What-If Analysis: A Visual Analytics Approach to Information Retrieval Evaluation

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. This paper focuses on the innovative visual analytics approach realized by the Visual Analytics Tool for Experimental Evaluation (VATE^2) system, which eases and makes more effective the experimental evaluation process by introducing the what-if analysis. The what-if analysis is aimed at estimating the possible effects of a modification to an Information Retrieval (IR) system, in order to select the most promising fixes before implementing them, thus saving a considerable amount of effort.
VATE^2 builds on an analytical framework which models the behavior of the systems in order to make estimations, and integrates this analytical framework into a visual part which, via proper interaction and animations, receives input and provides feedback to the user. We conducted an experimental evaluation to assess the numerical performances of the analytical model and a validation of the visual analytics prototype with domain experts. Both the numerical evaluation and the user validation have shown that VATE^2 is effective, innovative, and useful.

1 Introduction

IR systems operate using a best match approach: in response to an often vague user query, they return a ranked list of documents ordered by the estimation of their relevance to that query. In this context effectiveness, meant as the ability of systems to retrieve and better rank relevant documents while at the same time suppressing the retrieval of not relevant ones, is the primary concern. Since there are no a-priori exact answers to a user query, experimental evaluation [10] based on effectiveness is the main driver of research and innovation in the field. Indeed, the measurement of system performances from the effectiveness point of view is basically the only means to determine the best approaches and to understand how to improve IR systems.

Nowadays, user tasks and needs are becoming increasingly demanding, the data sources to be searched are rapidly evolving and greatly heterogeneous, the interaction between users and IR systems is much more articulated, and the systems themselves become increasingly complicated and constituted by many interrelated components. As an example consider what web search is today: highly diversified results are returned from web pages, news, social media, image and video search, products and more, and they are all merged together by adaptive strategies driven by current and previous interaction of the users with the system.

Understanding and interpreting the results produced by experimental evaluation is not a trivial task, due to the complex interactions among the components of an IR system. Nevertheless, succeeding in this task is fundamental for detecting where systems fail and hypothesizing possible fixes and improvements. As a consequences, this task is mostly manual and requires huge amounts of time and effort.

Moreover, after such activity, the researcher needs to come back to design and then implement the modifications that the previous analysis suggested as possible solutions to the identified problems. Afterwards, a new experimentation cycle needs to be started to verify whether the introduced modifications actually give the expected improvement. Therefore, the overall process of improving an IR system is extremely time and resource demanding and proceeds through cycles where each new feature needs to be implemented and experimented.

The goal of this paper is to introduce a new phase in this cycle: we call it *what-if analysis* and it falls between the experimental evaluation and the design and implementation of the identified modifications. What-if analysis aims at estimating what the effects of a modification to the IR system under examination could be before actually being implemented. In this way researchers and developers can get a feeling of whether a modification is worth being implemented and, if so, they can go ahead with its implementation followed by a new evaluation and analysis cycle for understanding whether it has produced the expected outcomes.

What-if analysis exploits Visual Analytics (VA) techniques to make researchers and developers: (i) interact with and explore the ranked result list produced by an IR system and the achieved performances; (ii) hypothesize possible causes of failure and their fixes; (iii) estimate the possible impact of such fixes through a powerful analytical model of the system behavior.

What-if analysis is a major step forward since it can save huge amounts of time and effort in IR system development and, to the best of our knowledge, it has never been attempted before.

The paper is organized as follows: Section 2 discusses some related works; Section 4 explains in detail the proposed analytical framework which is then experimentally evaluated in Section 5; Section 6 presents the visual analytics environment called Visual Analytics Tool for Experimental Evaluation (VATE^2); and, Section 7 draws some conclusions and presents an outlook for future work.

2 Related Works

Experimental evaluation is based on the Cranfield methodology [6] which makes use of experimental collections $\mathscr{C} = (D, T, GT)$ consisting of: a set of documents D representing the domain of interest; a set of topics T, which simulates and abstracts actual user information needs; and, the ground-truth GT, i.e. a kind of "correct" answer, where for each topic $t \in T$ the documents $d \in D$ relevant to it are determined. System outputs are then scored with respect to the ground-truth using whole breadth of performance measures [15]. The ground-truth can consist of both binary relevance judgments or multi-graded ones [12].

