

There could be many different ways to implement such a general framework. In particular, we propose 3 different levels of support for evaluation that can accommodate different trade-offs. The main idea is illustrated in Figure 1. The

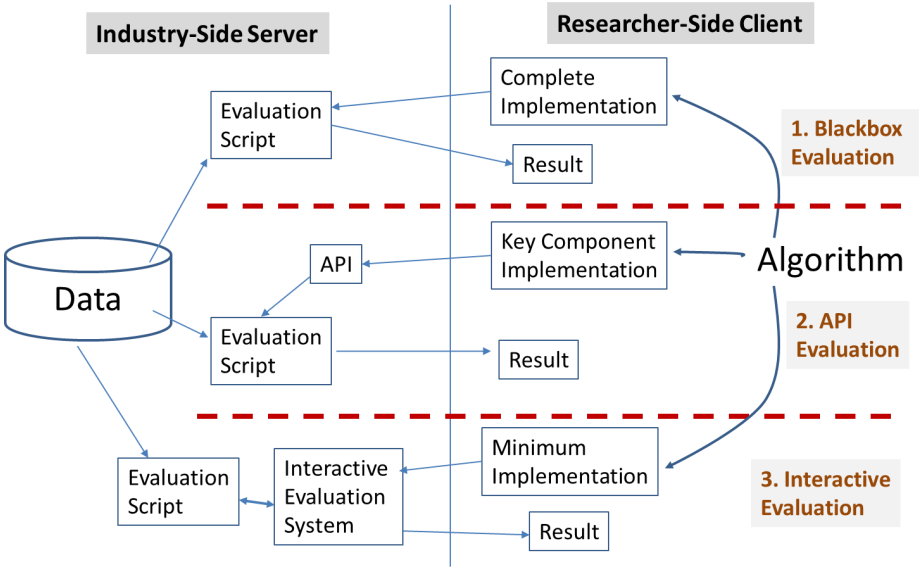


Fig. 1: Three-level Support for PPE

top level requires most work from the researcher's side but is most general as it can support evaluation of any algorithm in any language. Users of this kind of system have more access to the underlying system than other kinds of systems. For example, they could know how the index is built and thus are free to play with the index in order to better fulfill the ranking task. The middle level provides API support, so the researcher can focus on implementing just the key component of the algorithm to be evaluated, but it requires the researcher to use a particular API. The low level provides an interactive evaluation Web interface and attempts to minimize a researcher's work, but the algorithm that can be supported in this way may also be limited by the code that can be "opened up" by the system. An interactive system won't be able to provide full API support, so this would limit the algorithms that can be implemented and evaluated. It is clear that the top level is most general to support any algorithm, while the low level is most advanced with minimum effort on users, but has restriction on the algorithms to be evaluated.

We have already implemented the top level and the middle level, and will try to implement the low level in the future. Since the top level is trivial to implement, we focus on explaining how to implement the middle level in the next section.

3 A Specific Implementation

We now describe our implementation of the previously described privacy-preserving evaluation (PPE) framework. The implemented system focuses only on the evaluation of basic IR models, and is referred to as *PPE-M*.

