Visual Interactive Failure Analysis: Supporting Users in Information Retrieval Evaluation

Marco Angelini
Sapienza University of Rome, Italy
angelini@dis.uniroma1.it

Nicola Ferro
University of Padua, Italy
ferro@dei.unipd.it

Giuseppe Santucci
Sapienza University of Rome, Italy
santucci@dis.uniroma1.it

Gianmaria Silvello
University of Padua, Italy
silvello@dei.unipd.it

ABSTRACT

Measuring is a key to scientific progress. This is particularly true for research concerning complex systems, whether natural or human-built. Multilingual and multimedia information access systems, such as search engines, are increasingly complex: they need to satisfy diverse user needs and support challenging tasks. Their development calls for proper evaluation methodologies to ensure that they meet the expected user requirements and provide the desired effectiveness. In this context, failure analysis is crucial to understand the behaviour of complex systems. Unfortunately, this is an especially challenging activity, requiring vast amounts of human effort to inspect query-by-query the output of a system in order to understand what went well or bad. It is therefore fundamental to provide automated tools to examine system behaviour, both visually and analytically. Moreover, once you understand the reason behind a failure, you still need to conduct a "what-if" analysis to understand what among the different possible solutions is most promising and effective before actually starting to modify your system. This paper provides an analytical model for examining performances of IR systems, based on the discounted cumulative gain family of metrics, and visualization for interacting and exploring the performances of the system under examination. Moreover, we propose machine learning approach to learn the ranking model of the examined system in order to be able to conduct a "what-if" analysis and visually explore what can happen if you adopt a given solution before having to actually implement it.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Benchmarking, evaluation, methodology*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IIiX 2012, Nijmegen, The Netherlands
Copyright 2012 ACM 978-1-4503-1282-0/2012/08 ...$10.00.

General Terms

Design, Experimentation, Management, Measurement, Performance

Keywords

experimental evaluation, scientific data, evaluation infrastructure, data test collection, best practices

1. INTRODUCTION

Designing, developing, and testing an Information Retrieval (IR) system is a challenging task, especially when it comes to understanding and analysing the behaviour of the system under different conditions in order to tune or to improve it as to achieve the level of effectiveness needed to meet the user expectations.

Failure analysis is especially resource demanding in terms of time and human effort, since it requires inspecting, for several queries, system logs, intermediate output of system components, and, mostly, long lists of retrieved documents which need to be read one by one in order to try to figure out why they have been ranked in that way with respect to the query at hand.

To give the reader an idea of how much demanding it can be, please consider the case of the the Reliable Information Access (RIA) workshop [Harman and Buckley, 2009], which was aimed at investigating in a systematic way the behaviour of just one component in a IR system, namely the relevance feedback module. [Harman and Buckley, 2009] reported that, for analysing 8 systems, 28 people from 12 organizations worked for 6 weeks requiring from 11 to 40 person-hours per query.

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 just failure analysis.

Considering this, it is important to define new ways to help IR researchers, analysts and developers to understand the limits and strengths of the IR system under investigation. Visual analytics techniques can give assistance to this process by providing graphic tools which interacting with IR techniques may ease the work of the users.

The goal of this paper is to exploit a visual analytics approach to design a methodology and develop an interactive visual system

which support IR researchers and developers in conducting experimental evaluation and improving their systems by:

1. reducing the effort needed to conduct failure analysis;
2. allowing them to anticipate what the impact of a modification to their system could be before needing to actually implement it.

These two goals are described in more detail in the following section. The paper is organized as follows: Section 2 describes the overall approach and contributions of the proposed methods. Section 3 discusses related work. Section 4 describes how the analytical models for interaction we adopt to conduct failure analysis. Section 5 explains how the visualization and interaction part works, while Section 6 discusses the experimentation we conducted; it describes the experimental setup and provides some application examples over the used experimental data. Finally, Section 7 concludes the paper, pointing out ongoing research activities.

2. APPROACH AND CONTRIBUTIONS

2.1 Supporting Failure Analysis

As far as the first goal, i.e. failure analysis, is concerned, we introduce a ranking model that allows us to understand what happens when you misplace documents with different relevance grades in a ranked list, e.g. you put a marginally relevant document before a highly relevant document for your topic.

The proposed ranking model is able to quantify, rank by rank, the gain/loss obtained by an IR system with respect to both the ideal ranking, i.e. the best ranked list that can be produced for a given topic, and the optimal ranking, i.e. the best ranked list that can be produced using the documents actually retrieved by the system. The ranking model builds on the Discounted Cumulative Gain (DCG) family of measures [Järvelin and Kekäläinen, 2002, Keskustalo et al., 2008], which are designed to work with graded relevance and are well-suited both to quantify system performances and to give an idea of the overall user satisfaction with a given ranked list considering the persistence of the user in scanning the list.

Starting from the DCG measures, we introduce two functions: the relative position $R_{pos}(V[i])$, which quantifies how much a document has been misplaced with respect to its ideal (optimal) position, and the delta gain $\Delta\_Gain(V,i)$, which quantifies how much each document has gained/lost with respect to its ideal (optimal) DCG. On top of this ranking model, we propose a visualization, see for example Figure 1, where the DCG curves for the experiment ranking, the ideal ranking, and the optimal ranking are displayed together with two bars, on the left, representing the $R_{pos}(V[i])$ and $\Delta\_Gain(V,i)$ with a color code that allows us to easily spot problematic and misplaced documents. The user can then interact with the visualization and the underlying ranking model to explore the behaviour of the system.