Experimental evaluation is a demanding activity in terms of effort and required resources that benefits from using shared datasets, which allow for repeatability of the experiments and comparison among state-of-the-art approaches. Therefore, over the last 20 years, experimental evaluation has been carried out in large-scale evaluation campaigns at international level, such as the Text REtrieval Conference (TREC)[3] in the US and the Conference and Labs of the Evaluation Forum (CLEF)[4] in Europe.

The activity described in the previous section and aimed at understanding how and why a system has failed is called *failure analysis*. To give the reader an idea of how demanding it can be, let us consider the case of the the Reliable Information Access (RIA) workshop [9], which was aimed at systematically investigating the behavior of just one component in an IR system, namely the relevance feedback module. Harman and Buckley in [9] reported that, for analyzing 8 systems, 28 people from 12 organizations worked for 6 weeks requiring from 11 to 40 person-hours per topic for 150 overall topics. These figures do not take into account the time then needed to implement the identified modifications and perform another evaluation cycle to understand if they had the desired effect.

VA is typically exploited for the presentation and exploration of the documents managed by an IR system [18]. However, much less attention has been devoted to applying VA techniques to the analysis and exploration of the performances of IR systems [4]. To the best of our knowledge, our previous work is the most systematic attempt. We explored several ways in which VA can help the interpretation and exploration of system performances [7]. This preliminary work led to the development of Visual Information Retrieval Tool for Upfront Evaluation (VIRTUE), a fully-fledged VA prototype which specifically supports performance and failure analysis [3].

We then started to explore the need for what-if analysis in the context of large-scale evaluation campaigns, such as TREC or CLEF. In this context, evaluators do not have access to the tested systems but they can only examine the final outputs, i.e. the ranked result lists returned for each topic. In [1, 2] we continued the study for the evaluation campaigns and we set up an analytical framework for trying to learn the behavior of a system just from its outputs, in order to obtain a rough estimation of the possible effects of a modification to the system.

Therefore, in this paper, we put VATE^2 in a different context from the one of evaluation campaigns. Indeed, the present version of VATE^2 is thought for designers and developers of IR systems who have access to all the internals of the system being tested and they have the know-how to hypothesizing possible fixes and improvements.

3 Intuitive Overview

In order to understand how what-if analysis works we need to recall the basic ideas underlying performance and failure analysis as designed and realized by VIRTUE [3].

To quantify the performances of an IR system, VIRTUE adopts the Discounted Cumulated Gain (DCG) family of measures [11] which have proved to be especially well-suited for analyzing ranked results list. This is because they allow for graded relevance judgments and embed a model of the user behavior while he scrolls down the result list which also gives an account of its overall satisfaction.

[3] `http://trec.nist.gov/`

[4] `http://www.clef-initiative.eu/`

We compare the result list produced by an experiment with respect to an *ideal* ranking created starting from the relevant documents in the ground-truth, which represents the best possible results that an experiment can return. In addition to what is typically done, we compare the result list with respect to an *optimal* one created with the same documents retrieved by the IR system but with an optimal ranking, i.e. a permutation of the results retrieved by the experiment aimed at maximizing its performances by sorting the retrieved documents in decreasing order of relevance. Therefore, the *ideal ranking* compares the experiment at hand with respect to the best results possible, i.e. also considering relevant documents not retrieved by the system, while the *optimal ranking* compares an experiment with respect to what could have been done better with the same retrieved documents.

Looking at a performance curve, as the DCG curve is, it is not always easy to spot what the critical regions in a ranking are. Indeed, DCG is a not-decreasing monotonic function which increases only when a relevant document is found in the ranking. However, when DCG does not increase, this could be due to two different reasons: either you are in an area of the ranking where you are expected to put relevant documents but you are putting a not relevant one and thus you do not gain anything; or, you are in an area of the ranking where you are not expected to put relevant documents and, correctly, you are putting a not relevant one, still gaining nothing.

In order to overcome this and similar issues, we introduce the Relative Position (RP) indicator [8]; RP allows us to quantify and explain what happens at each rank position and its paired with a visual counterpart which eases the exploration of the performances across the ranking. RP allows us to immediately grasp the most critical areas as we can see in Figure 1 showing the VATE^2 system. RP quantifies the effect of misplacing relevant documents with respect to the ideal case, i.e. it accounts for how far a document is from its ideal position. Overall, the greater the absolute value of RP is, the bigger the distance of the document from its ideal interval is.