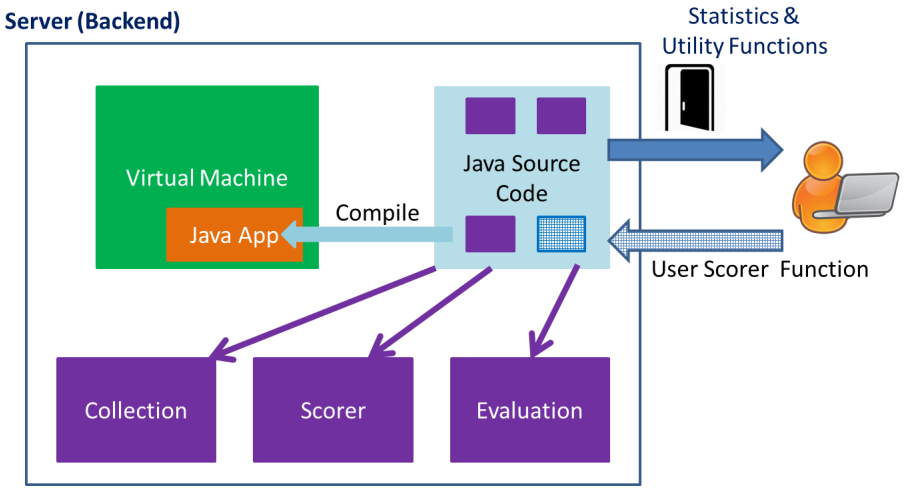


Fig. 2: System Architecture

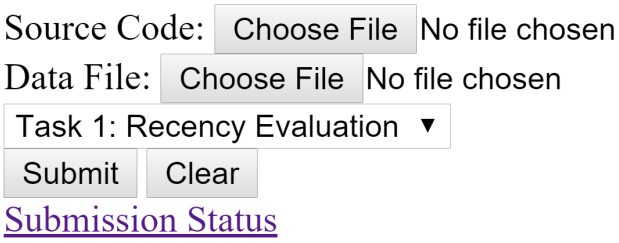


Fig. 3: Screenshot of code submission interface

PPE-M is a web service with a typical client/server architecture. It hosts data collections on the server, and enables users to implement and submit their codes of retrieval models. Figure 2 shows the architecture of the implemented system. Once a code is uploaded to the server, it will be executed to retrieve documents from the collections. The retrieval results will be evaluated at the back end, the evaluation results based on standard measures such as MAP will be returned to the user for each data collection.

The front end of the system is a web form, which allows the users to upload a Java source code file that includes the implementation of a retrieval function. The screenshot of the code submission interface is shown in Figure 3. Users can first select which task to participate, and then upload the source code as well as the data file (if necessary). The task could be ad hoc retrieval task, recency-based retrieval task, etc. The source code includes the implementation of a retrieval function in Java. The data file is optional, and it could include some prior information. Since the code will be executed on the server, it has to follow some conventions on accessing the collection statistics and calling external functions. To help users get familiar with these conventions, a user guide and

Code Id	Task Id	Data Attached	Submitted Time	Status	Message
2538	0	false	2016-02-10 14:27:32	FAILED	Compilation Error.
2537	0	false	2016-02-10 14:27:32	FINISHED	ndcg3:0.7145 map:0.8397
2536	0	false	2016-02-10 14:17:49	FINISHED	ndcg3:0.7145 map:0.8397
2535	0	false	2016-02-10 14:17:49	FINISHED	Exception in thread "main" java.lang.NullPointerException

Fig. 4: Screenshot of the result page

example codes are provided. The user guide includes a list of currently supported collection statistics that can be accessed by the code as well as a list of utility functions that the code can call. The restrictions posed on the codes are to prevent potential malicious attack from outsiders while making it possible for users to leverage the provided statistics to evaluate their models.

The core component of the *PPE-M* system is the Java source code package as the "Server (Backend)" rectangle. It consists of several modules, and each of them is responsible for a functionality. The *Collection* module basically preprocesses the collection and build the index of the collection. Key processes include tokenization, stop words removal, stemming, etc. It also provides the APIs to interact with the index so that the index does not necessarily need to be exposed to the users. The *Scorer* module is the base module for all ranking models. Models that are implemented and uploaded by the users have to complete some key functions that are required by the base *Scorer* module (think about this as derivation in object oriented programming). The *Scorer* module is then compiled and the binary is executed in the virtual machines. After the code execution, the *Evaluation* module kicks in. It computes the relevance scores of documents, generates the ranking list for each document and finally evaluate the model for different metrics. Such a modularized architecture offers flexibility in the implementation of the framework since each module could be re-implemented and tested independently.

The system returns the evaluation results to users through an interface as shown in Figure 4. If the code can be compiled and executed correctly, the evaluation results will be returned, as shown in the last column. Otherwise, error messages will be displayed. Users are able to see how well their ranking models perform over each available data set. Moreover, they are able to see the evaluation results of codes submitted by others. In the future, we plan to further enhance the interface with a leader-board that sort and display all the submitted runs based on their performance.

The *PPE-M* system is implemented and designed in the above way for the following reasons. First, the system preserves the privacy of the industry data collections. The data collections are not distributed to users but are stored on the server. Users' code may only access certain type of information about the collection and use them to compute the relevance scores, but the collection infor-

Table 1: Statistics of Test Collections

Collections	IC	ROBUST04	WT2G	GOV2
#queries	3,274	250	50	150
avg(ql)	2.80	2.73	2.44	3.11
avg(idf(qt))	13.75	11.50	9.81	13.49
#documents	71,406	528,155	247,491	25,205,179
avg(dl)	583.44	467.55	1057.59	937.25

mation would not be passed to the client side. Second, the system is configurable based on the level of the privacy concerns about the data collections. For example, if more information can be released about a data collection, users can use the information in their codes or access more information from the evaluation results. Finally, the system can be easily generalized to evaluate other tasks in IR such as recency-based retrieval and click-prediction.

