Improving Ranking Evaluation Employing Visual Analytics

Marco Angelini[2], Nicola Ferro[1], Giuseppe Santucci[2], and Gianmaria Silvello[1]

[1] University of Padua, Italy
{ferro,silvello}@dei.unipd.it
[2] "La Sapienza" University of Rome, Italy
{angelini,santucci}@dis.uniroma1.it

Abstract. In order to satisfy diverse user needs and support challenging tasks, it is fundamental to provide automated tools to examine system behavior, both visually and analytically. This paper provides an analytical model for examining rankings produced by IR systems, based on the discounted cumulative gain family of metrics, and visualization for performing failure and "what-if" analyses.

1 Introduction

Information Retrieval (IR) systems, ranging from World Wide Web search engines to enterprise search or expertise retrieval systems and passing through information access components in wider systems such as digital libraries, are key technologies to get access to relevant information items in a context where information overload is day-to-day experience of every user.

In order to present this considerable amount of information to the user, IR systems rely on sophisticated ranking models where many different parameters affect the obtained results. Furthermore, they are comprised of several components interacting together in complex ways to produce a list of relevant documents in response to a user query. Ranking is a central and ubiquitous issue in this context since it is necessary to return the results retrieved in response to a user query according to the estimation of their relevance to that query. The interactions among the components of an IR system are often hard to trace down, to explain in the light of the obtained results, and to interpret in the perspective of possible modifications to be made to improve the ranking of the results, thus making this activity extremely difficult. This activity is usually called, in the IR field, *failure analysis* and it is deemed a fundamental activity in experimental evaluation even if it is too often overlooked due to its difficulty [1].

To give the reader an idea of how much demanding failure analysis can be, please consider the case of the the Reliable Information Access (RIA) workshop [4], which was aimed at investigating in a systematic way the behaviour of just one component in a IR system, namely the relevance feedback module. [4] reported that, for analysing 8 systems, 28 people from 12 organizations worked for 6 weeks requiring from 11 to 40 person-hours per topic for 150 overall topics.

P. Forner et al. (Eds.): CLEF 2013, LNCS 8138, pp. 29–40, 2013.

Such a big effort was just aimed at understanding why a system behaved in a certain way. Nevertheless, in a real setting, after such inspection, you have to come back to design and development and implement the modifications and new features that the previous analysis suggested as possible solutions to the identified problems and, then, you have to start a new experimentation cycle to verify whether the newly added features actually give the expected contribution. Therefore, the overall process of improving an IR system is much more time and resource demanding than failure analysis alone.

The contribution of the paper is the design, implementation, and initial test of a Visual Analytics (VA) system, called Visual Analytics Tool for Experimental Evaluation (VATE^2), which supports all the phases of the evaluation of an IR system, namely performance and failure analysis, greatly reducing the effort needed to carry them out by providing effective interaction with the experimental data. Moreover, VATE^2 introduces a completely new phase in the experimental evaluation process, called *what-if analysis*, which is aimed at getting an estimate of what could be the effects of a modification to the IR system under examination before needing to actually implement it.

The paper is organized as follows: Section 2 discusses related work. Section 3 describes how the analytical models for interaction we adopt to conduct failure analysis and what-if analysis. Section 4 explains how the visualization and interaction part works and gives and overview of VATE^2 and Section 5 presents an initial evaluation of the system conducted with experts of the field. Finally, Section 6 concludes the paper, pointing out ongoing research activities.

2 Related Work

The graded-relevance metrics considered in this paper are based on cumulative gain [5]; the Discounted Cumulated Gain (DCG) measures are based on the idea that documents are divided in multiple ordered categories, e.g. highly relevant, relevant, fairly relevant, not relevant. DCG measures assign a gain to each relevance grade and for each position in the rank a discount is computed. Then, for each rank, DCG is computed by using the cumulative sum of the discounted gains up to that rank. This gives rise to a whole family of measures, depending on the choice of the gain assigned to each relevance grade and the used discounting function.