Comparing the system ranking with respect to the ideal and optimal ranking allows the user to easily understand, for example: whether the system retrieved the correct documents but ranked them badly, and in this case the optimal curve is close to the ideal one while the system curve is lower than both of them, or whether the system missed to retrieved too many documents, and in this case the system and optimal curves are both faraway from the ideal one. In the former case, the user can understand that he needs to perform just a re-ranking of the retrieved documents while in the second case he needs to completely change the search strategy. In other terms, the proposed techniques allow us to understand whether the system under examination is satisfactory from the recall point of view but unsatisfactory from the precision one, thus possibly benefiting from re-ranking, or if the system also has a too low recall, and thus it would benefit more from re-querying.

The proposed ranking model and the related visualization are quite innovative because, usually, information visualization and visual analytics are exploited to improve the presentation of the results of a system to the end user, rather than applying them to the exploration and understanding of the performances and behaviour of an IR system. Secondly, comparisons are usually made with respect to ideal ranking only while our method allows user to compare a system also which respect to the optimal ranking produced with the system results, thus giving the possibility of better interpreting the obtained results.

2.2 Supporting What-If Analysis

When it comes to the second goal, i.e. allowing users to anticipate the impact of a modification, we allow them to simulate what happens when you change the ranking of a given document for a certain topic not only in terms of which other documents will change their rank for the that topic but also in terms of the effect that this change has on the ranking of the other topics. In other terms, we try to give the user an estimate of the "domino effect" that a change in the ranking of a single document can have.

Let us consider the following scenario: in the visualization described above, you have spotted a misplaced document and after inspecting the document you have also understood the motivation of its misplacement, e.g. a bug in a component of the evaluated IR system. At this point, before actually re-implementing a part of the system, you would like to see what could happen if you move that document to a higher position in the ranking, that is a reasonable expected outcome of fixing the detected bug. Due to the complexity of the system, the interaction among the different components, and the inter-relationships among the documents and the topic, you will not end up moving just that single misplaced document but actually a whole cluster of related documents, for that topic, will be moved and some of them can be relevant while some other not. Therefore, even in the case of a single topic, moving a single document causes a kind of "domino effect" since a whole cluster of documents is moved and this could affect performances in a different way than the expected one.

Moreover, when you simulate the move of a single document (and all the related documents), you produce a new ranking for a given topic which corresponds to a new version of your system, in our case a bug fixing ina a component of the system. However, this new version of the system will now behave differently when ranking documents for the other topics in your experimental collection. Therefore, a change in the system which positively affects the performances on topic t_1 may have the side-effect to be detrimental for the performances on topic t_2 and we would like to give users an estimate also of this kind of "domino effect".

Therefore, the overall goal is to have an initial raw estimate of the effect of a planned modification before actually implementing it in terms of effect both for the topic under examination and for the other topics. This gives researchers and developers the possibility of exploring several alternatives before having to implement them and of determining a reasonable trade-off between the effort and costs for given modifications and the expected improvements.

In order to achieve this goal we have defined two analytical models which have been then exploited to drive the interaction with the visualization, as shown, for example, in Figures 3a and 3b.

The first analytical model is based on learning to rank tech-

niques [Liu, 2009] in order to learn a model of the system under examination from the ranked lists produced for each topic $t \in T$.

From the learned model of the system, we then perform clustering in order to understand which documents would be moved together with a selected one, as part of the same cluster according to the system way of working. Learning is needed for a twofold reason: first, during evaluation campaigns you do not have the possibility of dealing with the actual systems but only with their results and so, to analyze the behaviour of a system, you need to learn a model of it from its results; secondly, and more in general, as described above the overall objective of this work is to conduct an analysis before we need to interact with and modify the system, and so a model of the system is needed to conduct such analysis.

Learning to rank is also needed in the subsequent step, i.e. when we estimate the impact of a modification on a topic on the other topics and on the run seen as a whole. In that case, after we have produced a new ranking for the given topic, operation which simulates the implementation of a new version of the system, we need to learn a new model which represents this new version of the system, in order to be able to exploit that model to produce a new ranking for another topic and be able to estimate the effect of the changes on this second topic.

The interactive visual system we present here, allows for a topic-by-topic analysis of the domino effect as well as for a general analysis based on the metrics calculated on the whole set of topics.

The second analytical model is devoted to frame what happens when you try to move a document from one position to another one in the ranking, how the other documents in the same cluster move in accordance with the move of the selected document, and how the other documents in the list relocate themselves.

These two joint-working analytical models and the related interaction in the visualization represent a quite original contribution of the paper since, to the best of our knowledge, in the IR field, neither this kind of what-if analysis nor this way of exploiting learning to rank have been attempted before, probably also because they make more sense when paired with a corresponding visualization and interaction part.

3. RELATED WORK

The graded-relevance metrics considered in this paper are based on cumulative gain [Järvelin and Kekäläinen, 2002], which is related to the idea behind a graded-relevance metric called the sliding ratio proposed back in the 1960s [Korfhage, 1997].

The 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.

Typical instantiations of DCG measures make use of positive gains and logarithmic functions to smooth the discount for higher ranks – e.g. a $\log_2$ function is used to model impatient users while a $\log_{10}$ function is used to model very patient users in scanning the result list. More recent works [Keskustalo et al., 2008] have tried to assign negative gains to not relevant documents: this gives rise to performance curves that start falling sooner than the standard ones when non relevant documents are retrieved and let us better grasp, from the user's point of view, the progression of retrieval towards success or failure.