We envision the following scenario. By exploiting VIRTUE a failure analysis is conducted and the user hypothesizes the problem of the IR system at hand. At the same time, the user hypothesizes that if he fixes such failure, a given relevant document would be ranked higher than in the current system. As shown in Figure 1, what VATE^2 offers to the user is: (i) the possibility of dragging and dropping the target document in the estimated position of the rank; (ii) the estimation of which other documents would be affected by the movement of the target document and how the overall ranking would be modified; (iii) the computation of the system performances according to the new ranking.

4 Analytical Framework

We introduce the basic notions regarding experimental evaluation in IR regarding the functioning of VATE^2; for a complete and formal definition of experimental evaluation in IR refer to [3] and [8].

We consider relevance as an ordered set of naturals, say $REL \in \mathbb{N}$, where $rel \in REL$ indicates the degree of relevance of a document $d \in D$ for a given topic $t \in T$. The *ground truth* associates a relevance degree to each document with respect to a topic. So, let T be a set of topics and D a set of documents, then we can define a ground truth for each topic, say $t_k \in T$, as a map GT_{t_k} of pairs (d, rel) with size $|D|$, where $d \in D$ is a document and $rel \in REL$ is its relevance degree with respect to t_k.

We indicate with L_{t_k} a list of triples (d_i, sim_i, rel_i) of length N representing the ranked list of documents retrieved for a topic $t_k \in T$, where $d_i \in D$ is a document $(d_i \neq d_j, \forall i, j \in [1, N] \mid i \neq j)$, $sim_i \in \mathbb{R}$ is a degree indicating the similarity of d_i to t_k, and $rel_i \in REL$ indicates the relevance degree of d_i for t_k; the triples in L_{t_k} are in decreasing order of similarity degree such that $sim_i \geq sim_j$ if $i > j$.

Now, we can point out some methods to access the elements in a ranked list L_t that we will use in the following. $\mathrm{L}_t(1,1) = d_1$ returns the document in the first triple of L_t, $\mathrm{L}_t(2,1) = d_2$ the second document and so on; whereas, $\mathrm{L}_t(:,1)$ returns the list of documents in L_t. In the same vein, $\mathrm{L}_t(1,2) = sim_1$ returns the similarity degree of the first document in L_t and $\mathrm{L}_t(1,3) = rel_1$ returns its relevance degree; moreover, $\mathrm{L}_t(1,:) = (d_1, sim_1, rel_1)$ returns the first triple in L_t.

Relative Position (RP) is a measure that quantifies the misplacement of a document in a ranked list with respect to the ideal one. In order to introduce RP we need to define the concepts of minimum rank and maximum rank of a given relevance degree building on the definition of ideal ranked list. Given an ideal ranked list I_t with length $N \in \mathbb{N}^+$ and a relevance degree $rel \in REL$, then the minimum rank $\min_{\mathrm{I}_t}(rel)$ returns the first position $i \in [1, N]$ at which we find a document with relevance degree equal to rel, while the maximum rank $\max_{\mathrm{I}_t}(rel)$ returns the last position i at which we find a document with relevance degree equal to rel in the ideal ranking.

$$RP_{\mathrm{L}_t}[i] = \begin{cases} 0 & \text{if } \min_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3))\big) \leq i \leq \max_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3)\big) \\ i - \min_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3)\big) & \text{if } i < \min_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3)\big) \\ i - \max_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3)\big) & \text{if } i > \max_{\mathrm{I}_t}\big(\mathrm{L}_t(i,3)\big) \end{cases}$$

The RP measure points out the instantaneous and local effect of misplaced documents and how much they are misplaced with respect to the ideal ranking I_t. In the following definition, zero values denote documents which are within the ideal interval; positive values denote documents which are ranked below their ideal interval, i.e. documents of higher relevance degree that are in a position of the ranking where less relevant ones are expected; and negative values denote documents which are above their ideal interval, i.e. less relevant documents that are in a position of the ranking where documents of higher relevance degree are expected.

In VATE^2, document clustering is adopted in the context of the *failure hypothesis*: "closely associated documents tend to be affected by the same failures", stating the common intuition that a given failure will affect documents with common features, and, consequently, that a fix for that failure will have an effect on the documents sharing those common features.; once the user has selected a target document, say d_j, within a ranked list L_t, our goal is to get a cluster of documents, C_{d_j}, similar to d_j, where the similarity is quantified by the IR system, say S_X, which generated L_t.