A work that exploits DCG to support analysis is [8] where the authors propose the potential for personalization curve. The potential for personalization is the gap between the optimal ranking for an individual and the optimal ranking for a group. The curves plots the average nDCG's (normalized DCG) for the best individual, group and web ranking against different group size. These curves were adopted to investigate the potential of personalization of implicit content-based and behavior features. Our work shares the idea of using a curve that plots DCG against rank position, as in [5], but using the gap between curves to support analysis as in [8]. Moreover, the models proposed in this paper provide the basis for the development of VA environment that can provide us with: (i) a quick and intuitive idea of what happened in a ranking list; (ii) an understanding of what are the main reasons of its perceived performances; and, (iii) the possibility of exploring the consequences of modifying the system characteristics through an interactive what-if scenario. The work presented here builds on a precedent work by the authors [1] refining the what-if model and introducing a validation with expert users.

3 The Models Behind VATE^2

3.1 Clustering via Supervised Learning

IR systems are seen as black boxes in experimental evaluation, because, in most cases, we can analyze the ranking lists produced by a system, but we cannot analyze the system which produced them. This means that we cannot modify a systems, run new and diversified tests to understand how the system behaves and how it can be improved. To this end we have to rely only on the outputted ranking lists and from these we need to infer how the system behave under specific conditions.

In this context machine learning based on supervised learning techniques can help because they are effective tools to automatically tune parameters and combine multiple evidences [6] and they can be employed starting from the rankings outputted by test systems. Supervised learning methods are feature-based and a widely-used list of features usually adopted by these techniques is described in [3].

The purpose of learning to rank techniques is to improve the original ranking model in order to obtain better performances or to grip on machine learning to build new and more effective ranking models. In VATE^2 we leverage on these techniques with a slightly different purpose; indeed, we use the produced ranking lists, the experimental collection and a machine learning algorithm (i.e. a classification algorithm based on regression trees) to learn a ranking model of a given IR system in order to throughly study it without actually having it available.

Most of the state-of-the-art learning to rank algorithms are "feature-based", which means that they learn the optimal way of combining features extracted from topic-document pairs. So, the topic-document pairs under investigation are represented as vectors of features, representing the relevance of documents w.r.t. a given topic. We can divide the typical features used in learning to rank into three main categories: document-based, topic-based, and model-based. Document-based features are extracted from the given document; topic-based features are the same as the document-based but calculated on the text of the topic, and model-based features are the output of ranking models. In VATE^2 we adopt document-based and topic-based features and we do not consider the model-based ones. This choice derives from the fact that our goal is to learn the ranking model of a system in the most reliable way and not to improve their performances. The most used and reliable list of features used in learning to rank framework is provided by the LEarning TO Rank (LETOR)[1] initiative run by Microsoft Research and proposed by Liu et al. in [7].

In this work we exploit this framework to learn the ranking model of the IR system under investigation in order to simulate the way in which it ranks the documents. Our aim is to support a "what if" investigation on the ranking list outputted by the system taken into account; the basic idea is to show how the ranking list and the DCG change when we move upward or downward a document in the list. To this purpose, the "cluster hypothesis" saying that "closely associated documents tend to be relevant to the same requests" [9] has to be taken into account; indeed, there can be a correlation in the ranking list between a document and its "closed associated documents". We lever on the hypothesis that if we change the rank of a document also the cluster of documents associated with it will accordingly change their rank.

There are several algorithms for clustering as described in [2]. In this work we focus on the ranking of the considered documents and on how the ranking model can be improved. To this purpose we form the cluster for a target document by grouping together the documents which are similar from the considered ranking model point-of-view. Let us take into account a full result vector FV_j retrieved for a given query q_j, for each document $FV_j[i]$ we create a cluster of documents C_i by: (i)) employing a test IR system and submitting $FV_j[i]$ as a query, thus retrieving a result vector FV_i of documents; (ii) determining $C_i = FV_j \cap FV_i$; and, (iii) ranking the documents in C_i by employing the learned ranking model.

Therefore, we retrieve a result vector FV_i of relevant documents w.r.t. $FV_j[i]$, then we pick out only those documents which are in the original result vector (say FV_j), and lastly we use the learned ranking model to order these documents accordingly to their "ranking" similarity to $FV_j[i]$. In this way, the higher a document is into the cluster C_i, the more similar it is to the target document $FV_j[i]$. We can see that the similarity measure is based on how the documents are seen by the learned ranking model.