A work that exploits DCG to support analysis is [Teevan et al., 2010] 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 [Järvelin and Kekäläinen, 2002], but using the gap between curves to support analysis as in [Teevan et al., 2010]. Moreover, the models proposed in this paper provide the basis for the development of Visual Analytics (VA) environment that can provide us with:

- a quick and intuitive idea of what happened in a ranking list;
- an understanding of what are the main reasons of its perceived performances;
- the possibility of exploring the consequences of modifying the system characteristics through an interactive what-if scenario.

In the VA community previous approaches have been proposed for visualizing and assessing a ranked list of items, e.g. using rankings for presenting the user with the most relevant visualizations [Seo and Shneiderman, 2005], or for browsing the ranked results [Derthick et al., 2003a]; other proposals, see, e.g., [Seo and Shneiderman, 2004], use rankings for presenting the user with the most relevant visualizations, or for browsing the ranked result, see, e.g., [Derthick et al., 2003b], but do not deal with the problem of observing the ranked item position, comparing it with an ideal solution to assess and improve the ranking quality.

Visualization strategies have been adopted for analyzing experimental runs, e.g. beadplots in [Banks et al., 1999]. Each row in a beadplot corresponds to a system and each "bead", which can be gray or coloured, corresponds to a document. The position of the bead across the row indicates the rank position in the result list returned by the system. The same color indicates the same document and therefore the plot makes it easy to identify a group of documents that tend to be ranked near to each other. The colouring scheme uses spectral (ROYGBIV) coding; the ordering adopted for colouring (from dark red for most relevant to light violet for least relevant) is based on a reference system, not on graded judgements and the optimal ranking as in our work. Moreover, in [Banks et al., 1999] the strategies are adopted for a comparison between the performance of different systems, i.e. the diverse runs; our approach aims at supporting the analysis of a single system, even though it can be generalized for systems comparison.

Another related work is the Query Performance Analyzer (QPA) described in [Sormunen et al., 2002]. This tool provides the user with an intuitive idea of the distribution of relevant documents in the top ranked positions through a *relevance bar*, where rank positions of the relevant documents are highlighted; our VA approach extends the QPA relevance bar by providing an intuitive visualization for quantifying the gain/loss with respect to both an optimal ranking. QPA also allows for the comparison between the Recall-Precision graphs of a query and the most effective query formulations issued by users for the same topic; in contrast, the curves considered in this work allow the comparison between the system performance with the optimal and ideal ranking that can be obtained from a result list.

However, none of these works deal with the problem of observing the ranked item position, comparing it with an ideal solution, to

assess and improve the ranking quality. In [blinded] the authors explored the basic issues associated with the problem, providing basic metrics and introducing a VA web-based system for exploring the quality of a ranking with respect to an optimal solution. This paper extends such results, allowing for assessing the ranking quality with both the optimal and the ideal solutions and presenting an experiment based on data from runs of the TREC7 Ad-hoc track and the pool obtained in [Järvelin and Kekäläinen, 2002].

4. MODELS FOR INTERACTION

4.1 Clustering via Learning to Rank

Ranking models highly depends by the tuning of several parameters which in most of the cases is done manually. This is a difficult task especially when the ranking model have many parameters and it is the result of the combination of several other models. To this purpose "Learning to Rank" methods can help because they are effective tools to automatically tune parameters and combine multiple evidences [Liu, 2009]. Learning to rank methods are feature-based and a widely-used list of features usually adopted by learning to rank techniques is described in [Geng et al., 2007]. The discriminative training is an automatic learning process based on the training data with four pillars: the input space (e.g. the object under investigation, usually represented as feature vectors), the output space (e.g. the learning target w.r.t. the input space), the hypothesis space (e.g. the class of functions mapping the input space into the output space), and a loss function (e.g. a function that measures to what degree the prediction is in accordance with the ground truth).

A training set consists of n training queries $q_j (j = 1 \ldots n)$, their associated documents represented as feature vectors $\mathbf{x^{(j)}} = \{x_i^{(j)}\}_{i=1}^{m^{(j)}}$ (where $x_i^{(j)}$ is the i^{th} document retrieved for q_j and $m^{(j)}$ is the number of documents retrieved for q_j) and the corresponding relevance values (i.e. $y^{(j)}$). Then a learning algorithm is employed to learn the ranking model corresponding to the way of combining features.

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" [van Rijsbergen, 1979] 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 [Berkhin, 2006]. 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:

1. employing a test IR system and submitting $FV_j[i]$ as a query, thus retrieving a result vector FV_i of documents;
2. determining $C_i = FV_j \cap FV_i$;
3. 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. It is worthwhile to point out that FV_i usually contains a different set of documents respect to FV_j; we are interested only in the documents belonging to the original rank list (i.e. the documents in FV_j) because we want to specifically evaluate the effect of the tuning of the ranking model and not other aspects related to an IR system as a whole, such as its ability of retrieving relevant documents.

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 $FV_j[i]$.

4.2 Rank Gain/Loss Model

According to [Järvelin and Kekäläinen, 2002] 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 a model that is parametric with respect to this choice. However, to fix the ideas, we recall the original DF proposed in [Järvelin and Kekäläinen, 2002]:

$$DF(V[i]) = \begin{cases} GT(V[i]), \text{ if } i \leq x \\ GT(V[i])/\log_x(i), \text{ if } i > x \end{cases} \quad (1)$$

that reduces, in a logarithmic way, the relevance of a document whose index is greater than the logarithm base. For example, if $x = 2$ a document at position 16 is valuable as one fourth of the original value. The quality of a result can be assessed using the function $DCG(V,i) = \sum_{j=1}^{i} DF(V[j])$ that estimates the information gained by a user who examines the first i documents of V. This paper exploits the variant adopted in `trec_eval` where GT is divided by $\log_x(i+1)$.

