CLAIRE: A combinatorial visual analytics system for information retrieval evaluation

Marco Angelini[a], Vanessa Fazzini[a], Nicola Ferro[b], Giuseppe Santucci[a], Gianmaria Silvello[b]

[a] *"La Sapienza" University of Rome, Italy.*
[b] *University of Padua, Italy.*

Abstract

Information Retrieval (IR) develops complex systems, constituted of several components, which aim at returning and optimally ranking the most relevant documents in response to user queries. In this context, experimental evaluation plays a central role, since it allows for measuring IR systems effectiveness, increasing the understanding of their functioning, and better directing the efforts for improving them. Current evaluation methodologies are limited by two major factors: (i) IR systems are evaluated as "black boxes", since it is not possible to decompose the contributions of the different components, e.g., stop lists, stemmers, and IR models; (ii) given that it is not possible to predict the effectiveness of an IR system, both academia and industry need to explore huge numbers of systems, originated by large combinatorial compositions of their components, to understand how they perform and how these components interact together.

We propose a *Combinatorial visuaL Analytics system for Information Retrieval Evaluation (CLAIRE)* which allows for exploring and making sense of the performances of a large amount of IR systems, in order to quickly and intuitively grasp which system configurations are preferred, what are the contributions of the different components and how these components interact together.

The CLAIRE system is then validated against use cases based on several test collections using a wide set of systems, generated by a combinatorial composition of several off-the-shelf components, representing the most common denominator almost always present in English IR systems. In particular, we validate the findings enabled by CLAIRE with respect to consolidated deep statistical analyses and we show that the CLAIRE system allows the generation of new insights, which were not detectable with traditional approaches.

Keywords: information retrieval systems evaluation, visual analytics, visual component-based evaluation, grid of points

1. Introduction

Search engines, and *Information Retrieval (IR)* systems in general [17], deal with vague and imprecise user information needs and try to retrieve relevant documents while at the same time suppressing noisy and not relevant ones.

These systems are consisting of "pipelines" of components, like stop lists, stemmers, IR models, and so on, which are stacked together in order to process both documents and user queries and to match them returning a ranked result list of documents in decreasing order of estimated relevance.

Due to the intrinsic vagueness of user queries and to the uncertainty in the matching process, the performance of an IR system in terms of *effectiveness*, i.e., its ability to retrieve (only) relevant documents

Email addresses: angelini@dis.uniroma1.it (Marco Angelini), fazzini.1416850@studenti.uniroma1.it (Vanessa Fazzini), ferro@dei.unipd.it (Nicola Ferro), santucci@dis.uniroma1.it (Giuseppe Santucci), silvello@dei.unipd.it (Gianmaria Silvello)

and rank them higher, cannot be predicted [28] but, instead, you need to experimentally evaluate it, after the system has been already built. Hence, experimental evaluation [31] is a pillar for advancing IR research and state-of-the-art, because it allows researchers and developers to assess, explore, and tune their systems.

An IR system can consist of many alternative components and, since it is not possible to determine the effectiveness of each individual component separately, the only option to measure their impact on the overall performances is to test all the different combinations of such components. This leads to an explosion in the number of cases to be considered, making the space of system combinations very large and complex to explore. This is what typically happens today not only in academia but also in large-scale search companies.

Besides requiring a great deal of effort and resources to be produced, these combinatorial compositions constitute a challenge when it comes to explore, analyze, and make sense of the experimental results with the goal of understanding how different components contribute to the overall performances and interact together. Indeed, to this end, it is typically needed to resort to complex statistical tools requiring a careful experimental design and producing results which call for a considerable extent of expertise to be interpreted.

The main goal of this work is to design and develop a *Visual Analytics (VA)* system, called CLAIRE, which allows researchers and developers to explore combinatorial compositions of IR system components in order to quickly and intuitively understand which combinations perform best under specific criteria, how components behave across a wide range of cases, and how they interact together. Note that our aim is not to analyse the effects of any possible kind of components, such as word compounding, entity extraction, parser or query expansion, just to name a few, which you may find in an operational IR system.

CLAIRE is a remarkably simple, yet powerful, VA system based on multiple coordinated views approach where different views allow the user to explore simultaneously multiple facets of the data while maintaining an overview of the possible system configurations.

We have experimented with the CLAIRE system on an extensive set of $612 \times 6 = 3,672$ systems, arising from the combinatorial composition of several open-source publicly available components such as stop lists, stemmers, and IR models, and run against 6 different public test collections shared by the *Text REtrieval Conference (TREC)* international evaluation initiative; these collections comprise both news search and Web search tasks. This allowed for validating the findings devised by using the CLAIRE system with respect to previous deep statistical analyses conducted on the same test collections [24, 25].

Summarizing, the main contribution of the paper is a VA system designed for analyzing and comparing a complex set of measures associated with a large combinatorial space of IR systems. CLAIRE addresses a fundamental IR evaluation problem for both industry and academia, i.e., the exploration and understanding of a combinatorial space of configurations, and it is characterized by the following key features:

- a seamless visualization of both the solution space parameters and the associated measures;
- the availability of a set of visual and analytical components that allow for making sense and getting insights on a large number of solutions, identifying trends, common patterns, and outliers;
- a simple visual encoding that allows for a quick identification and a better understanding of statistical properties of the analyzed systems, which in traditional IR evaluation are the result of complex and hard to digest statistical analyses.

The paper is organized as follows: Section 2 introduces the application domain, i.e., IR and its evaluation methodology; Section 3 discusses related works; Section 4 presents the experimental setup used by the CLAIRE system and in the validation use cases; Section 5 describes the CLAIRE system; Section 6 presents the validation use cases; finally, Section 7 wraps up the discussion and presents an outlook of future work.

2. Problem Definition

Experimental evaluation [60] is based on the Cranfield methodology [14] which makes use of experimental collections $\mathcal{C} = (D, T, GT)$ consisting of: a set of documents D representing the domain of interest; a set of topics T, which simulate and abstract 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 a whole breadth of performance measures.

Experimental evaluation is a demanding activity in terms of effort and requires 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 25 years, experimental evaluation has been carried out in large-scale evaluation campaigns at international level, such as the *Text REtrieval Conference (TREC)*[1] in the US, the *Conference and Labs of the Evaluation Forum (CLEF)*[2] in Europe, or the *NII Testbeds and Community for Information access Research (NTCIR)*[3] in Japan and the other Asian countries.

Understanding and interpreting the results produced by experimental evaluation is a non-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 consequence, this task is mostly manual and requires huge amounts of time and effort.

However, a limitation of the current experimental methodology is that it allows to evaluate IR systems only as "black-boxes", without an understanding of how their different components interact with each other and contribute to the overall performances. In other terms, the current experimental methodology considers system performances as indivisible and it cannot break them down into the contributions of the different components constituting an IR system. This severe impediment has been pointed out a long time ago by Robertson [55]: "if we want to decide between alternative indexing strategies for example, we must use these strategies as part of a complete information retrieval system, and examine its overall performance (with each of the alternatives) directly".

This limitation has several additional drawbacks: it prevents from gaining a deep comprehension of IR system performances; it precludes the possibility of knowing ahead which mix of components is best suited for a specific search task or collection of documents [28]; and, it hampers the possibility of determining on which components is more convenient to invest effort and resources because they or their combination have the highest impact in terms of performance gains.

The impossibility of testing a single component by setting it aside from the complete IR system is a long-standing and well-known problem in IR experimentation. Component-based evaluation methodologies Ferro and Harman [22], Hanbury and Müller [30] mixed different components in order to avoid to build an IR system from scratch; but, even though these approaches allowed researchers to focus on the components of their own interest, they have not delivered estimates of the performance figures of each component.

A statistical methodology able to address this issue and to allow for decomposing the effects of different components has been proposed in Ferro and Silvello [24, 25]. The methodology is based on *General Linear Mixed Model (GLMM)* and *ANalysis Of VAriance (ANOVA)* [58] and it makes use of a *Grid of Points (GoP)*, i.e., a set of systems originating from all the possible combinations of the targeted components.

The idea of creating all the possible combinations of components has been proposed by Ferro and Harman [22], who noted that a systematic series of experiments on standard collections would have created a GoP, where (ideally) all the combinations of retrieval methods and components are represented, allowing us to gain more insights about the effectiveness of the different components and their interaction; this would have called also for the identification of *suitable baselines* with respect to which all the comparisons have to be made.

A GLMM explains the variation of a dependent variable ("Data") in terms of a controlled variation of independent variables ("Model") in addition to a residual uncontrolled variation ("Error"): Data = Model + Error. In GLMM terms, ANOVA attempts to explain data (the dependent variable scores) in terms of the experimental conditions (the model) and an error component. Typically, ANOVA is used to determine which experimental condition dependent variable score means differ and what proportion of variation in the dependent variable can be attributed to differences between specific experimental groups or conditions, as defined by the independent variable(s).

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

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

[3] `http://research.nii.ac.jp/ntcir/`

In this context, the focus is both on single independent variables, i.e., the *main effects* of the different components alone, and on their combinations, i.e., the *interaction effects* between components. Note that some independent variables are considered *fixed effects* – i.e., they have precisely defined levels, and inferences about its effect apply only to those levels – which in our case are different kinds of systems and components; and, some others are considered *random effects* – i.e., they describe a randomly and independently drawn set of levels that represent variation in a clearly defined wider population – which in our case are the topics.

A *Type I* error occurs when a true null hypothesis is rejected and the significance level α is the probability of committing a Type I error. When performing multiple comparisons, the probability of committing a Type I error increases with the number of comparisons and we keep it controlled by applying the Tukey *Honestly Significant Difference (HSD)* test [35] with $\alpha = 0.05$. Tukey's method is used in ANOVA to create confidence intervals for all pairwise differences between factor levels, while controlling the family error rate.

The *main effects plot*, as in Figure 12, graphs the response mean for each factor level connected by a solid line and, by means of this plot, we can easily determine the impact of the different levels of a factor.

An *interaction effects plot*, as in Figure 13, displays the levels of one factor on the X axis and has a separate line for the means of each level of the other factor on the Y axis; it allows to understand whether the effect of one factor depends on the level of the other factor. Two parallel lines indicate that no interaction occurred, whereas nonparallel lines indicate an interaction between factors; the more nonparallel the lines are, the greater the strength of the interaction.

Finally, in the *Tukey HSD plots*, as in Figure 16, each point represents the mean performances of a component. Vertical dotted lines in grey represent the range, according to the Tukey HSD test, within which approaches are not significantly different; blue dots represent approaches in the top group, i.e., approaches not significantly different from the top performing one.

The goal of this paper is to consider the lay of the land in experimental evaluation from a new visual analytics perspective by developing a system, CLAIRE, which allows us not only to intuitively obtain the findings resulting from the complex statistical analyses described above but also to get new ones, which are typically difficult or impossible to obtain with the traditional approaches.

3. Related work

The problem of exploring a large combinatorial space associated with several configuration parameters has long been addressed by VA solutions, with the typical goal of solving (multi-objective) optimization problems. Typical solutions rely on simulations and on the use of ad-hoc linear regression models (see, e.g., [64]) to define the solution space and compute measures relevant to the actual application domain. Measures are visually plotted, allowing the user to explore them, with the goal of finding suboptimal solutions and looking for correlations among them. To this aim, most proposals use the idea of presenting the user with multiple coordinated views (see, e.g., the seminal proposal by Tweedie et al. [68], and more recent approaches coming from the demanding automotive field [45] or fishery [12]), allowing interactive coordinated brushing to explore solutions and relationships among measures. Other proposals, instead, rely on the idea of visually navigating the space associated with multi-objective optimization functions in order to find the desired solution (see, e.g., [19]) or visually validating the selected regression model, (see, e.g., [52]).

Some proposals share the idea of sampling the parameter spaces to cope with high-dimensional domain problems and to reduce the number of simulation runs, (see, e.g., [11], [62]); in order to increase the precision of the solutions space, some systems try to increase the number of sampling points to cover as many of the possible combinations as needed within a continuous parameter space, balancing computational time and approximation generated by the sampling (see, e.g., [10]).