The creation of a cluster of documents similar to d_j is very close to the operation done by the S_X IR system to get a ranked list of documents starting from a topic t. Indeed, S_X, takes the topic t and calculates the similarity between the topic and each document in D; afterwards it returns a ranked list L_t of documents ordered by decreasing similarity to the topic. The document cluster creation methodology we adopt in VATE^2 follows this very procedure: given a target document d_j we use it as topic and we submit it to the IR system S_X which returns a ranked list of documents, say C_{d_j}, ordered by decreasing similarity to d_j.

The first document in C_{d_j} is always d_j and then we encounter progressively less similar documents. C_{d_j} tell us which documents are seen in "the same way" by the IR system being tested and that will probably be affected by the same issues found for d_j.

We limit to 10 the number of documents in the cluster to be moved in order to consider only the most similar documents to the target document d_j.

A cluster C_{d_j} is defined as a list of pairs (d_k, sim_k) where $d_k \in D$ is a document and sim_k is the similarity of d_k to d_j. The first document in C_{d_j} is d_j and, by definition, it has the maximum similarity value in the cluster. For every considered cluster we normalize the similarities in the $[0,1]$ range by dividing their values by maximum similarity value in the cluster.

The clusters of documents so defined 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, the ten documents in the C_{d_4} cluster are also moved accordingly.

We developed two variations of movement: a simple one called *constant movement* (*cm*) and a slightly more complex one called *similarity-based movement* (*sbm*). We present the constant movement first and then we build on it to explain the similarity-based one.

Let us consider a general environment where a ranked list L_t is composed of N triples such that the list of documents is $d_1, d_2, \dots, d_N$, where the subscript of the documents indicates their position in the ranking. As a consequence of the *failure hypothesis*, if we move a document $d_j \in \mathrm{L}_t$ from position j to position k – which means that we move d_j of $\Delta = j - k$ positions – we also move the documents in the cluster C_j of Δ positions accordingly.

The constant movement is based on three assumptions:

1. Linear movement: if d_j is moved upward from position j to position k where $\Delta = j - k$, all the documents in its cluster C_j are moved by Δ in the same direction.
2. Cluster independence: the movement of the cluster C_j does not imply the movement of other clusters. This means that when we move d_j in the position of d_k, d_k is influenced by the movement, but the cluster C_k is not.
3. Unary shifting: if C_j is moved by Δ positions, then the other documents in the ranking have to make room for them and thus they are moved downward by one position.

The pseudo-code of the constant movement is reported by Algorithm 1 (implementing the actual movement) and Algorithm 2 (implementing the operations necessary to reorder the ranked list after a movement)[5]. It starts to move by Δ the last document

[5] For the sake of simplicity in these algorithms we employ four convenience methods: POSITION$(\mathtt{L}, \mathtt{d})$ which returns the rank index of d in L, SIZE$(\mathtt{L})$ which returns the number of element of L, ADD$(\mathtt{L}, \mathtt{L}')$ which adds the elements of L' at the end of L and GETCLUSTER$(\mathtt{L_t}, \mathtt{d_j})$ which returns the cluster of d_j.

Algorithm 1: MOVEMENT

```
Input: The ranked list $L_t$, the document $d_j$ to be moved, the target rank position $tPos$.
Output: The ranked list $L_t$ after the movements.
1  $C_{d_j} \leftarrow$ GETCLUSTER($L_t, d_j$)
2  $sPos \leftarrow$ POSITION($L_t, C_{d_j}(1,1)$)
3  for $i \leftarrow$ SIZE($C_{d_j}$) to 1 do
4      $oldPos \leftarrow$ POSITION($L_t, C_{d_j}(i,1)$)
5      if $oldPos == 0$ then
6          $oldPos \leftarrow$ SIZE($L_t$) $+ 1$
7      $newPos \leftarrow oldPos - (sPos - tPos)$
8      $L_t \leftarrow$ REORDERLIST($L_t, oldPos, newPos, C_{d_j}(i,1)$)
9  end
10 return $L_t$
```

in the cluster C_{d_6} which is d_4, so we can see that d_4 is put in the place of d_1 and d_1 is shifted downward by one position; afterwards, the algorithm repeats the same operation for all the other documents in the cluster generating the reordered list $L_t^{'}$.