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. We denote with $OptPerm(V)$ the set of optimal permutations of V such that $\forall OV \in OptPerm(V)$ it holds that $GT(OV[i]) \geq GT(OV[j]), \forall \{i,j\} \leq n \bigwedge i < j$, that is, OV maximizes the values of DCG(OV,i)$\forall i$. In other words, $OptPerm(V)$ represents the set of the optimal rankings for a given search result.

It is worth noting that each vector in $OptPerm(V)$ is composed of $k+1$ intervals of documents sharing the same GT values. As an example, assuming a result vector composed by 12 elements and $k = 3$, a possible sequence of GT values of an optimal vector OV is $< 3,3,3,3,2,2,2,2,1,1,0,0 >$; according to this we de-

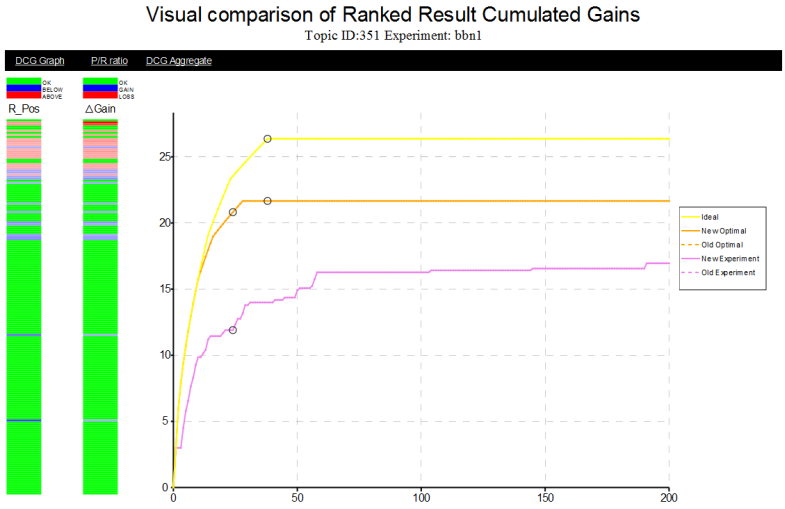


Figure 1: A Screen-shot of the Visual Interactive System.

fine the $max\_index(V,r)$ and $min\_index(V,r)$ functions, with $0 \leq r \leq k$, which return the greatest and the lowest indexes of elements in a vector belonging to $OptPerm(V)$ that share the same GT value r. For example, considering the above 12 GT values, $min\_index(V,2) = 5$ and $max\_index(V,2) = 8$.

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

$$R\_Pos(V[i]) = \begin{cases} 0, \text{ if } min\_index(V,GT(V[i])) \leq i \leq max\_index(V,GT(V[i])) \\ min\_index(V,GT(V[i])) - i, \text{ if } i < min\_index(V,GT(V[i])) \\ max\_index(V,GT(V[i])) - i, \text{ if } i > max\_index(V,GT(V[i])) \end{cases} \quad (2)$$

$R\_Pos(V[i])$ 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 compares $\forall i$ the actual values of $DF(V[i])$ with the corresponding values in OV, $DF(OV[i])$: $\Delta\_Gain(V,i) = DF(V[i]) - DF(OV[i])$. Note that while $\text{DCG}(V[i]) \leq \text{DCG}(OV[i])$ the $\Delta\_Gain(V,i)$ function assumes both positive and negative values. In particular, 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. Visually inspecting the values of these two metrics allows the user to easily locate misplaced elements and understand the impact that such errors have on DCG.

4.3 What-if Analysis Model

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: $IC_i = \{j | FV[j] \in C_i\}$. It is worth noting that documents in C_i might belong to a part of FV that is not shown in the actual results ($index > 200$). 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.

Formally, after applying a $Move(i,j)$ operator, we obtain a permutation FV' of the vector FV. The steps to compute FV' are the following.

1. $\triangle = min(min(IC_i) - 1, j - i)$; Initialize FV' to 0; $holes = \{k | k \in [min(IC_i), max(IC_i)] \wedge k \notin IC_i\}$
2. $FV'[i] = FV[i]$, $if\ i > max(IC_i) \vee i < min(IC_i) - \triangle$;
3. $FV'[i - \triangle] = FV[i]$, $if\ i \in IC_i$;
4. Iterate

- $j = min\{k|FV'[k] = 0\}$;
- $FV'[j] = FV[min(holes)]$;
- $holes = holes - min(holes)$;

until $holes = \emptyset$.

5. THE VISUAL INTERACTIVE SYSTEM

Step 1 computes the allowed shift, fill FV' of 0s, and computes the set of indexes that corresponds to documents in the range $[min(IC_i), max(IC_i)]$ not belonging to the cluster C_i. Step 2 copies the part of FV that is not affected by the shift and step 3 moves up of $\triangle$ the elements in C_i. Step 4 moves down the documents ousted in step 3.

The interactive visual system consists of a Web application that retrieves data from a remote server and allows the user to visually analyze it in an interactive way. it deals with one topic t at time: it 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. While visually inspecting the ranked list, 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. This cluster of documents simulates the "domino effect" within the given topic t, discussed in Section 2.

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 Learning to Rank 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, as discussed in Section 2.

Figure 1 shows a screen shot of the running interactive visual system. Three different type of visualizations can be selected from the analyst:

DCG Graph: this visualizes the ranking list of a single topic. The analyst can evaluate the ranking list of retrieved documents and change the position of a "misplaced" document and, consequently, its associated document cluster, in order to obtain a better ranking.