In the end of this process, for each document $FV_j[i]$ obtained by an IR system for a query q_j, we define a cluster of documents C_i ordered by their relevance with respect to $FV_j[i]$.

3.2 Rank Gain/Loss Model

According to [5] we model the retrieval results as a ranked vector of n documents V, i.e. $V[1]$ contains the identifier of the document predicted by the system to be most relevant, $V[n]$ the least relevant one. The ground truth GT function assigns to each document $V[i]$ a value in the relevance interval $\{0..k\}$, where k represents the highest relevance score. Thus, the higher the index of a relevant document the less useful it is for the user; this is modeled through a discounting function DF that progressively reduces the relevance of a document, $GT(V[i])$ as i increases. We do not stick with a particular proposal of DF and we develop

[1] http://research.microsoft.com/en-us/um/beijing/projects/letor/

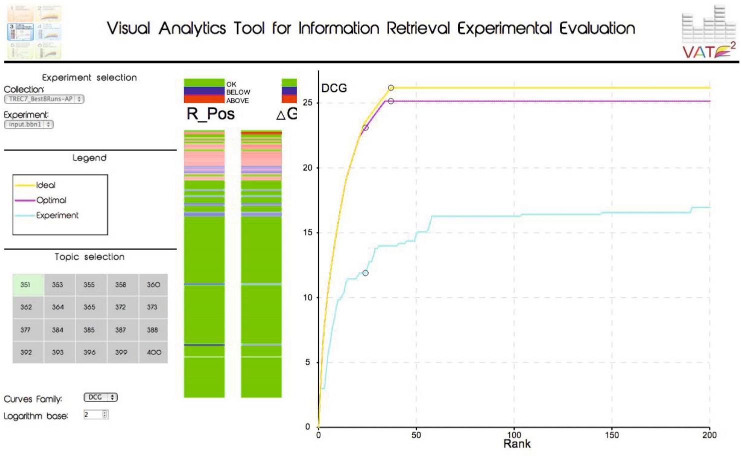

Fig. 1. A Screen-shot of the failure analysis interface of VATE[2]

a model that is parametric with respect to this choice. However, to fix the ideas, we recall the original DF proposed in [5]:

$$DF(V[i]) = \begin{cases} GT(V[i]), \ if \ i \leq x \\ GT(V[i])/\log_x(i), \ if \ i > x \end{cases} \tag{3.1}$$

that reduces, in a logarithmic way, the relevance of a document whose index is greater than the logarithm base.

The DCG function allows for comparing the performances of different IR systems, e.g. plotting the DCG(i) values of each IR system and comparing the curve behavior. However, if the user's task is to improve the ranking performance of a single IR system, looking at the misplaced documents (i.e. ranked too high or too low with respect to the other documents) the DCG function does not help, because the same value DCG(i) could be generated by different permutations of V and because it does not point out the loss in cumulative gain caused by misplaced elements. To this end, we introduce the following definitions and novel metrics.

Using the above definitions we can define the relative position $R\_Pos(V[i])$ function for each document in V as follows:

$$R\_Pos(V[i]) = \begin{cases} 0, \ if \ min\_index(V, GT(V[i])) \leq i \leq max\_index(V, GT(V[i])) \\ min\_index(V, GT(V[i])) - i, \ if \ i < min\_index(V, GT(V[i])) \\ max\_index(V, GT(V[i])) - i, \ if \ i > max\_index(V, GT(V[i])) \end{cases} \tag{3.2}$$

$R\_Pos(V[i])$ allows for pointing out misplaced elements and understanding how much they are misplaced: 0 values denote documents that are within the optimal interval, negative values denote elements that are below the optimal interval (pessimistic ranking), and positive values denote elements that are above the optimal (optimistic ranking). The absolute value of $R\_Pos(V[i])$ gives the minimum distance of a misplaced element from its optimal interval.