Other solutions, closer to our approach, deal with *categorical* parameters that may reduce the design space cardinality (w.r.t., continuous parameters); however, they still need effective means for representing their categorical values and the associated relevant measures. Some proposals deal with *On-Line Analytical Processing (OLAP)* applications, but typically they do not deal with the issue of providing an overview of the data by focusing on the visual inspection of the result of a query or on the visual query specification and exploratory analysis, see, e.g., [44]. With a different approach, Padua et al. [50] use decision trees for

exploring categorical decisions, while others still rely on multiple coordinated views to represent categorical parameters (see, e.g., [45]) or use bi-dimensional data projections (see, e.g., [69]). Other solutions focus on parallel exploration interfaces, investigating means to allow designers to work at the same time with multiple design variations (see, e.g., [63]). However, such solutions fail to scale with the cardinality of the possible combinations that is one of the issue this work deals with; indeed, the composition of the available open-source IR components when applied to several collections generates thousands of possible solutions and CLAIRE has been designed to allow the user to interactively navigate this complex solution space, explicitly keeping track of the components used to build each system.

Indeed, one of the main difference between CLAIRE and the aforementioned proposals is the way in which it deals with the design space itself, the *Grid of Points (GoP)*, that contrasts the widely adopted approach that allows the user to deal with the *measures* associated with each configuration and to look for correlations among them and (sub-)optimal configurations, disregarding the configurations that are behind them. CLAIRE, instead, in order to fulfill the task of comparing configurations, looking for trends, similarities, and outliers, treats configurations as first class objects, allowing the user to inspect the difference in measures associated with the solutions while keeping track of the configurations that generated them. An additional issue CLAIRE has to deal with is that collected measures depend not only on the selected open source components but also on the actual inspected collection and topics; this adds an additional level of complexity to the analysis process. Such a complexity is further increased by the consideration that CLAIRE deals with families of measures computed by aggregating values coming from value distributions, e.g., calculating the confidence interval and the mean of a selected measure computed over hundreds of topics.

CLAIRE addresses the necessity to analyse large combinations of system components due to the proliferation of open source IR systems [65] which allow researchers to easily run systematic experiments. Indeed, the community started to investigate *reproducible baselines* [20, 21, 40]. For instance, Trotman et al. [66] conducted a vertical exploration of variations of two IR models while the "Open-Source Information Retrieval Reproducibility Challenge" [8, 40] provided several reproducible baselines over TREC and CLEF collections. Overall, both these efforts added a few points to the GoP mentioned above, but they do not propose any methodology for estimating the component effects. On the other hand, CLAIRE leverages on a new methodology which allowed a better estimation of component effects and produced a much more fine-grained GoP both in terms of the number of components and IR models experimented [24, 25].

VA techniques are typically exploited for the presentation and exploration of the *documents* managed by an IR system [72]. Some examples are: identification of the objects and their attributes to be displayed [27]; different ways of presenting the data [48]; the definition of visual spaces and visual semantic frameworks [71]. The development of interactive means for IR is an active field which focuses on search user interfaces [32, 33], displaying of results [16] and browsing capabilities [38].

However, much less attention has been devoted to applying visual analytics techniques to the analysis and exploration of the performances of IR systems; the way in which visual analytics can help the interpretation and exploration of system performances has been explored by Ferro et al. [23]. This preliminary work led to the development of *Visual Information Retrieval Tool for Upfront Evaluation (VIRTUE)*, a fully-fledged visual analytics prototype which specifically supports performance and failure analysis [4] dealing with large-scale evaluation campaigns, a context in which evaluators do not have access to the tested systems but they can only examine the final outputs.

Therefore, in Angelini et al. [5] is presented an analytical framework 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, while Angelini et al. [6] presents a formal and structured way to explore the complex data set of measures produced along an evaluation campaign. Similarly, an information visualization Web-based integrated information retrieval performance evaluation platform for the evaluation of an IR system is presented in Ioannakis et al. [36]; however, it does not explicitly focus on comparison among different IR systems. Recently, Lipani et al. [41] presented an information visualization system to explore pooling strategies to build the ground truth of a test collection.

However, even if the above information visualization and visual analytics approaches successfully dealt with the goal of supporting the evaluation and comparison of IR systems, to the best of our knowledge

no solution exists dealing with large sets of IR systems generated by the combinatorial composition of IR components allowing the inspection of both configurations and measures.

4. Experimental Setup

The following sections describe the experimental collections, the *Grid of Points (GoP)* and the evaluation measures we used to showcase and validate the CLAIRE system.

4.1. Collections

We considered the following standard and shared collections, each track using 50 different topics:

- *TREC Adhoc tracks* `T07` *and* `T08`: they focus on a news search task and adopt a corpus of about 528K news documents.
- *TREC Web tracks* `T09` *and* `T10`: focus on a Web search task and adopt a corpus of 1.7M Web pages.
- *TREC Terabyte tracks* `T14` *and* `T15`: focus on a Web search task and adopt a corpus of 125M Web pages.

4.2. Grid of Points (GoP)

We considered three main components of an IR system: stop list, stemmer, and IR model. We selected a set of alternative implementations of each component and, by using the Terrier v.4.0[4] open source system, we created a run for each system defined by combining the available components in all possible ways. The selected components are:

- *Stop list*: `nostop`, `indri`, `lucene`, `snowball`, `smart`, `terrier`;
- *Stemmer*: `nolug`, `weakPorter`, `porter`, `snowballPorter`, `krovetz`, `lovins`;
- *Model*: `bb2`, `bm25`, `dfiz`, `dfree`, `dirichletlm`, `dlh`, `dph`, `hiemstralm`, `ifb2`, `inb2`, `inl2`, `inexpb2`, `jskls`, `lemurtfidf`, `lgd`, `pl2`, `tfidf`.

Overall, these components define a $6 \times 6 \times 17$ factorial design with a GoP consisting of 612 system runs. They represent nearly all the state-of-the-art components which constitute the common denominator almost always present in any IR system for English retrieval and thus they are a good account of what can be found in many different operational settings.

The stop lists differ from each other by the number of terms composing them; specifically, `indri` has 418 terms, `lucene` has 33 terms, `snowball` has 174 terms, `smart` has 571 terms and `terrier` 733 terms.

Since for the *Lexical Unit Generator* (LUG) component we may have considered two distinct methods, i.e. stemmers and n-grams, we indicate the absence of the stemmer as `nolug` to specify that both no stemmer and no n-gram technique are employed. Stemmers can be classified into aggressive and weak stemmers. One of the first stemmers developed for IR systems is Lovins [43]; this is an iterative affix removal stemmer which removes the longest possible string of characters from a word, according to a set of rules. Lovins is the most aggressive stemmer amongst those we consider. The Porter algorithm [53] and its variants (snowball and a weaker version `weakPorter` that only applies the first two steps of the Porter's algorithm by focusing on plurals and suffixes) is inspired by the Lovins algorithm, but it adds well-defined rules for morphology; Porter-based stemmers are weaker than Lovins.

The Krovetz algorithm [39] adds a word-disambiguation algorithm to the Porter stemmer in order to stem the words which not only have a similar morphology, but also a similar meaning; basically, it is as aggressive as Porter and weaker than Lovins. experiments.

[4] `http://www.terrier.org/`

The models we employ are classified into the three main approaches currently adopted by search engines [57]: (i) the vector space model [59]: TFIDF and LemurTFIDF (the Lemur variant of tf*idf); (ii) the probabilistic model – including the BM25 models [56] and the *Divergence From Randomness (DFR)* models [3], and in particular[5]: BB2 (Bose-Einstein model for randomness), DFIZ, DFRee (hyper-geometric model which takes an average of two information measures), DLH (parameter free hyper-geometric DFR model), DPH (hyper-geometric DFR model with Popper's normalization), IFB2 (Inverse Term Frequency model for randomness using the ratio of two Bernoulli's processes for normalisation), InL2 (IDF model for randomness using Laplace succession for normalisation), InexpB2 (Inverse expected document frequency model for randomness using the ratio of two Bernoulli's processes for normalisation), PL2 (Poisson estimation for randomness using Laplace succession for normalisation), and InB2 (a variant of the PL2); and, (iii) the language models [70], and in particular: DirichletLM (Language Modelling with Bayesian smoothing and a Dirichlet Prior), HiemstraLM (Hiemstra's language model [34]), Js_KLs (Jeffreys' divergence with the Kullback Leibler's divergence [42]) and the LGD loglogistic model [15]. Note that the parameters of the models employed are set to their default values as predefined by the Terrier system; the parameters setting may have a sizable impact over a model performances, thus they could be object of a dedicated GoP and component-based evaluation.

To give a feeling of how valuable these GoP are, consider that their preparation required many weeks of processing on high-performance machines, such as an IBM Power7 with 6 CPUs (3.1 GHz) with 8 cores each and 512 Gbytes of RAM.

4.3. Measures

We evaluate the GoPs by employing 8 different evaluation measures: AP, P@10, Rprec, RBP, nDCG, nDCG@20, ERR[6], and Twist.

Average Precision (AP) [17] represents the "gold standard" measure in IR, known to be stable and informative, with a natural top-heavy bias and an underlying theoretical basis as approximation of the area under the recall-precision curve.

Precision at Ten (P@10) [17] is the classic precision measure with cut-off at the first 10 retrieved documents.

Rprec [17] is precision calculated with cut-off at the recall base – i.e., the total number of relevant documents for a given topic. It is an highly informative measure which shares with AP the geometric interpretation as approximation of the area under the recall-precision curve.

Rank-Biased Precision (RBP) [47] is built around a user model based on the utility a user can achieve by using a system: the higher, the better. The model implied by this measure is that a user always starts from the first document in the list and then s/he progresses from one document to the next with a probability p. We calculated RBP by setting $p = 0.8$ which represent a good trade-off between a very persistent and a remitting user.

Normalized Discounted Cumulated Gain (nDCG) [37] is the normalized version of the widely-known DCG which discounts the gain provided by each relevant retrieved document proportionally to the rank at which it is retrieved. nDCG is defined for graded relevance judgments and it is one of the most common measures used for evaluating Web search tasks. For `T07` and `T08`, we calculate nDCG in a binary relevance setting by giving gain 0 to non-relevant documents and gain 5 to the relevant ones; whereas, for `T09`, `T10`, `T13`, `T14`, and `T15` we assign a weight 0 to non-relevant documents, 5 to the relevant ones and 10 to the highly relevant ones. Furthermore, we use a log_{10} discounting function, which accounts for a reasonably persistent user. nDCG is calculated up to the last relevant retrieved document, whereas nDCG@20 is calculated up to rank position 20.

[5] Additional information about these models can be found in Terrier's documentation available at the URL: `http://terrier.org/docs/v4.2/javadoc/org/terrier/matching/models/package-summary.html`

[6] Due to the strong top heaviness of ERR, ERR@20 produces more or less the same scores as ERR. Therefore, we left it out since it does not add any interesting contribution to correlation analysis.