We have seen that in a general setting, if we move d_j upwards by Δ positions, all the documents in C_j move accordingly by Δ position. There are cases where this is not possible, because the movement is capped on the top by one or more documents in the cluster. As an example, consider a movement upward of d_j, if there is a document $d_w \in C_j$ such that $w < \Delta$, then d_w cannot be moved upwards by Δ position, but at most by w. In this case, d_w is moved by w positions while the other documents, if possible, are moved by Δ positions.

We can see that this movement can be easily changed by altering the three starting assumptions. For instance, one can decide that constant movement is no longer a valid assumption, e.g. by saying that when d_j in moved by Δ positions, the documents in C_j are moved by $\Delta - \sigma$, where σ is a variable calculated on the basis of the documents rank or similarity score. The similarity-based movement does exactly so by changing the way in which the new document positions are calculated starting from the Δ value defined for the starting document d_j; the new movement is obtained by substituting the instruction at line 7 of Algorithm 1 with the following:

$$\mathtt{newPos} \leftarrow \left\lceil \mathtt{oldPos} * \left(1 - \frac{\mathtt{sPos} - \mathtt{tPos}}{\mathtt{sPos}} * \mathtt{C}_{\mathtt{d_j}}(\mathtt{i},2)\right)\right\rceil$$

We can see that the new position of a document d_i is weighted by two terms $\mathtt{C}_{\mathtt{d_j}}(\mathtt{i},2)$ which is the normalized similarity of d_i to d_j in the cluster C_{d_j} and $\frac{sPos-tPos}{sPos}$ which determines the relative movement of the starting document d_j. Every document d_i is moved by the same increment of d_j (e.g. 1) weighted by the normalized similarity of d_i in the cluster. Basically, d_j, which has similarity 1 by definition, is always moved by the number of positions indicated by the user, whereas the other documents in the cluster are moved by a number of position depending on the similarity to d_j: the higher it is the bigger the movement upward.

Algorithm 2: REORDERLIST

```
Input: The ranked list $L_t$, the old position oldPos of $d_j$, the new position newPos of $d_j$ and the document $d_j$ to be
        moved.
Output: The reordered ranked list $L_t'$.
if newPos $< 1$ then
    newPos $\leftarrow 1$
if newPos $> 1$ then
    chunk1 $\leftarrow L_t(1 : \texttt{newPos}, :)$
else
    chunk1 $\leftarrow [\,]$
end
if oldPos $> \textsc{size}(L_t)$ then
    chunk2 $\leftarrow L_t(\texttt{newPos} : \textsc{size}(L_t), :)$
else
    chunk2 $\leftarrow L_t(\texttt{newPos} : \texttt{oldPos} + 1, :)$
end
if oldPos $> \textsc{size}(L_t)$ then
    chunk3 $\leftarrow L_t(\texttt{oldPos} + 1 : \textsc{size}(L_t), :)$
else
    chunk3 $\leftarrow [\,]$
end
$L_t' \leftarrow$ chunk1
$L_t' \leftarrow \texttt{add}(L_t', d_j)$
$L_t' \leftarrow \texttt{add}(L_t', \texttt{chunk2})$
$L_t' \leftarrow \texttt{add}(L_t', \texttt{chunk3})$
return $L_t'$
```

5 Experimental Evaluation of the Analytical Framework

VATE^2 is expected to work as follows: the user examines a bugged system S_B, identifies the cause of a possible failure and makes a hypothesis about how the fixed version of the system S_F would rank the documents by dragging the spotted document in the expected position.

To conduct an experiment in a controlled environment which accounts for this behaviour, we start from a properly working IR system S_F and we produce a "bugged" version of it S_B by changing one component or one specific feature at a time. Then, we consider all the possible movements that move a relevant document from a wrong position in S_B to the correct one in S_F and we count how many times the prediction of VATE^2, i.e. improvement or deterioration of the performances, corresponds to the actual improvement or deterioration passing from S_B to S_F.

The system we used is Terrier ver. 4.0[6][13], an open source and widely used system in the IR field which is developed and maintained by the University of Glasgow. To run the experiment, we used a standard and openly available experimental collection $\mathscr{C}$, the TREC 8, 1999, Ad-Hoc collection [17].