According to the actual relevance and rank position, the same value of $R\_Pos(V[i])$ can produce different variations of the DCG function. We measure the contributions of misplaced elements with the function $\Delta\_Gain(V, i)$ which quantifies the effect of a misplacement in the overall computation of DCG. The $\Delta\_Gain(V, i)$ function can assume both positive and negative values, where negative values correspond to elements that are presented too early (with respect to, their relevance) to the user and positive values to elements that are presented too late.

3.3 What-if Analysis Model

The clusters of documents defined above play a central role in the document movement estimation of VATE^2. Indeed, once a user spots a misplaced document, say d_4, and s/he decides to move it upward or downward, also the ten documents in the C_4 cluster are moved accordingly. The current implementation of VATE^2 employs the simple linear movement strategy where the movement of the document and the related document cluster happens according to a straightforward algorithm that tries to move the documents in the cluster of the same amount of positions as the document dragged and dropped by the user. However, this is not always possible since, for example, a document in the cluster might be ranked higher than the document selected by the user and may not exist enough space on the top of the ranking to place it; in this and similar cases, the movement algorithm "compresses" the movement of the documents in the cluster, approximating at its best the user intent.

The retrieval results are modeled as a ranked vector V containing the first 200 documents of the full result vector FV. The clustering algorithm we described, associates to each document $V[i]$ a cluster C_i of similar documents (we consider only the documents whose relevance with $V[i]$ is greater than a suitable threshold). Moreover, for the sake of notation we define the index cluster set IC_i, i.e., the set of indexes of FV corresponding to elements in C_i: $IC_i = \{j|FV[j] \in C_i\}$. As a consequence, according to the "cluster hypothesis", moving up or down the document $V[i]$ will affect in the same way all the documents in C_i and that might result in rescuing some documents below the 200 threshold pushing down some documents that were above such threshold.

We model the what-if interaction with the system with the operator $Move(i, j)$ whose goal is to move the element in position i in position j. In order to understand the effect on V of such an operation, we have to consider all the C_i elements and the relative position of their indexes, that ranges between $min(IC_i)$ and $max(IC_i)$. Different cases may occur and we analyze them assuming, without loss of generality, that $i < j$, i.e., that the analyst goal is to move up the element $V[i]$ of $j - i$ positions. For the clustering hypothesis that implies that all the C_i elements will move up of $j - i$ positions as well. There are, however, situations in which that is not possible: the maximum upshift is $max(min(IC_i) - 1, j - i)$ and if $j - i > min(IC_i) - 1$ the best we can do is to move up all the C_i elements of just $IC_i - 1$ positions. That corresponds to the situation in which the analyst wants to move up the element in position i of k positions, but there exists a document in C_i whose index is $\leq k$ and, obviously, it is not possible to move it up of k positions. In such a case, the system moves up all the documents in the cluster of $min(IC_i) - 1$ positions, approximating the user intent.

4 Overview of VATE^2

VATE^2 allows the analyst to perform three main activities: performance analysis, failure analysis and what-if analysis by employing the models described above. These three main activities can be carried out at the "topic level" or at the "experiment level".

At the topic level VATE^2 takes as input the ranked document list for the topic t and the ideal ranked list, obtained choosing the most relevant documents in the collection D for the topic t and ordering them in the best way. At the experiment level VATE^2 evaluates the overall quality of the ranking for all the topics of the experiment, focusing on the variability of the results. Basically, at the experiment level VATE^2 shows an aggregate representation based on the boxplot statistical tool showing the variability of the DCG family of metrics calculated on all the topics considered by an experiment. In this way the analyst will have a clearer insight on what to expect from her/his ranking algorithm both in a static way and in a dynamic one (which involves an interactive reordering of the ranked list of documents).

While visually inspecting the ranked list (i.e. failure analysis), it is possible to simulate the effect of interactively reordering the list, moving a target document d and observing the effect on the ranking while this shift is propagated to all the documents of the cluster containing the documents similar to d (i.e. what-if analysis). This cluster of documents simulates the "domino effect" within the given topic t. When the analyst is satisfied with the results, i.e. when he has produced a new ranking of the documents that corresponds to the effect that is expected by modifications that are planned for the system, he can feed the Clustering via Supervised Learning model with the newly produced ranked list, obtain a new model which takes into account the just introduced modifications, and inspecting the effects of this new model for other topics. This re-learning phase simulates the "domino effect" on the other topics different from t caused by a possible modification in the system.