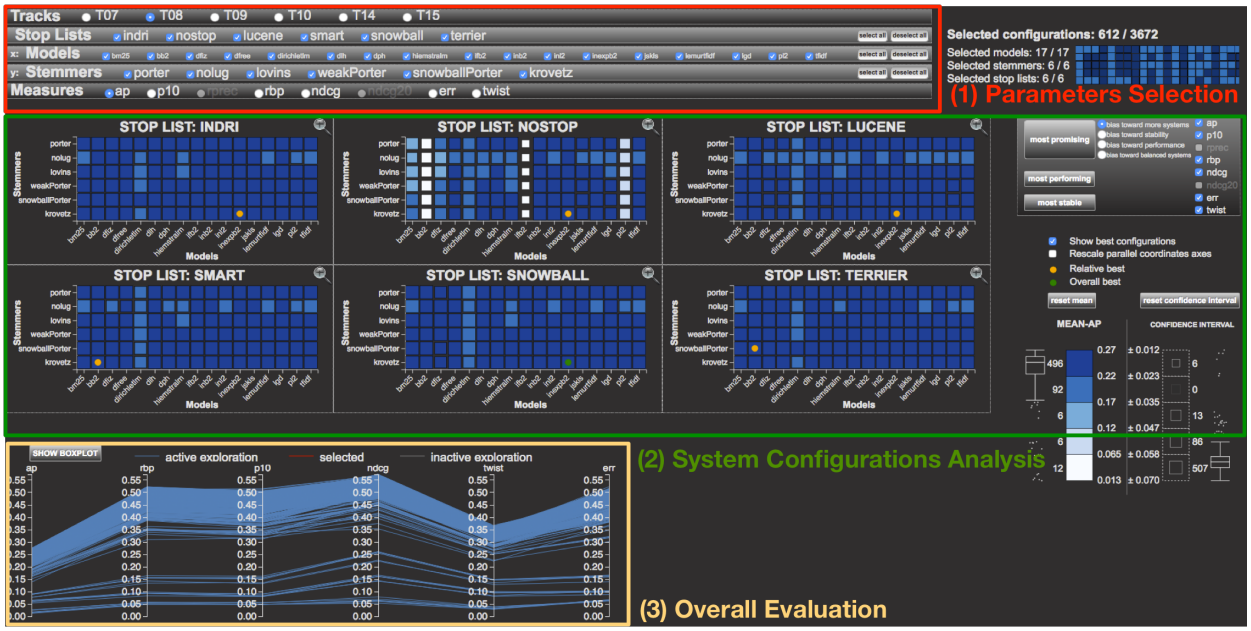


Figure 1: A comprehensive view of the CLAIRE system. We see a sequence of 6 tiles corresponding to the available stop lists; each tile presents the 17 IR models on the x-axis and the 6 stemmers on the y-axis. The selected track is `T08` and the evaluation measure is AP.

Expected Reciprocal Rank (ERR) [13] is a measure defined for graded relevance judgments and for evaluating navigational intents. It is particularly top-heavy since it highly penalizes systems placing not-relevant documents in high positions. For ERR we used the same relevance weights as for nDCG.

Twist [26] is a measure for informational intents, which adopts a user model where the user scans the ranked list from top to bottom until s/he stops, and returns an estimate of the effort required by the user to traverse the ranked list. Twist evaluates systems from the viewpoint of the avoidable effort for their users by accounting for their fatigue while visiting a non-ideal ranking of documents; thus, it evaluates IR systems from a different angle, i.e., user effort, than other measures such as nDCG and ERR which are more focused on user's gain.

Overall, IR evaluation measures embed (possibly quite) different user models and they constitute different ways of scoring systems according to the different viewpoints represented by their user model.

If you consider that we have a GoP consisting of 612 systems run against 6 collections with 50 topics each and we compute 8 evaluation measures, this amount to 1,468,800 data points and provides an idea why it is important to develop a visual analytics system to help us in exploring, comparing, and making sense of them.

5. Description of the system

This section presents the CLAIRE system (available at `http://awareserver.dis.uniroma1.it:11768/claire/`), detailing its visual and analytical components. The design of the system followed agreed solutions in the field of Infovis: the CLAIRE main structure relies on *multiple coordinated views* that, according to Roberts [54], "by utilizing a visualization design environment [enables] the user to examine different representations and also manages their interactions and automatically coordinates operations between views, [in this way, users] may perceive new and insightful relationships and facts from their data." Moreover the selected tiles structure uses small multiples that have been introduced by Tufte [67] who described them starting from resemblance of movie frames: a series of graphics, showing the same combination of variables,

showing changes in another variable. Color scales have been selected using strongly distinguishable colors and using a 5 value multi-hue colorbrewer scale [7] to make small multiple differences quickly graspable.

5.1. The Visual Component

CLAIRE comprises the three main areas shown in Figure 1:

1. The *Parameters Selection* area, dealing with the exploration coordinates, i.e., collections, stop lists, stemmers, IR models, and measures;
2. The *System Configurations Analysis* area, enabling the performance analysis of the system configurations using the actual evaluation measure;
3. The *Overall Evaluation* area, where the system configurations performances are evaluated on the complete set of given evaluation measures.

CLAIRE relies on the multiple coordinated views design, which allows the user to propagate the results of the analysis process steps among all the views.

5.1.1. Parameters Selection area

From the Parameters Selection area the user can customize the various families of components, generating the different system configurations, selecting the track, the evaluation measure, and different subsets of IR models, stemmers, and stop lists.

On the top-right the active number of system configurations is shown, providing numerical anchors during the exploration and maintaining the awareness on how many configurations are under analysis. The user can re-parametrize at any time (including or excluding) instances of different components and changing the mapping between component families (in a drag&drop fashion), i.e., what is presented on the x-axis, what on the y-axis, and what in the different tiles. After each parametrization step, CLAIRE dynamically updates the visualizations contained in the two areas described in the following.

5.1.2. System Configurations Analysis area

In this area the user can analyze the performance of the selected system configurations. The area is organized as follows: the central part presents a sequence of tiles containing a bi-dimensional matrix representation of configurations performances. The mapping of the components with respect to the two axes is managed by dragging & dropping the respective components family in the Parameters Selection Area described above. At the beginning, the default mapping is set to present IR models on the x-axis, stemmers on the y-axis and a stop list per tile. It is possible to alter this mapping in order to explore different combinations of these three main component families, as shown in Figure 2.

Figure 3 shows the systems not using a stop list (`nostop`) following a matrix-like representation, with the IR models on the x-axis and the stemmers on the y-axis. Each combination of IR model and stemmer is identified by a square within the tile; the color scale of the squares ranges from white to deep blue, where white represents a low mean value and deep blue represents a high mean value, averaged over the 50 topics of a track. The size of the square represents the confidence interval (small sizes tied to low values), averaged over the 50 topics of a track. The color and the size of each square are expressed on a discrete 5 values scale (mouse overing shows the actual values). The choice of this visual encoding is driven by the goal of representing many different configurations in a compact visualization.

Because values for all the possible components combinations exist, the resulting tile is regular and dense. This remarkably simple matrix-like layout naturally and effectively highlights trends in the data. As an example, in Figure 3, for the `nostop` component it is clearly visible that systems using the IR model `bb2` – the second column from the left – exhibits low performance (white squares) whatever choice of stemmer is made.

The discrete values represented by the squares in the tiles are made explicit in the interactive legends on the right side of the area. Each legend is composed of a discrete intervals bar encompassing the values of

[7] colorbrewer2.org

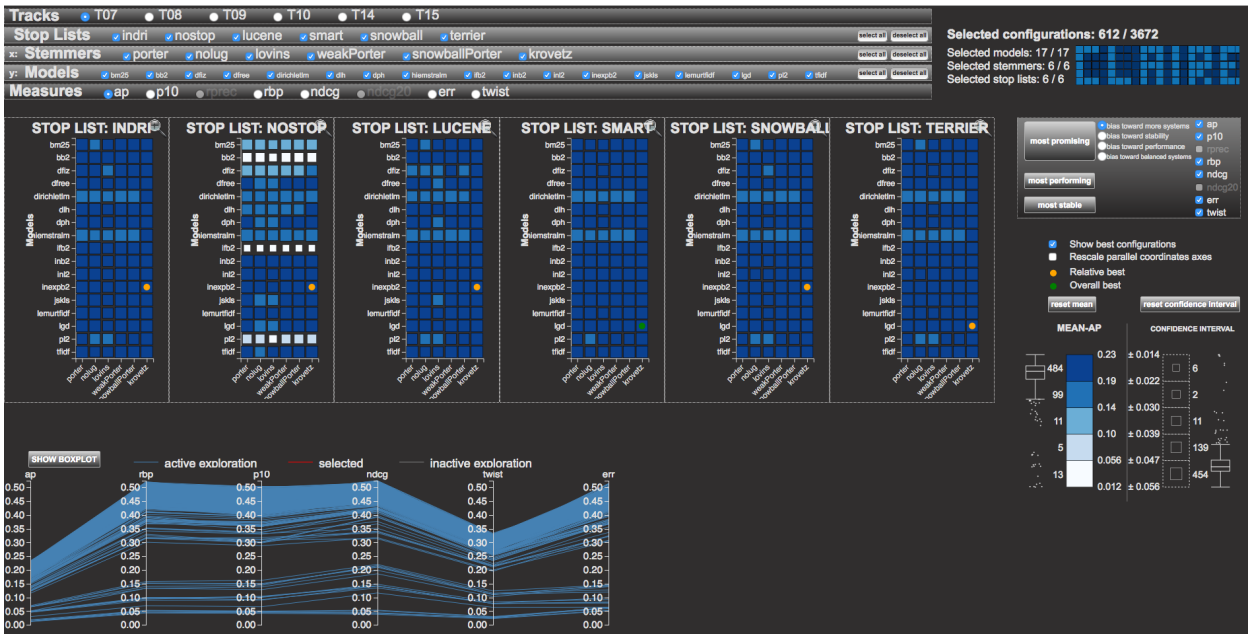


Figure 2: Example of changing the components mapping. In this case, we see that the user mapped the x-axis to the stop lists, the y-axis to the IR models and the tiles to the stemmers. The component mapping is updated by dragging and dropping the component rows in the Parameters Selection area; the environment changes accordingly. With orange circle markers relative best configurations are represented, while a green orange marker represents the overall best.

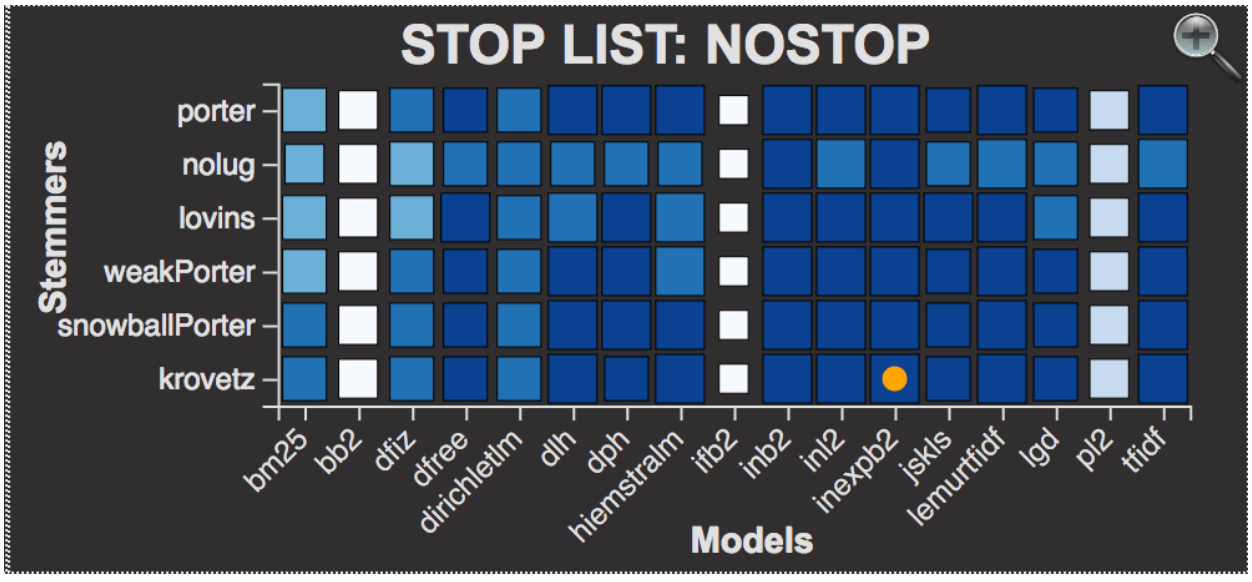


Figure 3: Detail of one tile of the System Configurations Analysis area. The tile is about stop-lists and on the y-axis there are the stemmers and on the x-axis there are the models.

the current selection and an aligned box-plot representing the distribution of all system configurations. The user can interact jointly with both legends and mouse overing a legend interval triggers an automatic filter, which highlights all the systems within that interval. The information about the cardinality of systems comprised within each interval is displayed aside the legend. At any time the user can select one specific system by clicking on the relative square within a tile. This versatile and integrated mechanism allows for defining complex filters and aggregations, resulting in a powerful way for discovering systems strengths and weaknesses.