Precision/Recall Graph: this allows the analyst to evaluate the overall precision-recall ratio on the whole set of topics. It also allows the analyst to evaluate the impact of an improvement of the DCG for a single topic on the run seen as a whole.

DCG Aggregate Graph: this aims at evaluating the overall quality of the ranking for all the topics of the experiment, focusing on the variability of the results.

5.1 DCG Graph

Figure 1 shows the DCG Graph. 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 Discounted Cumulated Gain DCG metric. This topic has been discussed in [blinded] and it will not be further investigated in this paper.

In this paper we focus on a novel what-if functionality allowing 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.

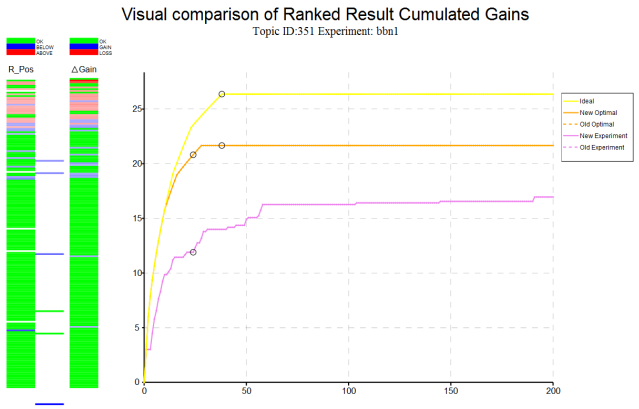
In Figure 2a it is possible to see the animated phase of interactive re-ranking of the documents: after highlighting and moving the target document t from the starting position to a new one, the user will be presented with an animated re-ranking of the documents connected to the target one. Once the 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.

Figure 2b shows 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 loss obtained from this particular re-rank.

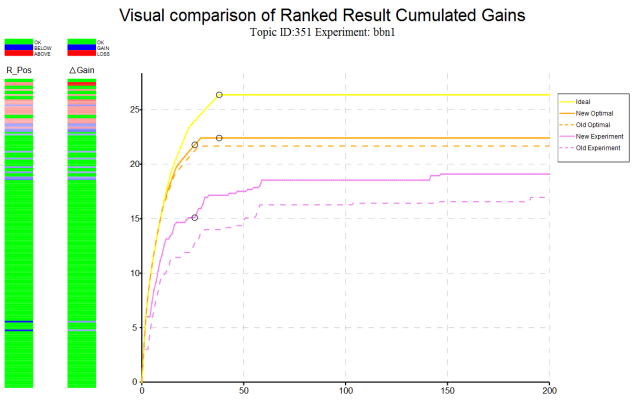
5.2 Precision/Recall Graph

Figure 3a shows the Precision/Recall graph of the whole experiment. The dotted line represents the actual trend and the continuous one the previous.

On the left side, there are the values of two aggregate measures: MAP (Mean Average Precision) and GMAP (Geometric Mean Average Precision). This graph is useful for both static and interactive

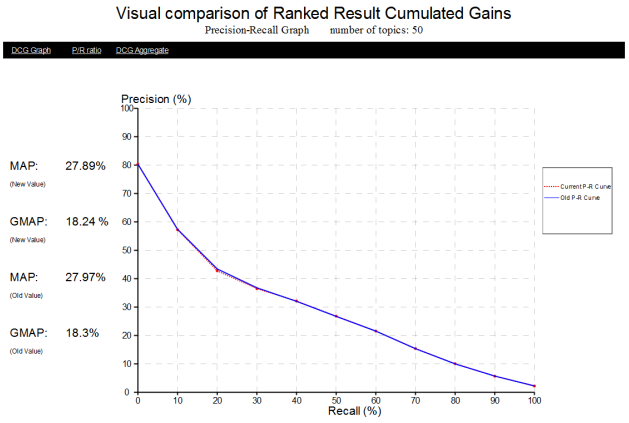


(a) Movement of a target document and its cluster.

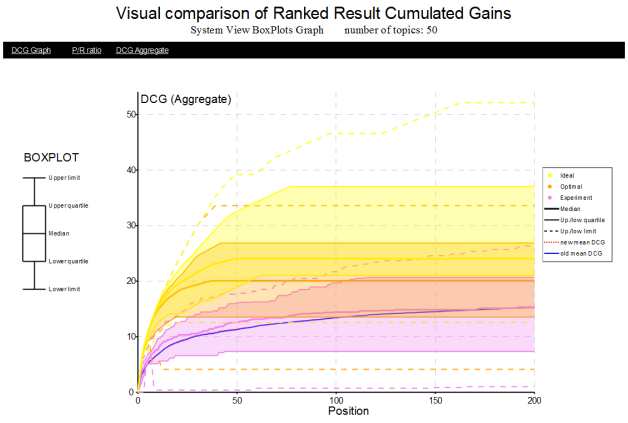


(b) New graphs after the movement.

Figure 2: An interaction with the system.



(a) Precision Recall trend after the interaction.



(b) Aggregated Ideal, Optimal and Experiment curves on the whole set of topics

Figure 3: Precision-Recall and DCG aggregate graphs.

analysis; in fact, the former provides the analyst with an overall view of the precision-recall trend, whereas the latter allows the analyst to understand which effects the re-ranking of a specified ranking list has produced to the overall precision-recall graph. This is visible in Figure 3a, where a change of position of a document along with its cluster has slightly worsened the precision-recall ratio in the central area of the graph. Accordingly to that, also the MAP and GMAP show a little worsening with respect to the initial values.

5.3 DCG Aggregate Graph

Figure 3b shows the DCG Aggregate Graph. It is an aggregate representation based on the boxplot statistical tool showing the variability of the DCG 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).