4.1 How to Perform the Failure Analysis

Figure 1 shows the DCG Graph for the topic level analysis. On the left side we can see two vertical bars representing the visualization of the ranking list. The first one represents the R_Pos vector. The visualization system computes the optimal ranking list of the documents and assigns to each document a color based on its rank. A green color is assigned to a document at the correct rank w.r.t. the calculated optimal rank; whereas a blue color is assigned to a document ranked below the optimal and a red color is assigned to a document ranked above the optimal. The color intensity gives the user an indication of how far the document is from its optimal rank: a weak intensity means that the document is close to the optimal, a strong intensity means it is far to the optimal. The second vertical bar represents the *Δ_Gain* function values for each document. We adopted the same color code as in the previous vector, but in this case the red color represents a loss and a blue color represents a gain in terms of *Δ_Gain*.

On the right side of Figure 1 we can see a graph showing three curves:

Experiment Ranking refers to the top n ranked results provided by the system under investigation;
Optimal Ranking refers to an optimal re-ranking of the experiment;
Ideal Ranking refers to the ideal ranking of the top n documents in the pool.

The visualization system is built in such a way that if a user selects a document in the *R_Pos* vector, also the DCG loss/gain in the *Δ_Gain* vector and all its contributions to the different curves (i.e. Experiment, Optimal and Ideal) will be highlighted.

The visualization described so far is well-suited to cope with a static analysis of the ranked result: the user can understand if there is the need to re-rank the documents or to perform a re-querying to retrieve a different set of documents with the aim of obtaining a better value of the DCG metric.

4.2 How to Perform the What-if Analysis

The what-if functionality allows the users to interact with the ranked vector of *R_Pos*. The system allows the user to shift a target document t from its actual position to a new one in a "drag&drop" fashion, with the goal of investigating the effect of this movement in the ranking algorithm by inspecting the DCG of the modified ranking list. Clearly, a change in the ranking algorithm will affect not only the target document t, but also all the documents in its cluster.

In Figure 2 it is possible to see the animated phase of interactive re-ranking of the documents at the topic level: after highlighting and moving the target document t from the starting position to a new one, the user will be presented with an animated re-ranking of the documents connected to the target one. Once the new position of the target document has been selected, the system moves it to the new position and the documents in its associated cluster are moved together into their new positions. This leads to the redrawing of the *R_Pos*, *Δ_Gain* and DCG graphs according to the new values assigned to each document involved in the ranking process.

It is possible to see that when a user select a document in the leftest bar, all the documents in its cluster are highlighted in yellow helping the user to understand which documents are involved in a potential movement.

Figure 2 shows also the result of the what-if process: the image presents two new curves, representing the new values assigned for both the experiment curve (purple one) and the optimal curve (orange one). To evaluate the changes in the DCG function, the image shows, in a dash-stroke fashion, the old curve trends. Thanks to this visualization, the user can appreciate the gain or the

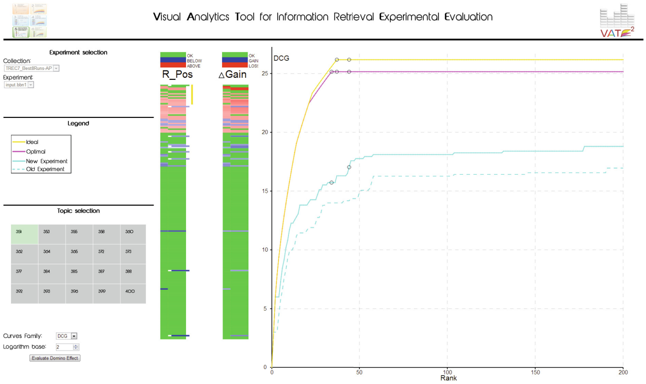

Fig. 2. A Screen-shot of the topic level what-if analysis interface of VATE^2

loss obtained from this particular re-rank. In the case shown in Figure 2 the movements performed by the user improved the performances at the topic level; indeed, the dashed line – i.e. the old experiment curve – is lower than the solid one – i.e. the new experiment curve. This means that we are simulating a change in the system that does improve it. On top of that, at the experiment level, the change in the ordering of a particular ranking list will result in changing also the other ranking lists within the same experiment: these changes can be intercepted by this graph in terms of variability of the curves and on the raising/declining of the "box" region of the boxplots (showed as filled area in the graph).