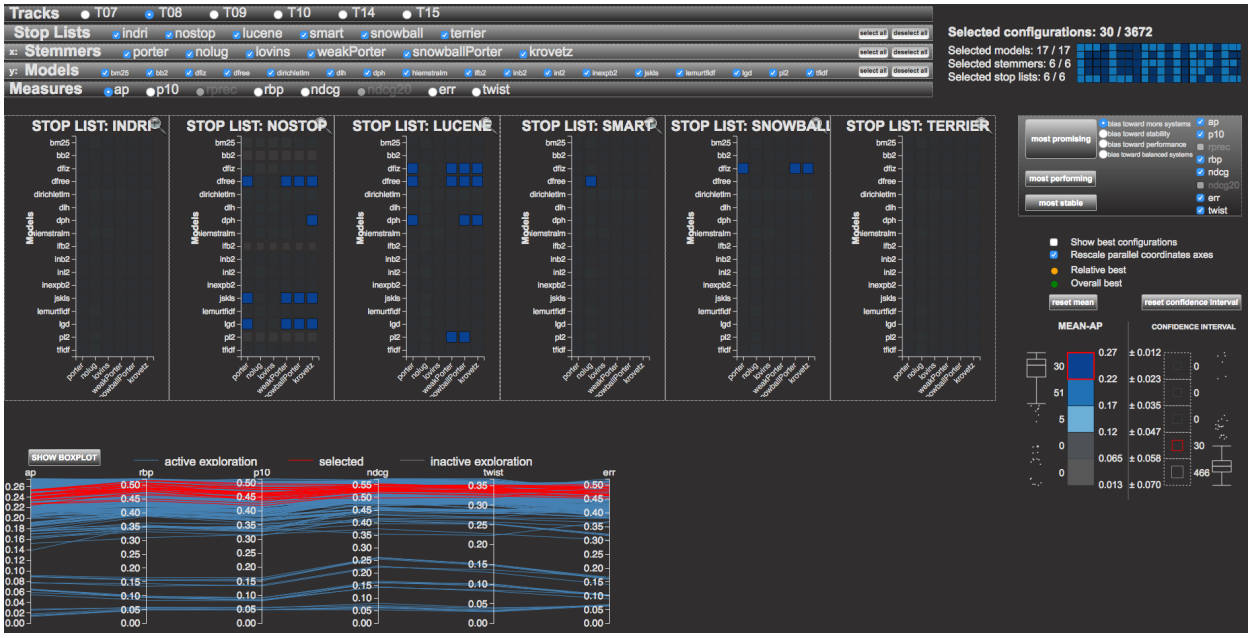


Figure 4: The figure shows the selection of a subset of system configurations based on interaction with the legend. Parallel coordinates axes are rescaled with respect to maximum values on each measure.

As an example, Figure 4 shows an analysis pattern where the user first selects the system configurations with high mean values, i.e., the upper limit of the box-plot on the left. This first selection updates the available intervals in the confidence interval legend. The user then selects the best available confidence interval by using the box-plot on the right, that in this case is the upper quartile.

In Figure 5 we apply two filters, one selecting the upper quartile of mean values and another selecting the third quartile of confidence intervals. In this case the tiles represent the IR models and the applied filter immediately highlights that the `jskls` model is one of the best models characterized by the highest mean with small confidence intervals across different combinations of stop lists and stemmers.

Finally, in order to support data exploration, it is possible at any time to select or deselect elements of the three components families by interacting with the corresponding check-boxes in the Parameters Selection area. In this way, after the selection of the mapping and the filters, the user can evaluate systems by the seamless application of the same analysis process to different solutions at no extra effort. An example is shown in Figure 6, where the user chooses to explore only 12 out of the 17 supported IR models. Visualizations change accordingly, devoting more space to the remaining systems.

While exploring large set of combinations, it is possible to focus on single solutions by clicking on the zoom lens in the upper right corner of a tile. A zoomed in version will be displayed in the lower-right part of the CLAIRE system for better visual exploration as shown in Figure 9.

5.1.3. Overall Evaluation area

The Overall Evaluation area represents each combination as an element in a parallel coordinates view, where each axis represents a different evaluation measure, e.g., AP, RBP, nDCG and so on. The rationale

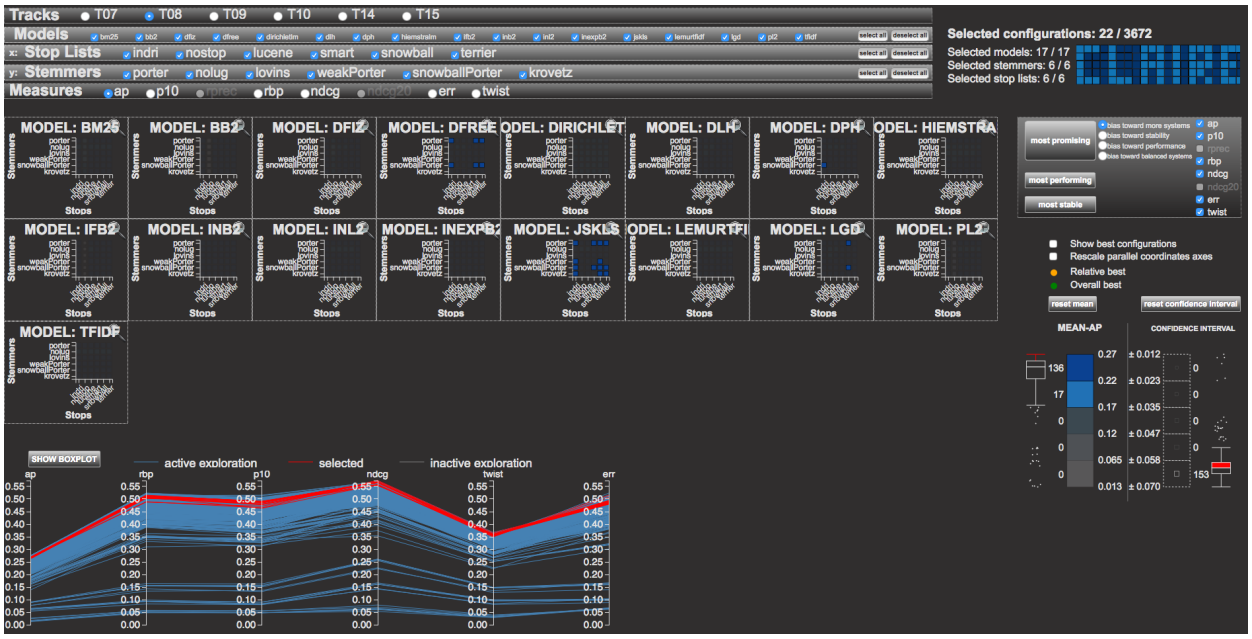


Figure 5: The figure shows the good and robust behavior of the `jskls` model with respect to the other selected configurations.

of this view is to allow the user to relate the good or bad performances of a system configuration on a single evaluation measure, performed in the System Configurations Analysis area, to all the other evaluation measures.

Besides the very common interactions for parallel coordinates – axes reordering, selection by brushing on multiple axes – the CLAIRE system further supports the comprehension of selected elements by allowing the users to superimpose on the axes the distribution of the system configurations represented as box-plots, as shown in Figure 7.

This helps in both relating elements selected from other visualizations to the distribution on all the evaluation measures and improving guidance in the selection by brushing, aligning the brush area to one or more of the quartiles of a box-plot. The analysis process can be started also from this view; indeed by selecting the systems within a particular interval of scores we will see the selection propagated to the System Configurations Analysis area.

5.2. The analytical engine

In order to automate the selection of relevant subsets of systems, CLAIRE uses the available measures to select clusters of similar systems. As an example, assuming that the user wants to explore *most promising* systems (i.e., performing and stable) in terms of AP, the analysis pattern focuses on systems with high AP (AP above the upper quartile) and low confidence interval (AP confidence interval below the lower quartile), i.e., systems with high performance and low variability. We have generalized this analysis pattern, allowing to assess systems using all the available measures and relaxing the constraint of using the highest and lowest quartiles to avoid empty results; as an example, if no systems exist with AP above the upper quartile and confidence interval below the lower quartile, we relax the search by looking for AP above the median or confidence interval below the median and so on, till we get a not empty answer.

More formally, let S be the set of all systems and $M = \langle m_1, ..., m_k \rangle$ the list of available k measures, sorted by importance order according to their usage in IR evaluation analysis, e.g., AP is the most used measure in IR thus it is the most important, followed by nDCG, P@10, ERR, RBP, and twist. For each measure (the higher the better) we have a distribution of $|S|$ values together with their confidence intervals (the lower the better) and we consider the four intervals defined by the distribution box-plots (min-lower

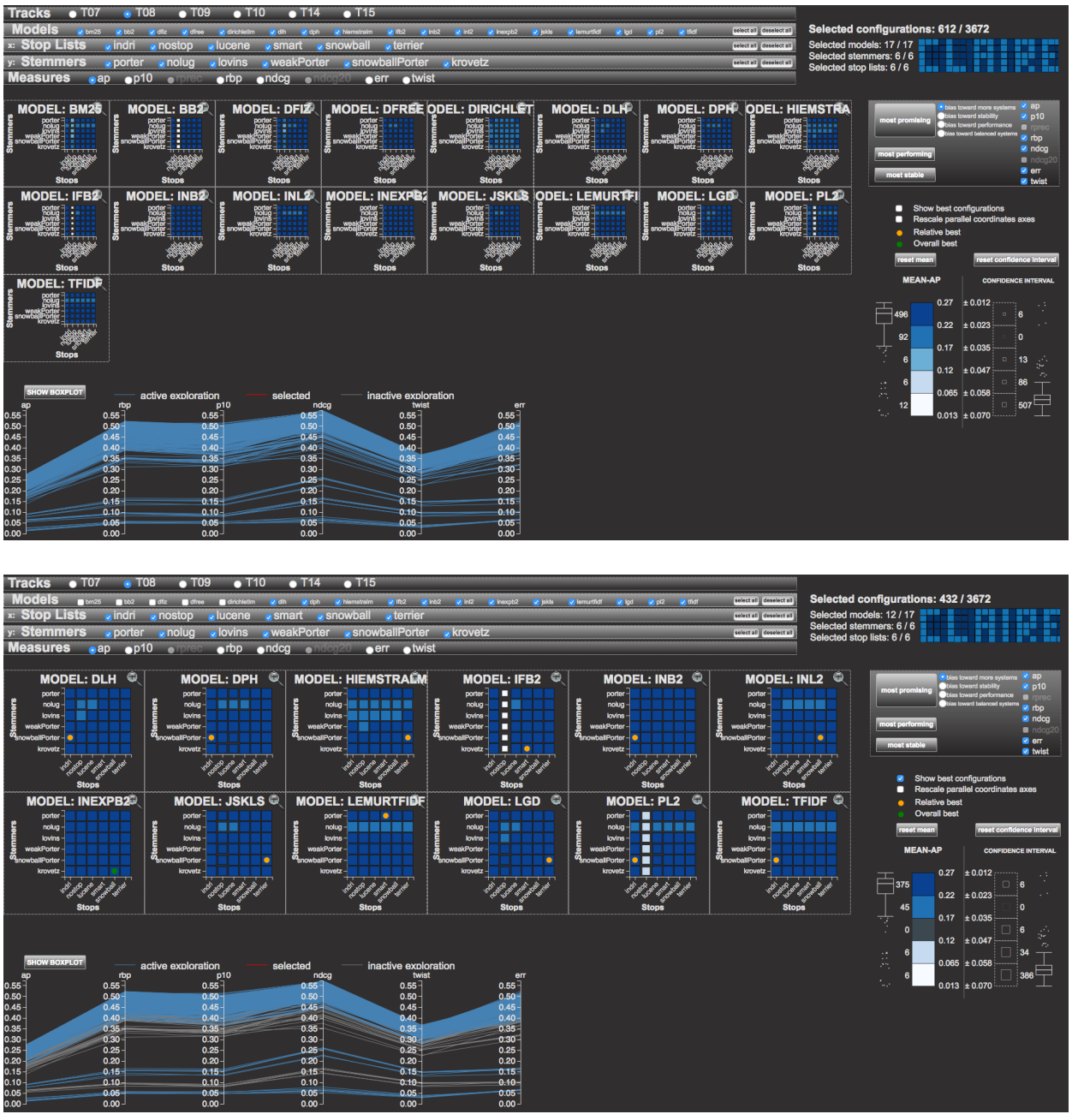


Figure 6: The figure shows the management of the visual space with respect to the number of selected combinations. On the top we see the possible system combinations (corresponding to 612 different systems) while on the bottom, after having deselected 5 not promising IR models, we have the remaining 432 combinations.