On top of that, 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) and the median values inside it.

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. Following this rationale the median lines are thick in order to be different to the upper/lower quartile ones represented with normal thickness and to upper/lower limit ones represented with dashed lines.

In figure 3b we can appreciate that, in this particular case, the optimal and experiment areas overlap for a good extent, and the median curve of the experiments tends to the one of the optimal; at the same time, also the median curve of the optimal is not far with respect to the ideal median curve. 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 not broad, demonstrating a good homogeneity in values, the ideals box area is big meaning a high variability of the data among the different topics.

6. EXPERIMENTATION

6.1 Experimental Collections

The test collection adopted is based on data from the TREC7 Ad-hoc test collection. A subset of all the *topics* 351-400 is considered, specifically those re-assessed in [Järvelin and Kekäläinen, 2002]; details on the re-assessment procedure can be found in [Sormunen et al., 2002]. Indeed, the *relevance judgments* adopted are those obtained by the evaluation activity carried out in that paper. All the relevant documents of 20 TREC7 topics and 18 TREC8 topics were re-assessed together with 5% of documents judged as not relevant, where assessment was performed using a four graded relevance scale; details on the re-assessment procedure can be found in [Järvelin and Kekäläinen, 2002]. The TREC7 Ad-hoc test collection together with this set of judgments were used because we adopt the DCG family of measures in the VA approach. The way the VA approach can be adopted to support researchers and analysts during the evaluation is based on runs submitted to the TREC7 Ad-hoc Track. In order to be consistent with the choice adopted in [Järvelin and Kekäläinen, 2002] we will visualize the curves for top $k = 200$ rank positions.

6.2 Construction of the Clusters

For each experiment we take into account all the submitted ranked lists, one per topic (i.e., query q_j) in the considered subset of topics. We define a feature extractor function $\phi(q_j, d_i)$ which represents each query-document pair $((q_j, d_i))$ as a 19-dimensional feature vector defined as: $\mathbf{v_{ji}} = \langle q_j,\ x^{(j)}_{1(i)},\ x^{(j)}_{2(i)},\ \ldots,\ x^{(j)}_{19(i)},\ \mathbf{y^{(j)}} \rangle$. We chose to adopt a subset of the features used in the LEarning TO Rank (LETOR) Datasets [Liu et al., 2007] which is a widely-used benchmark dataset defined for learning to rank algorithms. The 19 features we selected are those ones we consider more significant for the TREC7 Ad-hoc test collection data; e.g. document length, covered query term number, covered query term ratio, IDF (Inverse document frequency), sum of term frequency, mean of term frequency, BM25, etc.

For each available experiment we defined a training set composed by one thousands feature vectors for each one of the considered queries. We used this training set to learn the ranking model of each IR system participating to TREC7, by using the RankBoost algorithm [Freund et al., 2003]. We run RankBoost with the following settings: 30 iterations, 5-fold cross validation, 25 threshold candidates to search, and nDCG@10 optimization on the training data.

Once we learned the ranking model, we retrieved a set of relevant documents for each document in each ranked list for each available experiment, by employing the open-source Terrier *v*3.5[1] IR system. We rank the documents in the retrieved sets by using the learned ranking model so defining a cluster C_i for each considered document d_i.

6.3 Experimental Analysis

At time of writing a user study of the system has not been performed; however, the prototype has been tested with IR experts that have reported positive feedbacks together with several suggestions for improvement. The objective of the visual interactive system we developed is to support a researcher or analyst investigating how to improve the effectiveness of an IR approach, when the results for one or more queries on the same topic are available. As we have seen the system allows us to visualize the ranked list retrieved for a topic (e.g. Figure 2a) and interact with it by choosing a misplaced document and moving it in a new position. Afterwards, the system provides us with a set of tools to analyze the effect of this movement. Indeed, we can see: (i) how the DCG curve for the selected topic changes as a consequence of the movement of the target document and its cluster; (ii) how the precision-recall graph for the selected topic changes; (iii) how this movement affects the DCG and the precision-recall curves for the other topics of the run (i.e. "domino effect"); and (iv) how the precision-recall graph of the whole run is affected.

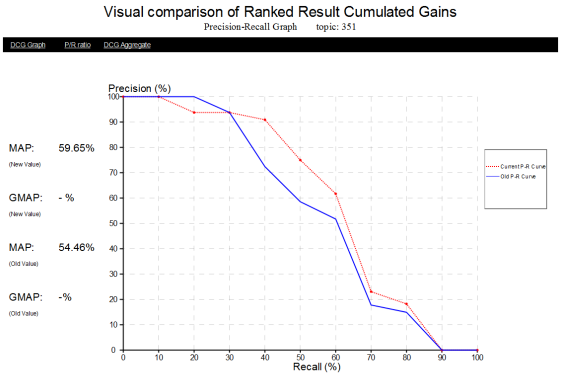


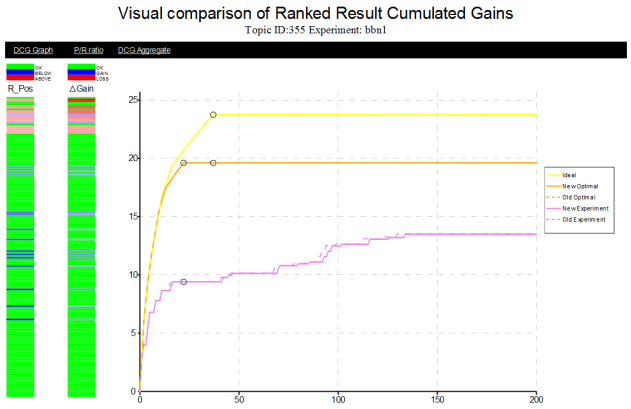
Figure 4: Precision-recall graph for topic 351 before and after moving upwards FBIS3-10551 and its cluster.