We experimented the use case about the stemmer and we setup the Terrier system with four different stemmers by keeping all the other components fixed, namely: Porter [14] stemmer, Weak Porter stemmer, Snowball stemmer and no stemmer.

We considered those pairs of systems S_B and S_F, which correspond to sensible and useful cases in practice, i.e. when you pass from a lesser performing system S_B to a better one S_F, as for example when you pass from no stemming S_B to stemming S_F.

[6] `http://terrier.org/`

For each topic, we identified the set of all the possible predictions TP, where each misplaced relevant document d_j in L_{B_t} is moved upwards along with its cluster C_{d_j} to the correct position determined by its rank in the fixed ranked list L_{F_t}, thus generating a predicted ranked list L_{P_t}.

We computed the DCG for each of the above ranked lists: $\mathrm{DCG}_{L_{B_t}}$ and $\mathrm{DCG}_{L_{F_t}}$ indicate the DCG of the bugged system S_B and fixed system S_F while $\mathrm{DCG}_{L_{P_t}}$ indicates the DCG of the predicted system S_P for the i-th possible movement in TP.

We consider a prediction by VATE^2 to be correct if a performance improvement (or deterioration) between the actual bugged system S_B and the fixed one S_F corresponds to a performance improvement (or deterioration) between the actual bugged systems S_B and the predicted one S_P. Let $\mathrm{sgn}(x) = 1$ if $x \geq 0$ and $\mathrm{sgn}(x) = -1$ if $x < 0$; then for each possible prediction $p \in TP$ we define the *Correct Prediction* (CP) measure as:

$$CP_t[p] = \left| \frac{\mathrm{sgn}\left(\mathrm{DCG}_{L_{F_t}} - \mathrm{DCG}_{L_{B_t}}\right) + \mathrm{sgn}\left(\mathrm{DCG}_{L_{P_t}} - \mathrm{DCG}_{L_{B_t}}\right)}{2} \right|$$

Lastly, we define the *Prediction Precision* (PP) for topic t as the number of correct predictions over the total number of possible predictions: $PP_t = \frac{1}{|TP|} \sum_{p \in TP} CP_t[p]$

PP_t ranges between 0 and 1, where 0 indicates that no correct prediction has been made and 1 indicates that all the predictions were correct.

In Table 1 we report the mean DCG value (DCG is calculated topic by topic and then it is averaged over all 50 topics) for the four different stemmers. We can see that there are substantial differences between the systems, in particular the “Weak Porter” and the “No Stemmer” systems have much lower performances with respect to the best one which is the “Porter” system. In Table 2 we report the results of the tests in terms

Table 1: DCG averaged over all the 50 topics of the systems considered for the stemmer failure family.

	Porter	**Weak Porter**	**Snowball**	**No Stemmer**
DCG	105.57	65.63	103.03	59.92

of Prediction Precision (PP) averaged over all the topics for the considered pairs of systems. We can see that the PP is in general satisfactory and it is higher for those pairs where the difference in DCG is higher – e.g. S_B = No Stemmer and S_F = Snowball.

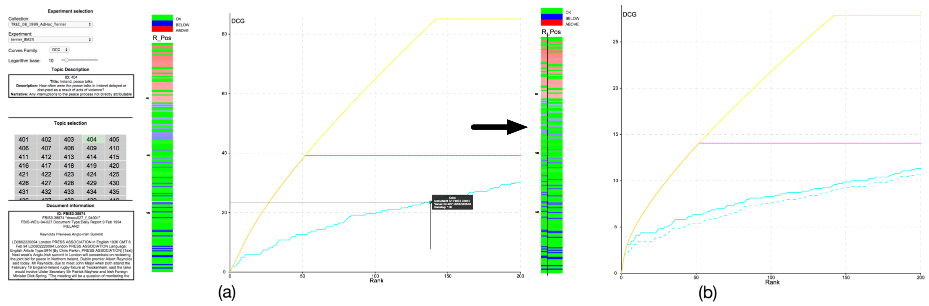
Even if the constant movement behaves better than the similarity-based one in 3 out of 5 considered cases, there is no clear evidence that one of the two movements performs better since they are not significantly different from the statistical point of view according to Student’s t test [16] which returns a p-value $p = 0.9459$. Therefore, in the running implementation of VATE^2, we decided to use the constant movement in VATE^2 because its behavior is more intuitive to the users.