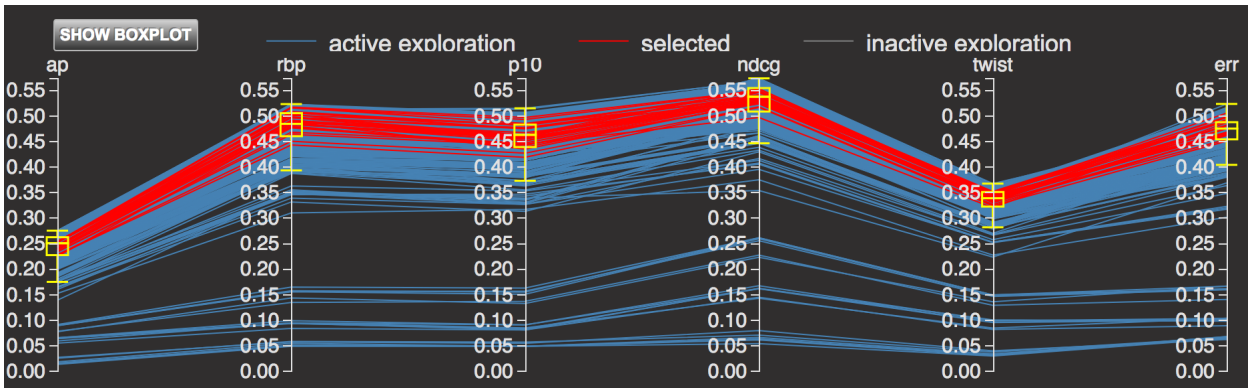


Figure 7: The figure shows a detail of the same selection made in figure 4, this time without rescaling the parallel coordinates axes. Box-plots representing the measures distributions along all the selected systems are superimposed on the parallel coordinates axes. It is visible that the selected subset of systems (in red) is contained in the box area for all the box-plots, denoting comprehensive good relative performances for these system configurations (30) for all the evaluation measures.

quartile, lower quartile-median, median-upper quartile, upper quartile-max). Using such intervals we can define for each measure m two partitions on S. The first partition $\langle s_1, s_2, s_3, s_4 \rangle$ is built by using the four m values distribution box-plot intervals: e.g., s_1 contains the systems that, measured by m, belong to the first box-plot interval (min-lower quartile). Analogously, the second partition $\langle c_1, c_2, c_3, c_4 \rangle$ is built by using the four confidence interval values distribution box-plot intervals. In the most general case, we are interested in systems that are in the intersection of s_i and c_j and we denote such intersection as $m(i,j) = s_i \cap c_j$. The partition index values corresponds to increasing quality: e.g., s_1 corresponds to poor systems with low m values (below the lower quartile) and c_1 corresponds to highly variable systems with high m confidence interval values (above the upper quartile). As an example, the best subset of S, according to m, would be $m(4,4)$; however, if such an intersection is empty we go either for $m(4,3)$ or $m(3,4)$ and so on, progressively lowering i and j till $m(i,j)$ is not empty.

According to this idea, we consider all the 16 combinations (s_i, c_j) to compute the set $D(m)$ of the not empty $m(i,j)$, each of them having a simple quality score $q(m(i,j)) = i + j$ given by the sum of the indexes; so, given the partition index values set $P = \{1,2,3,4\}$:

$$D(m) = \{m(i,j) \mid i,j \in P \ \wedge \ m(i,j) \neq \emptyset\} \tag{1}$$

The quality score q imposes a partial order on the elements in $D(m)$ and those sharing the $max(q)$ value (in general, more than one) represent the highest quality subsets of systems; according to the user input, we either select their union, *bias toward more systems* or we prefer the union of systems in which $i > j$, *bias towards performance*, or in which $i < j$, *bias toward stability*, or $i = j$, *bias toward balanced systems*, see the radio-buttons on the top-right part of Figure 1. In order to evaluate systems using more than one measure, we combine the $D(m_i)$ sets, with $m_i \in M, |M| = k$, in all possible not empty combinations:

$$DM(M) = \{(d_1, \ldots, d_k) \mid \ d_i \in D(m_i) \ \wedge \ \bigcap_{i=1}^{k} d_i \neq \emptyset\} \tag{2}$$

We define the quality score of an element $dm \in DM(M)$ as the sum of the quality of its k components d_i:

$$Q(dm) = \sum_{i=1}^{k} q(d_i) \tag{3}$$

The quality score Q imposes a partial order on the elements in $DM(M)$ and those sharing the $max(Q)$ value represent the highest quality subsets of systems. Thus, we present the user with their union since,

Algorithm 1: Greedy Get Most Promising algorithm

```
M = ordered list of metrics;
k = |M|;
maxQuality = minQuality = 0 ;
Res = ∅;
foreach m in M do
    compute the not empty intersections D(m) (eq. 2);
    POD(m) ← get the Partial Order of D(m) using q(m(i, j));
end
foreach m in M do
    maxQuality = maxQuality + q(POD(m)[0]);
end
minQuality = 2 × k;
currentQuality = maxQuality;
while Res == ∅ do
    if currentQuality < minQuality then
        % DM(M) generated only empty sets;
        remove the less relevant measure m_k from M;
        foreach m in M do
            maxQuality+ = q(POD(m)[0]);
        end
        currentQuality = maxQuality;
        k = k − 1;
        minQuality = 2 × k;
    end
    compute from the k POD(m_i) the combinations DM'(M) that have Q(dm) = currentQuality;
    foreach dm' in DM'(M) do
        compute the intersections of the k d_i I = ⋂_{i=1}^{k} d_i;
        Res = Res ∪ I;
    end
    currentQuality = currentQuality − 1;
end
return Res;
```

when mixing all measures together a finer grain analysis makes no sense. If $DM(M)$ is empty, we iteratively remove the less important measure from M, i.e., $M = M - m_k$, recompute $DM(M)$ using the new M, and so on, till we get a not empty result.

The maximum cardinality of $DM(M)$ is 16^k (much less in real cases); the cost of computing the intersections is $O(n \times log(n))$ in term of number of systems, while computing the sets sharing $max(Q)$ is linear in terms of $DM(M)$ cardinality and that calls for a good scalability of the method; in the actual implementation the response time does not affect the user interaction fluidity. However, in order to better scale with larger IR combinatorial spaces, we have designed a greedy algorithm that computes the partial orders of $D(m_i)$ and use them to build the DM combinatorial space for decreasing quality values starting from $max(Q)$, providing the user with an early result as soon as it returns a non empty set, see algorithm 1. In words, instead of exploring all the possible combinations in a blind fashion, the algorithm starts from those belonging to local optima; if one ore more non empty solutions exist, the algorithm will terminate selecting a (sub) optimal solution that can be used by the user as a starting point, while the system computes better solutions (if any). It is out of the scope of the paper to fully discuss this process: however it is worth mentioning that it belongs to field of Incremental Visualizations (see, e.g., [61]) and we are exploring how to update the initial early result with improved solutions that are computed in the background, estimating the response times and approximation errors figures to include them in the CLAIRE interface, increasing the user understanding of the computation approximation and progress, using techniques that are emerging in the field of Progressive Visual Analytics (see, e.g., [7]).

Moreover, we can run the same process using only measures values, disregarding confidence interval values, i.e., the *most performing* systems or focusing only on confidence intervals, disregarding measure values, i.e., the *most stable* systems.

6. Validation use cases

This section validates CLAIRE by discussing the three main analyses allowed by it – the analysis of IR models, stop lists, and stemmers. We show how CLAIRE leads to the discovery of findings previously obtained by the means of complex statistical analyses and also to new findings, which it is not possible to obtain with the traditional statistical approaches.

6.1. Use case 1: Study of the IR models

In order to analyze the IR models the user needs to change the axes of the tiles from the default settings, dragging the IR model row in the Parameters Selection area on top of the other components so that CLAIRE shows one tile for each model with the stop lists on the x-axis and the stemmers on the y-axis. Collection `T08` and the AP measure are used as the basis for the analysis.

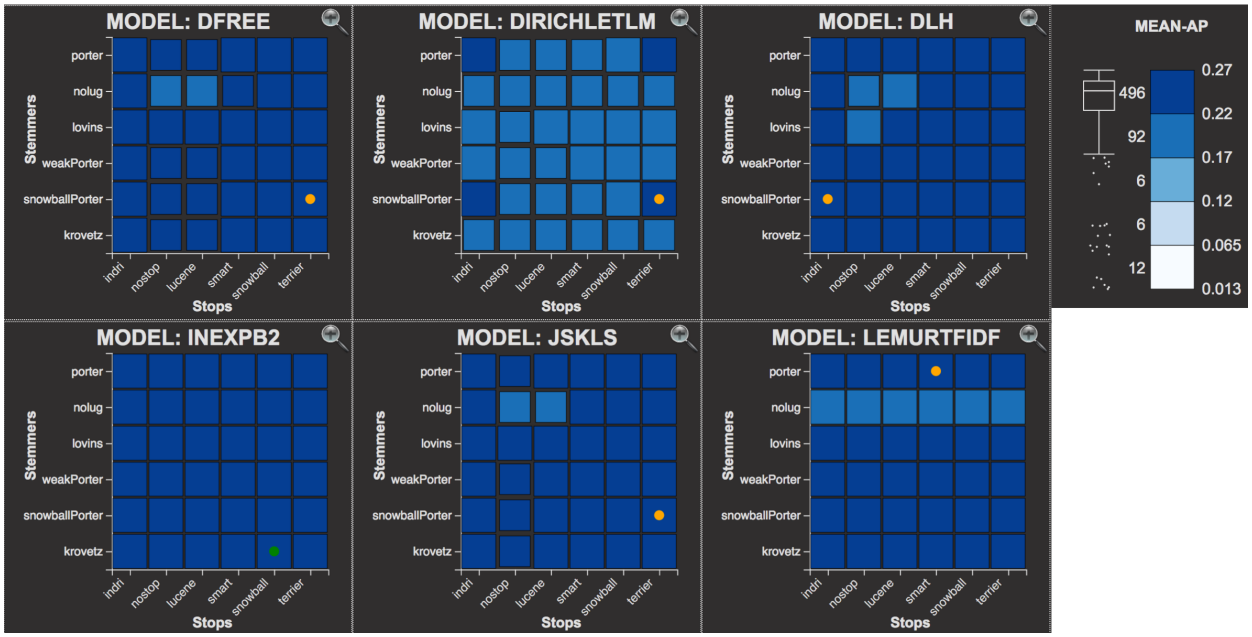


Figure 8: System Configurations Analysis area, encompassing 6 model tiles; stop lists are on the x-axis and stemmers are on the y-axis.

The tiles allow for estimating the performances of each single model at a glance: tiles where darker squares are the majority indicate IR models that perform better than those with a majority of lighter squares.

Figure 8 details the System Configuration Analysis area, showing 6 IR model tiles and the legend (the darker the color the higher the AP). Within these 6 IR models, it is straightforward to assess that `inexpb2` is the best model since all the squares in its tile are dark blue and that `dirichletlm` is the worst model since it presents many light blue squares (*finding 1*).