To maintain the graph as clear as possible, the choice of not representing the single boxplots, but simply the continuous lines joining the similar points has been taken. So, in the graph area there are five different curves which are: upper limit, upper quartile, median, lower quartile, and lower limit. All these curves are determined for the ideal, the optimal and the experiment cases. For each case, the area between lower and upper quartile is color filled in order to highlight the central area (the box of the boxplot) of the analysis.

In figure 3 we can appreciate that, in this particular case, the optimal and experiment areas do not overlap very much, and the median curve of the experiments is quite far from the one of the optimal. This can be asserted from an aggregate point of view, and not by a specific topic analysis like the one we proposed with the DCG graph. Different considerations can also be made on variability: in this case, while experiment and optimal box areas are quite broad, demonstrating a heterogeneity in values, and also the ideals box area is big meaning a high variability of the data among the different topics.

The domino effect due to the what-if analysis is highlighted by the experiment areas: the old one (before the what-if analysis) is shaded in blue, whereas the new one (after the what-if analysis) is shaded in green. We can see that a change in one topic at the topic level worsens the global performances; indeed, the blue area is better than the green one. This means that the change the user did at the

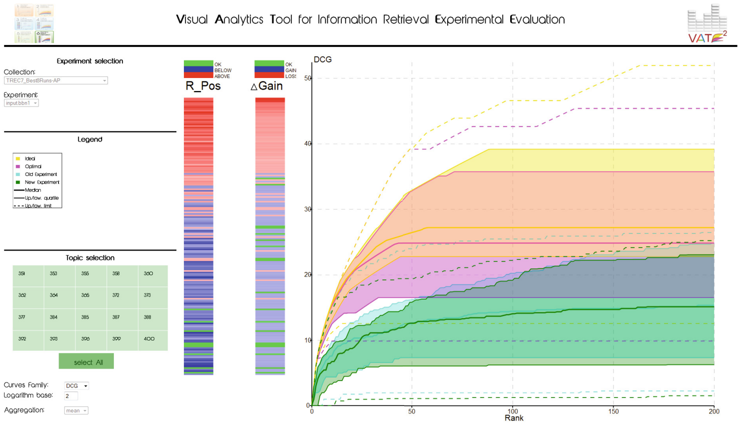

Fig. 3. A Screen-shot of the experiment level what-if analysis interface of VATE^2

topic level (which improved the local performances) reflects at the experiment (global) level worsening the overall performances of the system.

5 Initial Validation with Experts

VATE^2 has been tested in a laboratory setting involving 13 experts (i.e. academics, post-docs and PhD students) in IR. The functioning of VATE^2 was described by means of an oral presentation where its peculiar functions were explained. This introduction was necessary to get the experts to know the system and to let them understand how to use it. The performance analysis part as well as the failure analysis one are more straightforward and close to the day to day experience of the experts; whereas, the what if analysis evaluation represents a totally new paradigm which requires some time to be properly understood.

The study was conducted by allowing the experts to freely use VATE^2 for an hour and, at the end, by asking them to compile a questionnaire. The questionnaire was divided into seven parts, one for each interface and one for an overall evaluation of VATE^2 as a whole. Every part repeated the following seven questions referring to the specific functionality under evaluation:

Q1. Is the addressed problem relevant for involved stakeholders (researchers and developers)?
Q2. Are the currently available tools and techniques adequate for dealing with the addressed problem?
Q3. Do currently available tools and techniques for dealing with the addressed problem offer interactive visualizations?
Q4. Is the proposed visual tool understandable?