6 Visual Analytics Environment

In Figure 1(a) we can see an overview of VATE^2 system which is available at the URL: `http://ims-ws.dei.unipd.it/vate_ui/` and a video is available here: `http://`

Table 2: Prediction Precision of VATE^2 averaged over all the 50 topics. S_B indicates the bugged system; S_F indicates the fixed system; TP indicates the total number of predictions; *cm* indicates the constant movement and *sbm* the similarity-based one.

$\mathbf{S_B}$	$\mathbf{S_F}$	**TP**	**PP (cm)**	**PP (sbm)**
No Stemmer	Porter	442	.5659	.6047
No Stemmer	Snowball	279	.7106	.7278
No Stemmer	Weak Porter	410	.6694	.6648
Weak Porter	Porter	475	.6283	.6014
Weak Porter	Snowball	436	.6385	.6093



Experiment selection

Collection: TREC_08_1999_AdHoc_Terrier

Experiment: terrier_BM25

Curves Family: DCG

Logarithm base: 10

Topic Description

ID: 404
Title: Ireland, peace talks
Description: How often were the peace talks in Ireland delayed or disrupted as a result of acts of violence?
Narrative: Any interruptions to the peace process not directly attributable

Topic selection

401	402	403	404	405
406	407	408	409	410
411	412	413	414	415
416	417	418	419	420
421	422	423	424	425
426	427	428	429	430
431	432	433	434	435

Document information

ID: FBIS3-38874
FBIS3-38874 "drweu027_f_94001"
FBIS-WEU-94-027 Document Type:Daily Report 9 Feb 1994
IRELAND

Reynolds Previews Anglo-Irish Summit

LD0802220094 London PRESS ASSOCIATION in English 1936 GMT 8 Feb 94 LD0802220094 London PRESS ASSOCIATION Language: English Article Type:BFN [By Chris Parkin, PRESS ASSOCIATION] [Text] Next week's Anglo-Irish summit in London will concentrate on reviewing the joint bid for peace in Northern Ireland, Dublin premier Albert Reynolds said today. Mr Reynolds, due to meet John Major when both attend the February 19 England-Ireland rugby fixture at Twickenham, said the talks would involve Ulster Secretary Sir Patrick Mayhew and Irish Foreign Minister Dick Spring. "The meeting will be a question of monitoring the

Fig. 1: (a) selection of a document and highlight of its cluster; (b) the ranked list and the DCG curve after the movement.

`ims.dei.unipd.it/video/VATE/sigir-vate.mp4`. VATE^2 functioning has been described in details in [5].

We can see that the system is structured in three main components. The **Experimental collection information (A)** which allows the user to inspect and interact with the information regarding the experimental collection. More in detail, it is divided into three sub-components. The first is the "Experiment Selection" where the user can select the experimental collection, the experiment to analyze and the evaluation measure and its parameters. The second sub-component is the "Topic Information" composed of the structured description of the topic and the topic selection grid. The third sub-component is the "Document Information" reporting the content of the document under analysis.

The **Ranked list exploration (B)** which is placed on the center and shows a visual representation of the ranked list. More in detail, the documents are represented as rectangles ordered by rank from top to bottom where the color indicates the *RP* value. The intensity of the color encodes the severity of the misplacement, the more intense the worse the misplacement. This visualization provides the user with an intuitive insight into the quality of the ranking.

The **Performance view (C)** which is placed on the right side and shows the performance curves of the selected experiment. The yellow curve is the ideal one, the magenta curve is the optimal one and the cyan curve is the experiment one. The user can analyze the trend of the experiment by comparing the behavior of its curve with the ideal and optimal ranking by spotting the possible areas of improvement.

The user can interactively select the topic to be analyzed in the topic selection grid and the ranked list and the performance curves are updated accordingly to the selected topic for the given experiment. The ranked list can be dynamically inspected by hovering the mouse over the documents. Moreover, the user can interact with the "Performance view" by hovering the mouse over the curves which, by means of a tooltip, reports information about the document and the performance score.

As shown in Figure 1, concerning what-if analysis, once the user selects a document, the system displaces on the right the rectangles corresponding to the documents in its similarity cluster and reports their identifiers also on the right. Once the user selects a document, s/he can drag it to a new position in the ranked list; afterwards, the movement algorithm is triggered and moves the document along with its similarity cluster in the new positions. This action is visually shown to the user and it is represented with an animated movement of the corresponding rectangles to the new positions. After the movement the ranked list is split in two parts: the old ranked list on the left and the new ranked list produced after the movement on the right. In this way the user can visually compare the effects of the movement and see what other documents have been affected by it.