Moreover, since the squares in the tile are almost color invariant, we understand that this IR model almost does not interact with the other two components, since it performs poorly no matter which stop lists and stemmers are employed by the system (*finding 2*).

We can analyze the `dirichletlm` model more in detail by clicking on the zoom lens in the upper right corner of its tile. CLAIRE presents the user with a bigger model tile placed on the right side of the parallel coordinates plot in the overall evaluation area as shown in Figure 9. By selecting the squares within the tile, CLAIRE highlights in red the lines corresponding to the selected systems in the parallel coordinates view. It is possible to see that all the systems using the `dirichletlm` model are performing poorly consistently across all the evaluation measures; moreover, it is possible to see that all the selected lines in the parallel

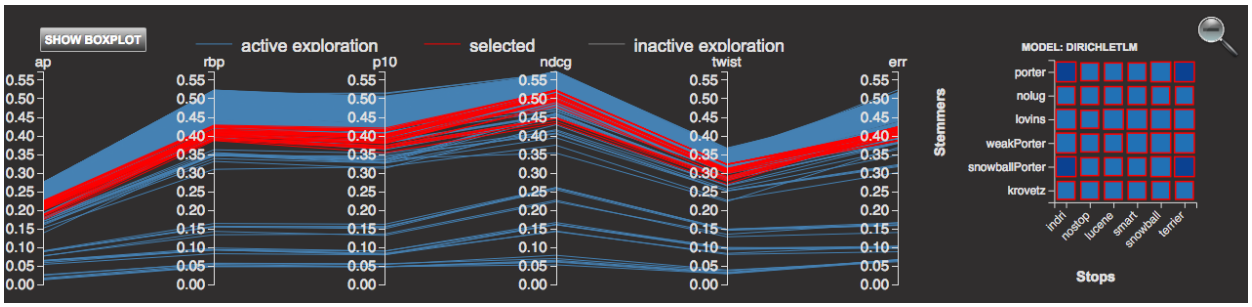


Figure 9: Zoom on the `dircheletlm` model tile and the parallel coordinates plot, where we can see that this IR model performs consistently across different evaluation measures.

coordinates view are within the third and fourth quartile of the square-plots reported on the axes (*finding 3*).

It is easy to verify if this model performs consistently across collections by selecting a different collection on the Parameters Selection area on the top of the screen. It is quite evident that the `dircheletlm` model is consistently one of the worst performing models across all the test collections: `T07`, `T09`, `T10`, `T14` and `T15`; this indicates that the model is not much influenced by the characteristics of the collection at hand (*finding 4*).

By looking at all the model tiles in the System Configurations Analysis area, it is possible to see that there are IR models with problems when not paired up with a stop list – the model tile presents a light column of squares for the `nostop` component – models which need to be paired up with a stemmer – the model tile presents a light row of squares for the `nolug` component – and, models manifesting problems both if they do not work in conjunction with a stop list and a stemmer – the model tile presents a light column of squares for the `nostop` component and a light row of squares for the `nolug` component.

Figure 10 reports 3 model tiles corresponding to the aforementioned visual archetypes: (a) the `bb2` model needs a stop list to function well (*finding 5*), (b) the `tfidf` model works better with a stemmer (*finding 6*), and (c) the `bm25` model suffers the absence of the stop list and also, even though the effect is less marked, the absence of a stemmer (*finding 7*); It is also possible to see that the three models need a stop list and/or a stemmer, but they do not discriminate between different stop lists and stemmers.

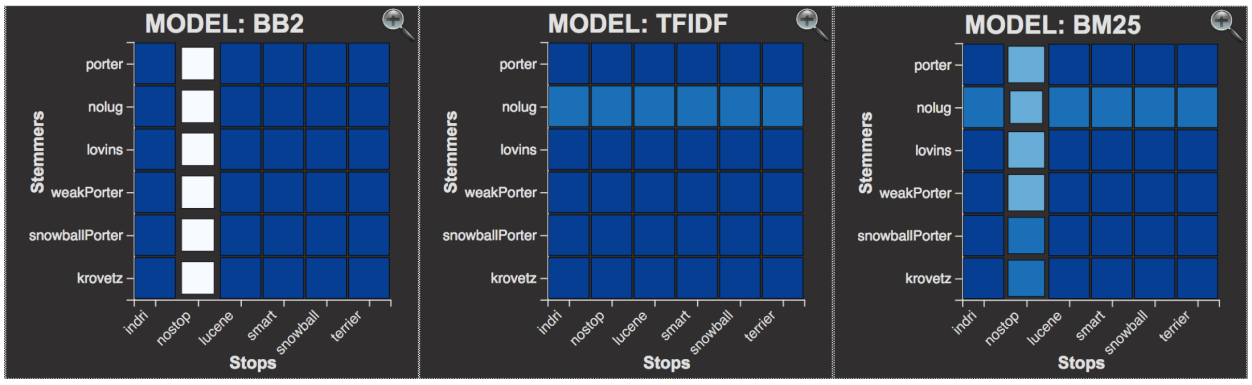


Figure 10: Three visual archetypes that can be identified from the model tiles: (a) a lighter column shows a problem with a stop list; (b) a lighter row shows a problem with a stemmer; (c) a lighter cross shows a problem with both a stop list and a stemmer.

In order to further investigate the measures of the `bm25` model the user explores the enlarged tile and the parallel coordinates plot. He selects the squares composing the light cross on the `bm25` tile and discovers that the problem with the absence of the stop list and stemmer is less marked for the more top-heavy measures

as for instance RBP. So, he examines the tiles for RBP by selecting the measure on the Parameter Selection area and sees that the absence of the stop list is still a problem, but the absence of the stemmer is not a problem anymore. The same goes for the other measures as shown in Figure 11.

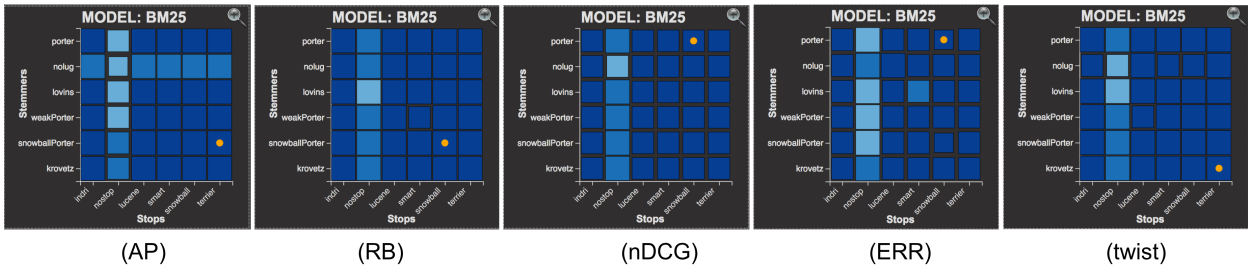


Figure 11: The `bm25` model tiles for different measures. We see that the lighter cross is present only for AP, whereas for more top-heavy measures there is only a lighter column.

This discussion shows how with very few interactions, CLAIRE allows us to quickly and easily make sense of a large amount of data, comparing across several configurations and evaluation measures in one shot. As it will be clarified in the next section, using traditional approaches this would have meant inspecting many different tables and plots to try to recognize some trends.

6.1.1. Statistical Validation

Finding 1: `inexpb2` is the best model and `dirichletlm` is the worst model for `T08` by using AP as evaluation measure.

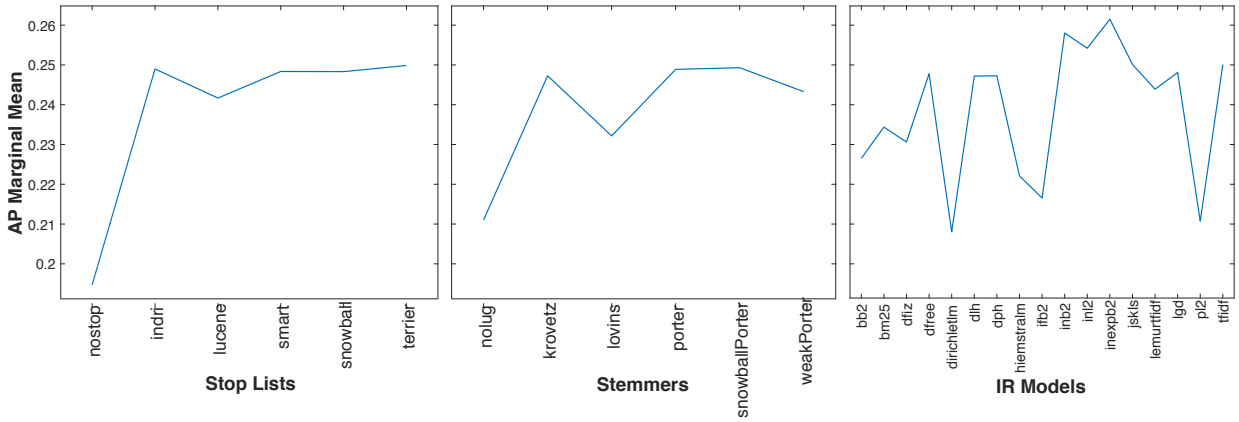


Figure 12: `T08` collection and AP measure main effects plots (see Ferro and Silvello [25]).

Figure 12 shows the main effects plots for the `T08` collection and the AP measure. By looking at the third plot about the model effects, it is possible to see that `dirichletlm` is the worst model and that `inexpb2` is the best one. This statistical evidence confirms the CLAIRE findings.

Finding 2: the `dirichletlm` model has little interaction with stop lists and stemmers.

Figure 13 shows the interaction plots for the `T08` collection and the AP measure. In the two plots showing the `model*stop list` (i.e., plot in row one and column three) and the `model*stemmer` (i.e., plot in row two and column three) interactions, it is possible to see that the lines of the `dirichletlm` model are one close to the other showing that little changes when different stop lists or different stemmers are used. This statistical evidence confirms the finding visually discovered with CLAIRE.

Finding 3: the `dirichletlm` model performs poorly across all the evaluation measures.

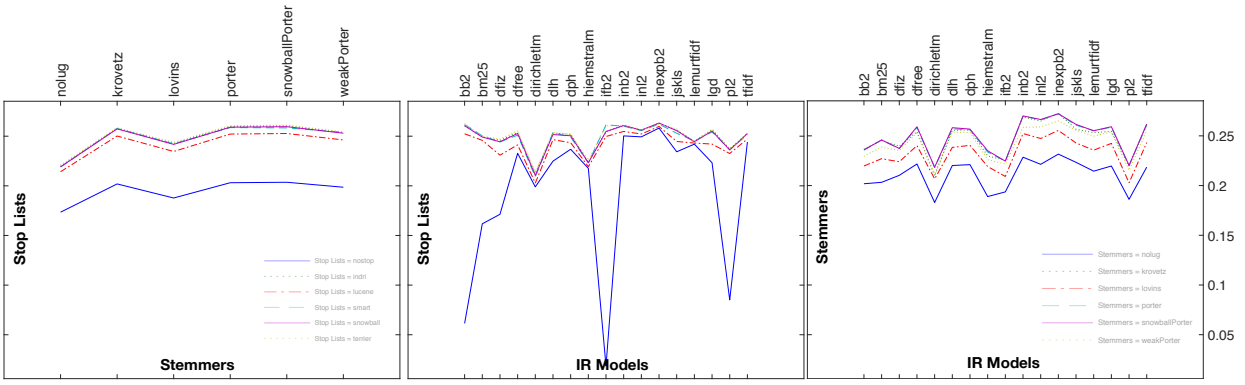


Figure 13: `T08` collection and AP measure interaction plots (see Ferro and Silvello [25]).

To validate this finding a main effects plot for each measure has been generated, checking whether the `dirichletlm` model is amongst the (four) worst performing models in all the cases. This analysis can be validated by consulting the electronic appendix where all the main effect plots and the Tukey HSD test plots for the considered measures based on `T08` are reported.