4 Experiments

We evaluate *PPE-M* using an industry data set, which contains 3,274 news-related queries and 71,406 articles. The queries were collected across a few months. For each query, around 20 documents are selected from all the news articles based on the ranking produced by a very simple retrieval method. For each query, editors manually assign each document with a relevance label (1-Bad, 2-Fair, 3-Good, 4-Excellent). In particular, we focus on using the industry data set to verify observations about basic retrieval models that people have made previously on TREC data sets. As the results will show, the new data set is useful since it can provide new insights on existing retrieval models.

4.1 Experiment Design

We denote the industry data set described earlier as *IC*. In addition to this data set, we also report results on a few representative TREC collections: *ROBUST04*, *WT2G* and *GOV2*. These data sets are selected to cover different types and sizes of the collections. The statistics of all the collections are summarized in Table 1.

We compare three representative retrieval functions: (1) Okapi BM25 (**BM25**) [9]: a function derived from the classic probabilistic model; (2) Pivoted document length normalization (**Piv**) [11]: a function derived from the vector space model; and (3) F2EXP (**F2EXP**) [4]: a function derived using axiomatic approaches. These three functions are selected because they are among the most effective retrieval functions based on the evaluation over multiple TREC collections. *F2EXP*, in particular, has been shown to be more robust than existing retrieval functions with comparable optimal performance. The main difference between *F2EXP* and other retrieval functions lies in its different implementation of IDF and document length normalization parts.

Table 2: Optimal Performance Comparison (MAP). Optimal parameter settings are reported in parenthesis.

Model	IC	ROBUST04	WT2G	GOV2
BM25	0.8687 (0.35)	0.2478 (0.20)	0.3152 (0.15)	0.2970 (0.35)
PIV	0.8693 (0.20)	0.2206 (0.05)	0.2945 (0.05)	0.2536 (0.05)
F2EXP	0.8595 (0.00)	0.2512 (0.30)	0.2973 (0.25)	0.2828 (0.25)

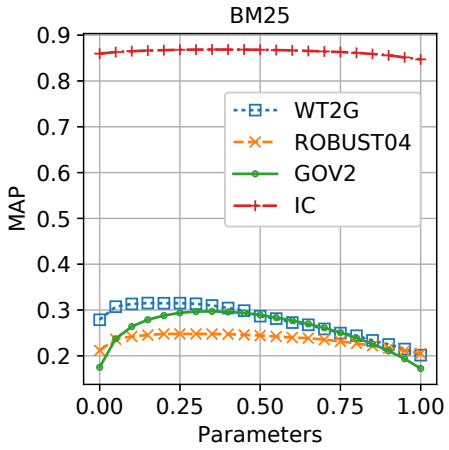


Fig. 5: Parameter Sensitivity (MAP) for BM25

4.2 Retrieval Performance Comparison

We compare the optimal performance of the retrieval functions over all the data sets and summarize the results in Table 2. Figure 5, 6 and 7 show the parameter sensitivity curves.

In general, the results on the *IC* data set are consistent with those on the TREC collections. Specifically, the optimal performance of the three functions are comparable. The optimal parameters are also within the reasonable range as mentioned in the previous study [2]. However, there are also a few new interesting observations that we can make based on the results from the industry data set.

The first interesting observation is that the evaluation results on the *IC* data set are much higher than on the TREC collections. This is not surprising since the *IC* data set is constructed by pooling top ranked documents for each query based on a simple ranking method while the documents of TREC collections are selected independently to the queries. Since most Web search engines now adapt a multi-level ranking strategy [12], the *IC* data set actually represents a more realistic problem setup.

The second interesting observation is about the *F2EXP* function. Although *F2EXP* has been shown to be robust in terms of the parameter values, its optimal

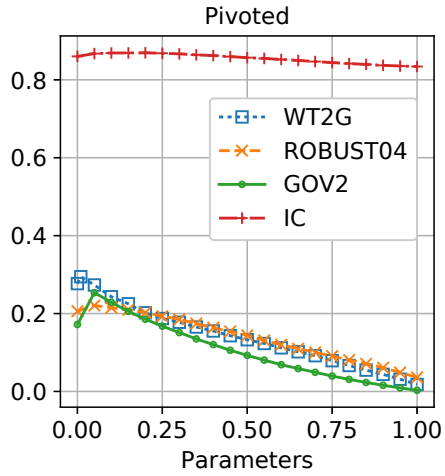


Fig. 6: Parameter Sensitivity (MAP) for Pivoted

parameter value is always larger than 0. However, its optimal parameter value is equal to 0 for the *IC* data set, which indicates that its length normalization part is not very effective. This is something that we have never observed based on the results for TREC collections.