For this analysis we consider the run named "bbn1" submitted to the TREC7 Ad-Hoc Track [Voorhees and Harman, 1999]. In particular, we consider topic 351, the DCG graph of which can be seen in Figure 2a. Starting from this graph, using the *R_Pos* vector representation as a guide, we selected the document identified as "FBIS3-10551" and ranked at position 57. This document is judged as "highly relevant" (with weight 3) for topic 351, but it is ranked very low in the ranking list of "bbn1". We chose to move FBIS3-10551 upwards of 56 positions and thus at position 1. The cluster of this document is composed by 73 documents, 30 of which are ranked between position 0 and 56; 20 of these 30 are relevant documents which do not increase their position (as described by the *Move* algorithm presented in Section 4.3). The current implementation of the system moves document FBIS3-10551 upwards of 56 positions along with the ten most similar documents in its cluster. The first 10 documents in the cluster are all relevant – i.e. 7 are graded as "highly relevant" and 3 are graded as "fairly relevant" – for topic 351. This means that the cluster is built in a good way and that if we move upwards the first 10 documents, we move only relevant documents thus improving the overall ranking.

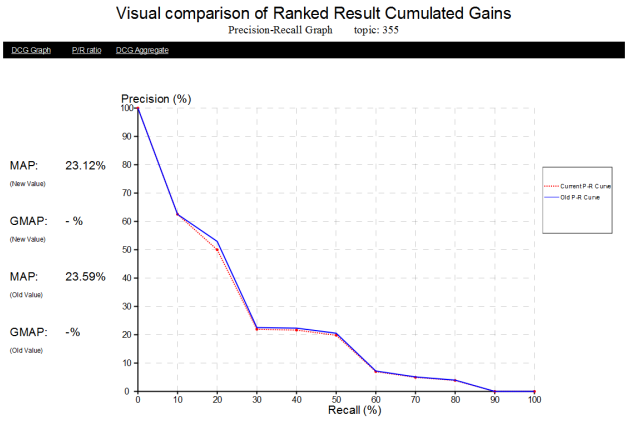
Figure 2b shows the effect of this movement in the DCG curve of topic 351. We can see that the DCG increases substantially because we move upwards only relevant documents. In Figure 4 we can see the precision-recall graph for topic 351 before and after the shifting of FBIS3-10551 and its cluster. Also in this case we can see a significant improvement of the measures (MAP is improved of 5%).

At this point, the system allows us to evaluate the domino effect

[1] http://terrier.org/



(a) DCG curve of topic 355.



(b) Precision-recall curve of topic 355.

Figure 5: Domino effect: The change in topic 351 slightly worsens the performances of topic 355 (MAP −0.47%).

this movement produced on the other topics. Indeed, the movement of a document along with its cluster corresponds to a change in the ranking algorithm of the IR system under investigation; we simulate this change thanks to the learning to rank methods we described and the visual system allows the final user to evaluate the effect of this change.

We have seen that in this experimentation we moved upwards a total of 6 documents, thus we expect a weak domino effect on the other topics and on the run as a whole.

In Figure 5a we can see the DCG curve and in Figure 5b we can see the precision-recall curve for topic 355 before and after the movement of document `FBIS3-10551` in the ranking list of topic 351. As expected the change is small but still visible; for topic 355 we have a worse DCG curve as well as a worse precision-recall curve. The graph shows that MAP decreases of 0.47%. On the other hand, in Figure 6a and 6b we can see the DCG and the precision-recall curves for topic 400. In this case, there is an improvement of the performances; we can say that an improvement on topic 351 has a positive effect also on topic 400.

In terms of MAP the improvement of the ranking for topic 351 produced a domino effect which improved 7 topics, worsened 18 topics, and did not affect the remaining topics. The visual system provides us with an overall view of the performances of the run after the change occurred; we can see the overall precision-recall graph of the run in Figure 3a where there the overall MAP decreases of 0.8%. Furthermore, in Figure 3b we can see the overall mean DCG curve before and after the change in the ranking algorithm has been simulated.

7. 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 behaviour; (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. Thus, the overall goal of the paper has been to provide users with tools and methods to investigate the performances of a system and explore different alternatives for improving it avoiding a continuous iteration of trials-and-errors to see if the proposed modifications actually provide the expected improvements.

This ambitious goal is especially important for the design and development of complex IR systems when you consider the amount of time and human effort that needs to be devoted to failure analysis and system re-implementation and the economic impact that experimental evaluation activities have both in terms of benefits for researchers and industries and of costs for organizing them.

The goal has been achieved by developing three coordinated and jointly-working analytical models – the first one for quantifying ranking gains/losses, the second one for clustering documents and learning to rank as the system under examination, and the third one for conducting the what-if analysis and determining the effect of moving a document in the ranked list. These three analytical model back the visualization part, where they are integrated and allow for a smooth interaction of the user with the experimental results.