Finding 4: the `dirichletlm` model performs consistently across all the considered collections.

To validate this finding a separate main effects plot for each collection for the considered measure has been generated, checking whether the `dirichletlm` model is among the worst performing models in all the cases. It is possible to verify that by checking the main effect plots reported in the electronic appendix where we can see that the `dirichletlm` model is often the worst or the second worst model for all the considered collections. This confirms the visual finding obtained by using the CLAIRE system.

Finding 5: the `bb2` model works badly without a stop list and it performs equally well with different stop lists.

By referring to the interaction plot in row one and column three of Figure 13 showing the `model*stop list` interaction, it is possible to see that the `bb2` model performs very poorly when no stop list is applied, whereas it performs much better when a stop list is used; it is also evident that the performances do not change consistently when using different stop lists. This statistical evidence validates the visual discovery obtained by using CLAIRE.

Finding 6: the `tfidf` model works better with a stemmer rather than without it.

By referring to the interaction plot in row two and column three of Figure 13 showing the `model*stemmer` interaction, it is possible to see that the `tfidf` model performs poorly without stemmer, whereas it performs better when a stemmer is used – indeed, the blue line corresponding to the absence of the stemmer is below all the other lines in the plot. This statistical evidence validates the visual discovery obtained by using CLAIRE.

Finding 7: the `bm25` model suffers from the absence of the stop list and the stemmer.

This is an interaction of the third order (`stop list*stemmer*model`): it cannot be visualized by any of the standard statistical plots and it is also hard to determine from a numerical analysis. Indeed, the interaction plots of Figure 13 allows us to say that `bm25` suffers from the absence of a stop list and that it suffers from the absence of a stemmer but not that the joint absences of a stop list and a stemmer is the most critical case. Hence, CLAIRE allows for capturing a facet of the `bm25` model that could not be grasped by the statistical analysis conducted in [25].

6.2. Use case 2: Study of the stop lists

In order to analyze the stop lists the user sets up the axes in the Parameters Selection area to obtain a tile for each stop list with the models on the x-axis and the stemmers on the y-axis.

The tiles allow to estimate the performances of each single stop list at a glance. The tiles where darker squares are the majority perform better than those with a majority of lighter squares, consistently with

what we discussed above for the models.

It is quite straightforward to see that the absence of a stop list is detrimental to many IR models; indeed, as it emerges by looking at the tiles shown in Figure 1, the second tile on the first row presents a higher number of light squares than the other tiles. In particular, it is possible to easily determine that the IR models suffering the most by the absence of a stop list are `bb2`, `bm25`, `dfiz`, `ifb2` and `pl2` just by looking at the columns of light squares within the tile (*finding 1*). The tiles reported in Figure 1 for `T08` allow us to assess the interaction between stop lists and stemmers for a given model. For instance, it is possible to see that the `inb2` and `inexpb2` are not influenced by the choice of the stemmer; indeed, they present a dark blue column in all the tiles, also in the `nostop` case, where many other IR models suffer from the absence of a stop list (*finding 2*).

Even though the `lucene` stop list is the shortest one, its use improves the performances of almost all the models with respect to the `nostop` case, but it suffers from the absence of a stemmer. This is highlighted by the lighter row for the `nolug` case in the `lucene` tile (*finding 3*). There are only three exceptions: `bb2`, `inb2` and `inexpb2`; these are the IR models insensitive to the use of a stemmer as we noted above. Moreover, it is possible to see that the `lucene` stop list performs poorly with any stemmer for the `dirichletlm` model; zooming over the tile and selecting the cross formed by the lighter squares reveals on the parallel coordinates that this behaviour is consistent across all the measures (*finding 4*), as shown in Figure 14.

Figure 14: The `lucene` stop list tile with the lighter cross highlighted and the parallel coordinates view showing that this stop list suffers the absence of a stemmer consistently across evaluation measures.

Overall, we see that the `indri`, `smart`, `snowball` and `terrier` stop lists are very close one to the other, since their tiles present very similar square patterns; they all show a vast majority of dark blue squares and a minority of lighter blue ones (*finding 5*).

Lastly, we check that stop lists behave consistently across collections. By looking at the stop list tiles for `T07`, `T08`, `T09` and `T10` we see that: `T07` and `T08` behave similarly; for `T09` and `T10` the absence of the stemmer has a bigger impact on all the stop lists and we can also see a reduction of the overall variance (especially on `T10`), i.e. the size of the squares. This is shown in Figure 15.

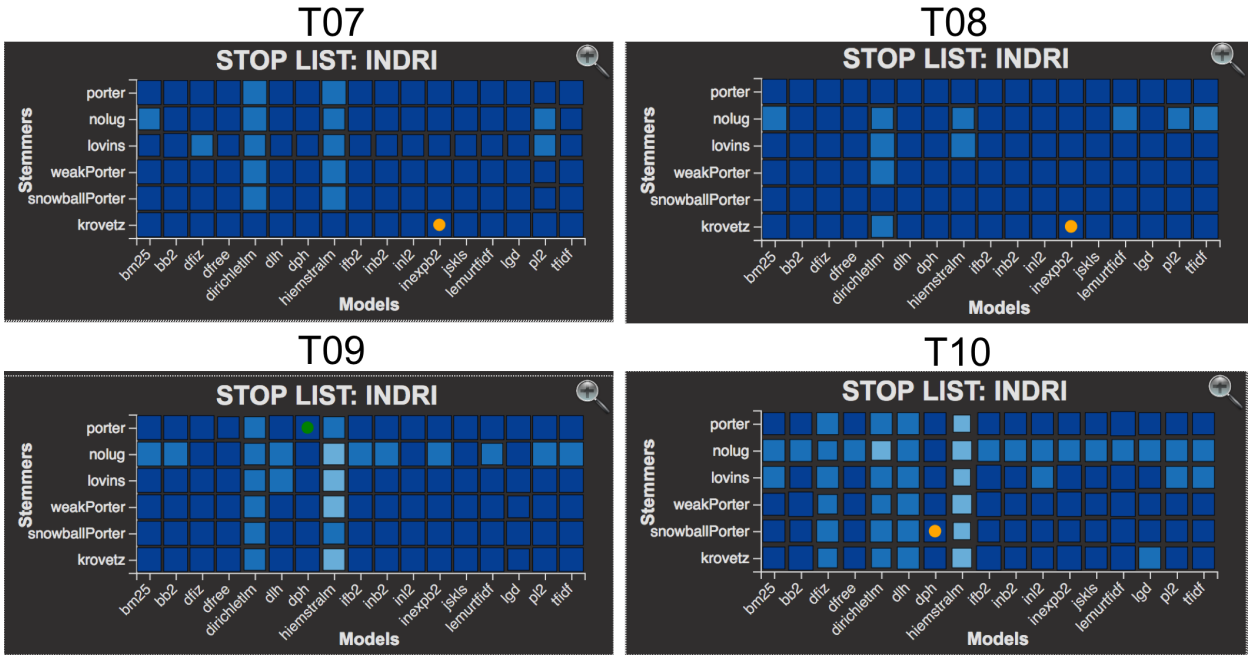


Figure 15: The `indri` stop list tiles of collections `T07`, `T08`, `T09`, `T10`.

6.2.1. Statistical Validation

Finding 1: the ***bb2****,* ***bm25****,* ***dfiz****,* ***ifb2*** *and* ***pl2*** *models require the presence of a stop list to work well.*

In Figure 13 we can see the interaction plots for the `T08` collection and the AP measure. In the plot showing the `model*stop list` (i.e., plot in row one and column three) we can see that the blue line (indicating the `nostop` component) drops for the `bb2`, `bm25`, `dfiz`, `ifb2` and `pl2` models indicating that they suffer from the absence of the stop list. This statistical evidence validate the visual discovery obtained by using the CLAIRE system.

Finding 2: the ***inb2*** *and* ***inexpb2*** *models are not influenced by the choice of the stemmer for whatever stop list is applied.*

This is a third order interaction (`stop list*stemmer*model`) and it is hard to grasp by using the traditional statistical analysis, as previously discussed. Figure 13 shows that the `inb2` and `inexpb2` models are not influenced by the stop list but it is not possible to assess the behaviour of an IR model with respect to different stemmers given a specific stop list as instead it is possible to do by using CLAIRE.

Finding 3: the ***lucene*** *stop list improves the performances for all the models with respect to the* ***nostop*** *case, but it suffers from the lack of a stemmer.*

The first part of this finding can be validated by the `stop list*model` interaction plot in Figure 13 where the red dashed line of the `lucene` stop list is consistently above the blue line of the `nostop` component. The second part of this finding requires an analysis of third order interactions which, as explained above, cannot be visually validated with the traditional statistical analysis as we can do with CLAIRE.

Finding 4: the ***lucene*** *stop list suffers from the absence of a stemmer and it does not perform well for the* ***dirichletlm*** *model no matter of what stemmer is applied.*

This finding requires an analysis of third order interactions, which, as explained above, cannot be validated with the statistical analysis at hand. Hence, CLAIRE allows for having a visual intuition of third order interactions which otherwise could not be grasped.

Finding 5: the ***indri****,* ***smart****,* ***snowball*** *and* ***terrier*** *stop lists can be considered equivalent.*

In the left part of Figure 16, we report the Tukey HSD plot of the stop lists for the `T08` collection and AP measure. It is possible to see that there is no statistical significant differences between the `indri`, `smart`,

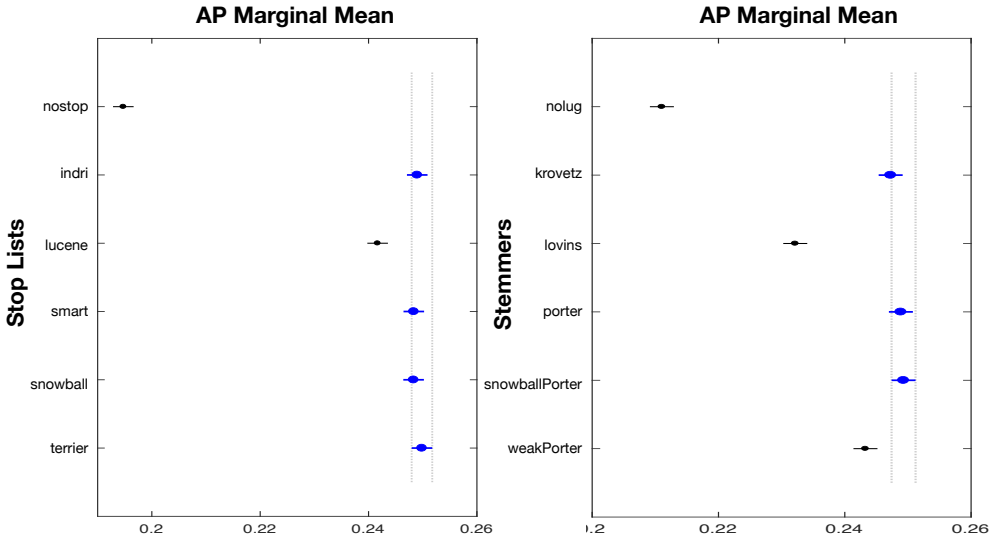


Figure 16: On the left, the Tukey plot of the stop lists for the `T08` collection and AP measure and on the right the Tukey plot of the stemmers.

`snowball` and `terrier` stop lists as visually outlined by CLAIRE.

6.3. Use case 3: Study of the stemmers

In order to analyze the stemmers the axes in the Parameters Selection area are combined to obtain a tile for each stemmer with the models on the x-axis and the stop lists on the y-axis.

Figure 17 shows that the absence of the stemmer is detrimental for the performances of all the systems (*finding 1*) with the solely exception of the `inb2` and `inexpb2` models, as described above.

It is possible to see that the presence of a stemmer has an impact, but the interaction with the other components is low (*finding 2*). Indeed, excluding the two corner cases analyzed above (i.e., the `dirichletlm` model and the `nostop` components), the tiles present a vast majority of dark blue squares.