4.3 Further Analysis

So far, we have demonstrated that the *PPE-M* system is able to evaluate retrieval models without releasing the data set. One new discovery made using this data set is about the "unusual" optimal parameter value in the *F2EXP* function.

To look into the reason behind this observation, we conduct more analysis using diagnostic evaluations [2]. The diagnostic evaluation methodology was proposed to identify the weaknesses and strengths of retrieval functions based on the perturbation of collections [2]. Each perturbation is designed to test a specific aspect of a retrieval function. Some perturbations can be done by changing simple statistics, while others may require additional information about the collections. In this paper, we only apply perturbation tests that can be implemented using the available statistics provided by the *PPE-M* system. These tests include two length variance sensitivity tests, one term noise resistance test and three TF-LN balance tests.

Figure 8 shows the perturbation results, split by the type of the perturbation. Here we choose three types of the perturbation, namely LV1, TN1 and TG3. For each type, we compare its results for *IC* data set with that of *GOV2* data set. We only show the results of the tests that are different on the two sets, so that we can focus on new insights gained by using the industry data set.

The first perturbation test is the length variance reduction test (LV1). We prefer curves that are lower because they indicate the functions would have more gain on length normalization part. Two plots on the first column in Figure 8a

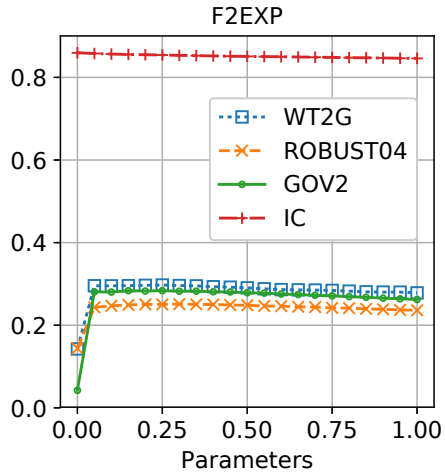


Fig. 7: Parameter Sensitivity (MAP) for F2EXP

indicate that *F2-EXP* has less gain on length normalization part for the *IC* data set, which is something we fail to observe from the TREC data set.

The second test is the term noise resistance test (TN). The curves that are higher means that they penalize long documents more appropriately. The plots on the second column in Figure 8b suggest that *F2-EXP* did a poor job to penalize long documents with more noisy terms on the *IC* data set. However, such a trend is not clear based on the results from the TREC data set.

The third test is the all query term growth test (TG3). We prefer curves that are higher since it means the corresponding function can balance TF and LN more appropriately. The last column in Figure 8c indicates that *F2-EXP* did a better job to avoid over-penalize long documents with more query terms on the *IC* data set. And this is something we can not see based on the results on the TREC data set.

In summary, our preliminary study has demonstrated the possibility of using the implemented *PPE-M* to evaluate IR models using a real world industry data set. More interestingly, we are able to gain new insights about existing retrieval functions by using the industry data.

5 Conclusions and Future Work

The paper addresses the issue of how to evaluate IR basic models with industry data sets. We present a specific implementation of the privacy-preserving evaluation framework, and conduct experiments on a real industry data set. This data set is usually not available for researchers outside the lab to use. With the implemented *PPE-M* system, we can make the data set available to outside researchers (still with limited access of course, but sufficient for some experimentation). Finally, we demonstrate that evaluating IR models on the industry data

set is useful for IR research. Because of the availability of this data set, we are able to make some new discoveries that would otherwise be impossible to make.

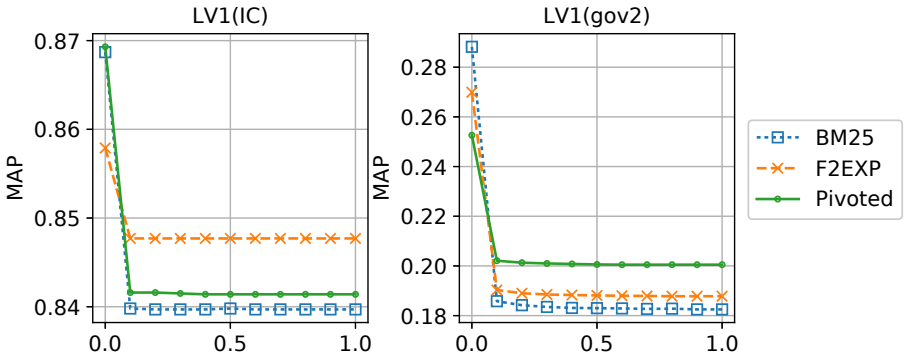
There are a few interesting future directions. First, we plan to make the implemented system publicly available as a new open-sourced IR evaluation platform. All IR researchers are welcome to use the system to evaluate their models. Second, we plan to extend the functionality of this system to support more IR tasks.