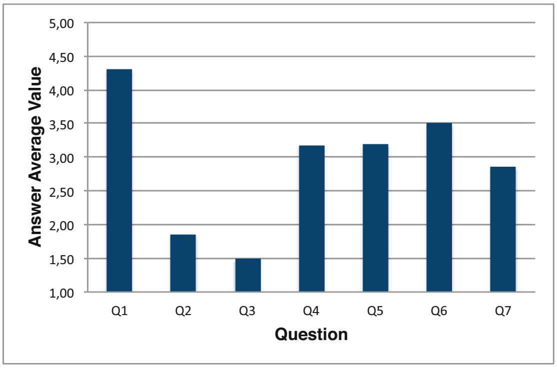

Fig. 4. The histogram reporting the average answers of the experts evaluating VATE^2 as a whole

Q5. Is the proposed visual tool suitable and effective for dealing with the addressed problem?

Q6. To what extent the proposed visual tool is innovative with respect to the currently available tools and techniques?

Q7. To what extent the proposed visual tool will enhance the productivity of involved stakeholders (researchers and developers)?

The first three questions regard the scientific relevance of VATE^2 and they are aimed to understand if the experts think the problem addressed is relevant and if there exist other tools with the same purpose. The last four questions are aimed to understand if the experts think VATE^2 is useful for experimental evaluation and if it is well-suited for its purposes. Every answer was graded from 1 to 5, where 1 stand for "not at all" and 5 for "quite a lot". In Figure 4 we report the average results of the questionnaire regarding the overall part which allows us to understand what the experts think about VATE^2 as a whole.

We can see that the problem addressed is of high relevance for the involved stakeholder (question 1) and that there not exist any other tool doing the work of VATE^2. Indeed, answers to questions 2 and 3 are both below 2 as an average value which means that VATE^2 proposes something totally new in the field. Questions 4 to 6 report that the tool is understandable, suitable and effective for dealing with the addressed problem, and innovative. The last question is about productivity; on average the experts think VATE^2 can improve productivity but the answer is not clear like for the other questions. We think this is due to the time necessary to learn how to effectively use the system. By analyzing the results of every single part we see that experts think that VATE^2 improves productivity for performance analysis and failure analysis, but it is less clear if it is useful for what-if analysis which as explained above is a brand new topic in IR evaluation and probably it requires more time to become useful to the experts.

6 Conclusion and Future Work

This paper presented a fully-fledged analytical and visualization model to support interactive exploration of IR experimental results with a two-fold aim: (i) to ease and support deep failure analysis in order to better understand system behavior; (ii) to conduct a what-if analysis to have an estimate of the impact that possible modifications to the system, identified in the previous step and aimed at improving the performances, can have before needing to actually re-implement the system.

Future work will concern two main issues: (i) while the informal results about the system usage are quite encouraging we plan to run a more structured user study, involving people that have not participated in the system design; and (ii) we want to improve the way in which the clusters produced by the The Clustering via Supervised Learning methods are used to compute the new ranking and the associated DCG functions.

Acknowledgements. The work reported in this paper has been supported by the PROMISE network of excellence (contract n. 258191) project as a part of the 7th Framework Program of the European commission (FP7/2007-2013).

References

1. Angelini, M., Ferro, N., Santucci, G., Silvello, G.: Visual Interactive Failure Analysis: Supporting Users in Information Retrieval Evaluation. In: Proc. of the 4th Information Interaction in Context Symposium, IIIX 2012, pp. 194–203. ACM, New York (2012)
2. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Kogan, J., Nicholas, C., Teboulle, M. (eds.) Grouping Multidimensional Data, pp. 25–71. Springer, Heidelberg (2006)
3. Geng, X., Liu, T.-Y., Qin, T., Li, H.: Feature Selection for Ranking. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) Proc. 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), pp. 407–414. ACM Press, New York (2007)
4. Harman, D., Buckley, C.: Overview of the Reliable Information Access Workshop. Information Retrieval 12(6), 615–641 (2009)
5. Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information System (TOIS) 20(4), 422–446 (2002)
6. Liu, T.-Y.: Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
7. Liu, T.-Y.Y., Xu, J., Qin, T., Xiong, W., Li, H.: LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. In: Joachims, T., Li, H., Liu, T.-Y., Zhai, C. (eds.) SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (2007)
8. Teevan, J., Dumais, S.T., Horvitz, E.: Potential for Personalization. ACM Transactions on Computer-Human Interaction (TOCHI) 17(1), 1–31 (2010)
9. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)