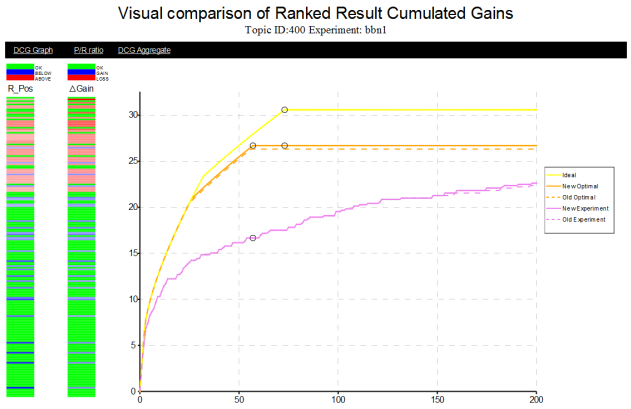
Finally, the proposed prototype has been tested and experimented with real data taken from the Text REtrieval Conference (TREC) 7 campaign. This allows us to assess not only its effectiveness when interacting we actual performances of an IR system but it also provides us with a great deal of comparability with previous work, since these datasets have been widely used in the IR field for developing and comparing new metrics.

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 Learning to Rank Model are used to compute the new ranking and the associated DCG functions. In particular, we want to define an adaptive threshold and we want to define a non uniform function associating each element in the cluster above the threshold with a shift that is function of its relevance value.

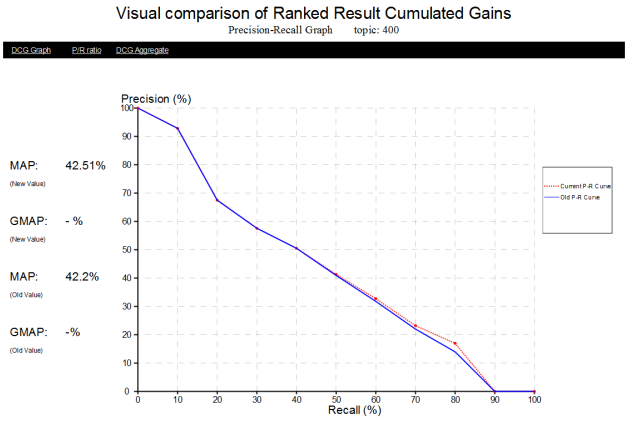
8. REFERENCES

[Banks et al., 1999] Banks, D., Over, P., and Zhang, N.-F. (1999). Blind Men and Elephants: Six Approaches to TREC data. *Information Retrieval*, 1:7–34.

[Berkhin, 2006] Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. In Kogan, J., Nicholas, C., and Teboulle,



(a) DCG curve of topic 400.



(b) Precision-recall curve of topic 400.

Figure 6: Domino effect: The change in topic 351 slightly improves the performances of topic 400 (MAP +0.31%).

M., editors, *Grouping Multidimensional Data*, pages 25–71. Springer-Verlag, Heidelberg, Germany.

[Derthick et al., 2003a] Derthick, M., Christel, M. G., Hauptmann, A. G., and Wactlar, H. D. (2003a). Constant density displays using diversity sampling. In *Proceedings of InfoVis'03*, pages 137–144, Washington, DC, USA. IEEE Computer Society.

[Derthick et al., 2003b] Derthick, M., Christel, M. G., Hauptmann, A. G., and Wactlar, H. D. (2003b). Constant density displays using diversity sampling. In *Proceedings of the IEEE Information Visualization*, pages 137–144.

[Freund et al., 2003] Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research*, 4(Nov):933–969.

[Geng et al., 2007] Geng, X., Liu, T.-Y., Qin, T., and Li, H. (2007). Feature Selection for Ranking. In Kraaij, W., de Vries, A. P., Clarke, C. L. A., Fuhr, N., and Kando, N., editors, *Proc. 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 407–414. ACM Press, New York, USA.

[Harman and Buckley, 2009] Harman, D. and Buckley, C. (2009). Overview of the reliable information access workshop. *Information Retrieval*, 12(6):615–641.

[Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information System (TOIS)*, 20(4):422–446.

[Keskustalo et al., 2008] Keskustalo, H., Järvelin, K., Pirkola, A., and Kekäläinen, J. (2008). Intuition-Supporting Visualization of User's Performance Based on Explicit Negative Higher-Order Relevance. In Chua, T.-S., Leong, M.-K., Oard, D. W., and Sebastiani, F., editors, *Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 675–682. ACM Press, New York, USA.

[Korfhage, 1997] Korfhage, R. R. (1997). *Information Storage and Retrieval*. Wiley Computer Publishing, John Wiley & Sons, Inc., USA.

[Liu, 2009] Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

[Liu et al., 2007] Liu, T.-Y. Y., Xu, J., Qin, T., Xiong, W., and Li, H. (2007). LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. In Joachims, T., Li, H., Liu, T.-Y., and Zhai, C., editors, *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*.

[Seo and Shneiderman, 2004] Seo, J. and Shneiderman, B. (2004). A rank-by-feature framework for interactive exploration of multidimensional data. In *Proceedings of the IEEE Information Visualization*, pages 65–72.

[Seo and Shneiderman, 2005] Seo, J. and Shneiderman, B. (2005). A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4:96–113.

[Sormunen et al., 2002] Sormunen, E., Hokkanen, S., Kangaslampi, P., Pyy, P., and Sepponen, B. (2002). Query Performance Analyser – a Web-based tool for IR research and instruction. In Järvelin, K., Beaulieu, M., Baeza-Yates, R., and Hyon Myaeng, S., editors, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, page 450. ACM Press, New York, USA.

[Teevan et al., 2010] Teevan, J., Dumais, S. T., and Horvitz, E. (2010). Potential for personalization. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(1):1–31.

[van Rijsbergen, 1979] van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, England, 2nd edition.

[Voorhees and Harman, 1999] Voorhees, E. and Harman, D. (1999). Overview of the Seventh Text REtrieval Conference (TREC-7). In *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)*. Springer-Verlag, Heidelberg, Germany.