Acknowledgments. This research was supported by the U.S. National Science Foundation under IIS-1423002.

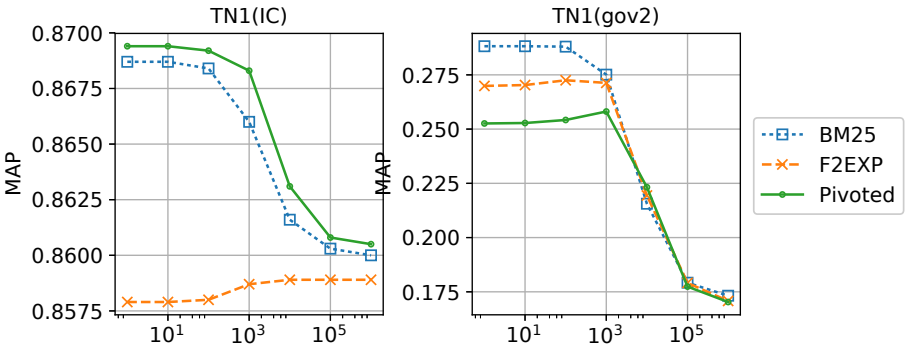
References

1. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. Journal of the ACM (JACM) 45(6), 965–981 (1998)
2. Fang, H., Tao, T., Zhai, C.: Diagnostic evaluation of information retrieval models. ACM Trans. Inf. Syst. 29(2), 7:1–7:42 (Apr 2011), http://doi.acm.org/10.1145/1961209.1961210
3. Fang, H., Wu, H., Yang, P., Zhai, C.: Virlab: A web-based virtual lab for learning and studying information retrieval models. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1249–1250. SIGIR '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2600428.2611178
4. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of the SIGIR'05 (2005)
5. Fang, H., Zhai, C.: Virlab: A platform for privacy-preserving evaluation for information retrieval models. In: Proceeding of the 1st International Workshop on Privacy-Preserving IR: (2014)
6. Hopfgartner, F., Hanbury, A., Müller, H., Kando, N., Mercer, S., Kalpathy-Cramer, J., Potthast, M., Gollub, T., Krithara, A., Lin, J., Balog, K., Eggel, I.: Report on the evaluation-as-a-service (eaas) expert workshop. SIGIR Forum 49(1), 57–65 (Jun 2015), http://doi.acm.org/10.1145/2795403.2795416
7. Lin, J., Efron, M.: Evaluation as a service for information retrieval. SIGIR Forum 47(2), 8–14 (Jan 2013), http://doi.acm.org/10.1145/2568388.2568390
8. Paik, J.H., Lin, J.: Retrievability in api-based "evaluation as a service". In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval. pp. 91–94. ICTIR '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2970398.2970427
9. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Proceedings of TREC (1996)
10. Si, L., Yang, H.: Privacy-preserving ir: when information retrieval meets privacy and security. In: Proceedings of the SIGIR'14 (2014)
11. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: Proceedings of the SIGIR'96 (1996)
12. Wang, L., Lin, J., Metzler, D.: Learning to efficiently rank. In: Proceedings of SIGIR'10
13. Yang, P., Fang, H.: A reproducibility study of information retrieval models. In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval. pp. 77–86. ICTIR '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2970398.2970415

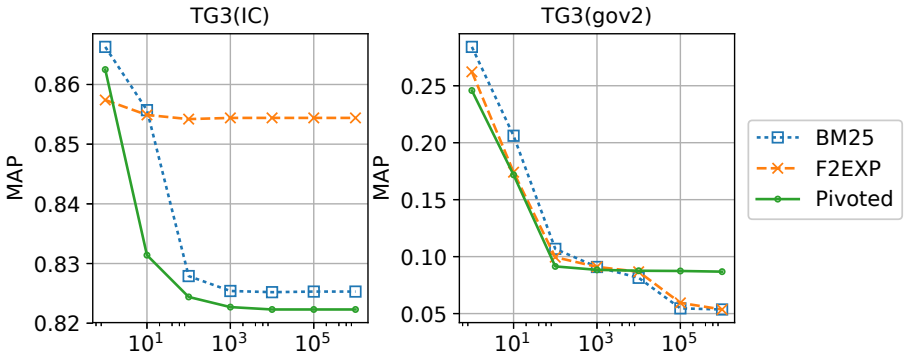
Fig. 8: Results of perturbation tests



(a) Perturbation analysis of LV1



(b) Perturbation analysis of TN1



(c) Perturbation analysis of TG3