7 Conclusions

In this paper we explored the application of VA techniques to the problem of IR experimental evaluation and we have seen how joining a powerful analytical framework with a proper visual environment can foster the introduction of a new, yet highly needed phase, which is the what-if analysis. Indeed, improving or fixing an IR system is an extremely resource demanding activity and what-if analysis can help in getting an estimate of what is worth doing, thus saving time and effort.

We designed and developed the VATE^2 system which has proven to be robust and well-suited for its purposes from a two-fold point of view. The experimental evaluation has numerically shown that the analytical engine, the *failure hypothesis* and the corresponding way of clustering documents together with the document movement estimation algorithms are satisfactory. The validation with domain experts has confirmed that VATE^2 is innovative, addresses an open and relevant problem and provides an effective and intuitive solution to it.

As future work, we plan to explore what happens when *multiple movements* are considered all together. This will require an extension of the analytical engine in order to account for the possible inter-dependencies among the different movements. Moreover, also the visual analytics environment will require a substantial modification in order to support users in intuitively dealing with multiple movements, interacting with the history of the performed movements and moving back and forth within it.

References

1. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. Visual Interactive Failure Analysis: Supporting Users in Information Retrieval Evaluation. In *Proc. 4th Symposium on Information*

Interaction in Context (IIiX 2012), pages 195–203. ACM Press, 2012.

2. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. Improving Ranking Evaluation Employing Visual Analytics. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. Proceedings of the Fourth International Conference of the CLEF Initiative (CLEF 2013)*, pages 29–40. LNCS 8138, Springer, 2013.
3. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. VIRTUE: A visual tool for information retrieval performance evaluation and failure analysis. *Journal of Visual Languages & Computing (JVLC)*, 25(4):394–413, 2014.
4. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. Visual Analytics for Information Retrieval Evaluation (VAIRË 2015). In *Advances in Information Retrieval. Proc. 37th European Conference on IR Research (ECIR 2015)*, pages 709–812. LNCS 9022, Springer, 2015.
5. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. A visual analytics approach for what-if analysis of information retrieval systems. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, July 17-21, 2016. Accepted for publication*, 2016.
6. C. W. Cleverdon. The Cranfield Tests on Index Languages Devices. In *Readings in Information Retrieval*, pages 47–60. Morgan Kaufmann Publisher, Inc., 1997.
7. N. Ferro, A. Sabetta, G. Santucci, and G. Tino. Visual Comparison of Ranked Result Cumulated Gains. In *Proc. 2nd International Workshop on Visual Analytics (EuroVA 2011)*, pages 21–24. Eurographics Association, 2011.
8. N. Ferro, G. Silvello, H. Keskustalo, A. Pirkola, and K. Järvelin. The Twist Measure for IR Evaluation: Taking User's Effort Into Account. *Journal of the American Society for Information Science and Technology (JASIST)*, 67:620–648, 2016.
9. D. Harman and C. Buckley. Overview of the Reliable Information Access Workshop. *Information Retrieval*, 12(6):615–641, 2009.
10. D. K. Harman. *Information Retrieval Evaluation*. Morgan & Claypool Publishers, USA, 2011.
11. K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, October 2002.
12. J. Kekäläinen and K. Järvelin. Using Graded Relevance Assessments in IR Evaluation. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(13):1120—1129, November 2002.
13. I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.
14. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.
15. T. Sakai. Metrics, Statistics, Tests. In *Bridging Between Information Retrieval and Databases - PROMISE Winter School 2013, Revised Tutorial Lectures*, pages 116–163. Lecture Notes in Computer Science (LNCS) 8173, Springer, 2014.
16. Student. The Probable Error of a Mean. *Biometrika*, 6(1):1–25, March 1908.
17. E. M. Voorhees and D. K. Harman. Overview of the Eigth Text REtrieval Conference (TREC-8). In *The Eighth Text REtrieval Conference (TREC-8)*, pages 1–24. National Institute of Standards and Technology (NIST), Special Publication 500-246, 1999.
18. J. Zhang. *Visualization for Information Retrieval*. Springer-Verlag, 2008.