By acting on the square plot on the right end side of the stemmer tiles, it is possible to select the third and fourth quartile in order to examine the top systems. By examining the remaining dark blue squares in the stemmer tiles, it is possible to see that the `porter`, `weakPorter`, `snowballPorter` and `krovetz` stemmers are well performing and quite close one to the other where `weakPorter` is slightly under-performing, whereas the `lovins` stemmer, while performing better than `nolug`, performs poorly.

6.3.1. Statistical Validation

Finding 1: the absence of the stemmer is detrimental for the performances of most of the systems.

This is validated by the `stemmer` main effects plot in Figure 12, where it is possible to see that the marginal mean for the `nolug` component is lower than all the other ones.

Finding 2: the interaction between the stemmer and the other components is low.

This is validated by the statistical analysis (the p-values) reported in the electronic appendix and in [25], where it is possible to see that the `stop list*stemmer`, `stemmer*model` and `stop list*stemmer*model` interactions are not statistically significant.

Finding 3: the `porter`, `weakPorter`, `snowballPorter` and `krovetz` stemmers are the top performing stemmers, `nolug` and `lovins` are the worst performing, and `krovetz` is the best performing stemmer.

In the Tukey HSD plot on the right part of Figure 16 it is possible to see that `nolug` is by any means the worst performing component followed by `lovins`. The `krovetz`, `porter`, `snowballPorter` stemmers are good performing stemmers, but they are in the same equivalence class so they do not present any significant statistical difference.

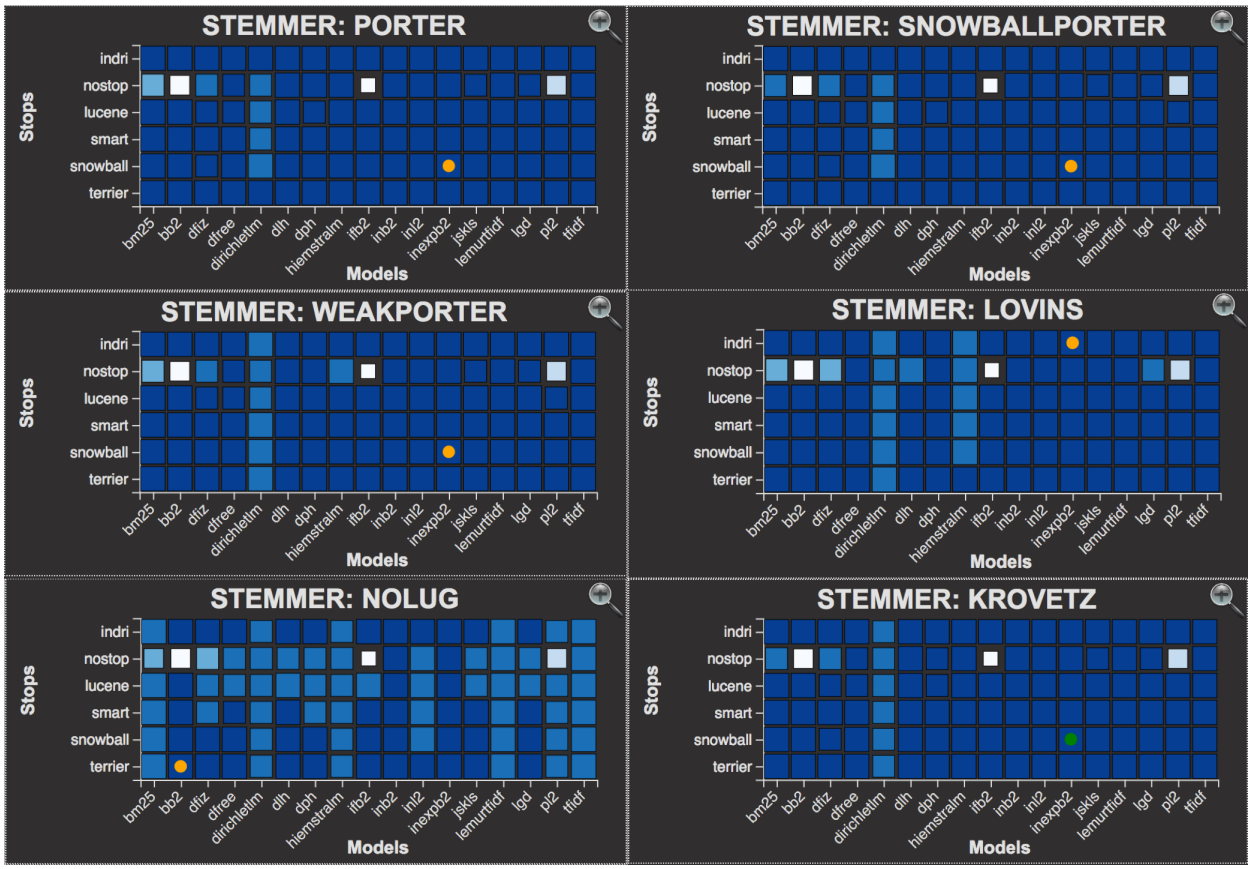


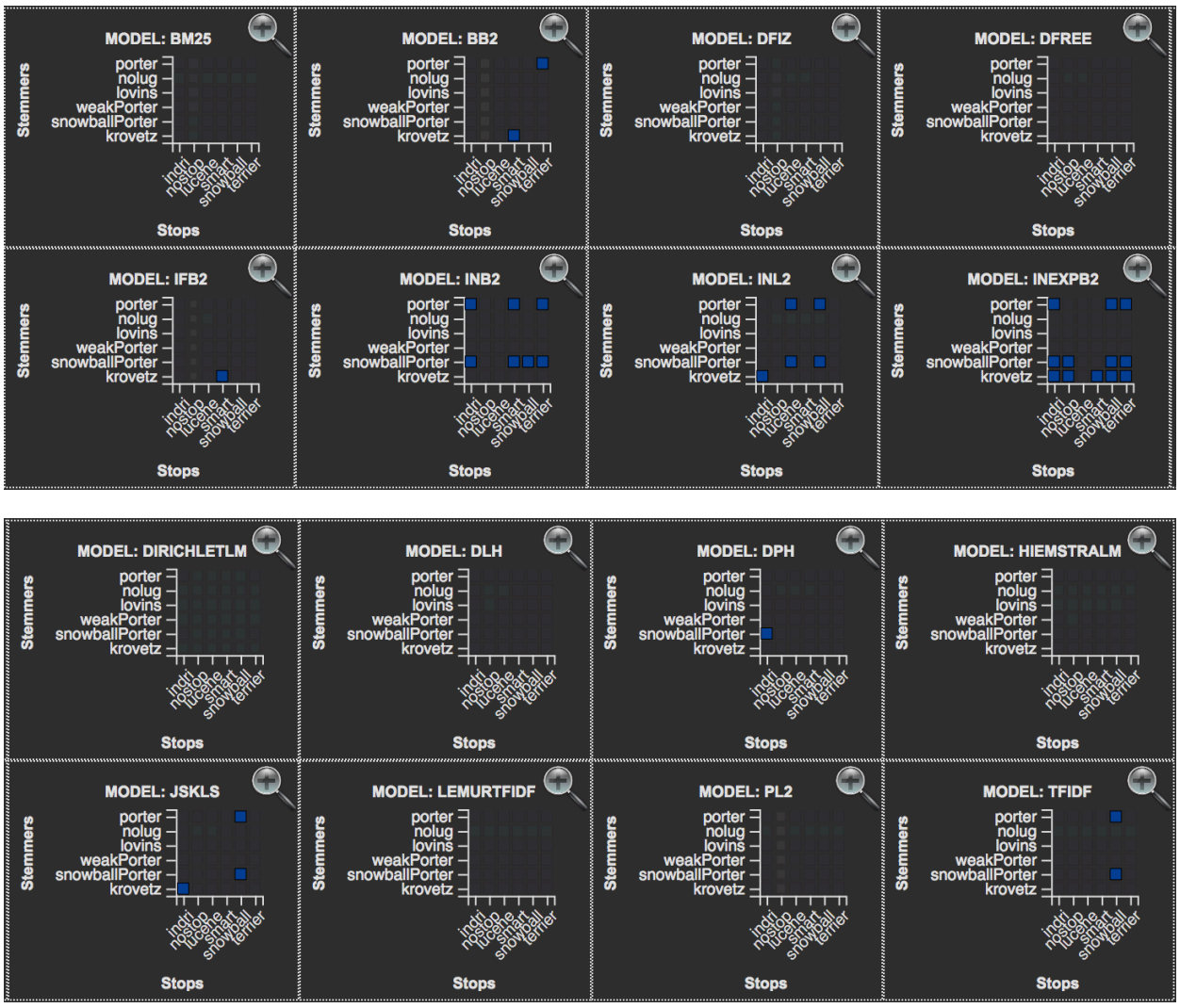
Figure 17: System configuration analysis area: we can see the stemmer tiles; IR models are on the x-axis and stop lists are on the y-axis.

6.4. Use case 4: The analytical engine

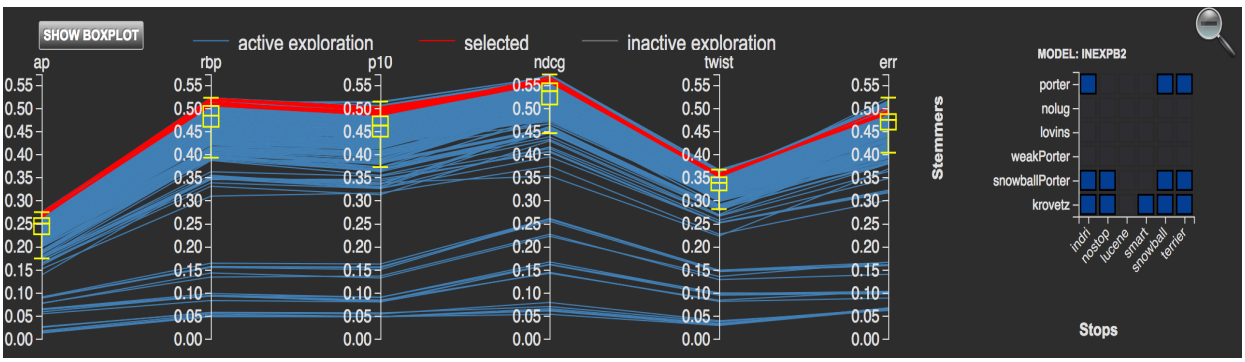
The analytical engine enables the selection of the "most performing" systems by weighting in all the available measures and it allows for filtering out models that are well performing on AP as we have seen above, but which are not consistently well performing across all the other measures. Figure 18 shows that the `inexpb2` model presents the highest number of highly performing systems (Figure 18 a) and that their performances are consistent across all the considered measures (Figure 18 b); moreover, it is possible to see that `inb2` and `inl2` are good models with performances across measures close to those of `inexpb2` (*finding 1*).

In particular, the analytical engine enables the selection of the "most promising" systems with a *bias towards performance*. These systems share a fixed high performance values range (fourth quartile) across measures and as little variance as possible across topics for all the considered measures. This automated analysis allows for understanding that for `T08` the most promising system is composed of the `jskls` model equipped with a `snowball` stop list and a `porter` stemmer (*finding 2*).

The very same analysis can be carried out very easily across collections, thus determining the most promising (with a bias towards performances) for all the available test collections and across all measures. We see that the most promising systems vary quite a lot between test collections; indeed, for `T07` the most promising combination of components is the `tfidf` model combined with the `terrier` stop list and the `porter` or `snowball porter` stemmers, for `T09` is the `lgd` model combined with the `indri` stop list and the `porter` or `snowball porter` stemmers, for `T10` is the `inexpb2` model combined with the `terrier` stop list and the `porter` or `snowball porter` stemmers, for `T14` is the `inl2` model combined with the `terrier` stop



(a)



(b)

Figure 18: (a) The model-oriented view of the tiles where the most performing systems are highlighted. (b) The parallel coordinates plot of the top performing systems using the `inexpb2` model.

list and the porter stemmer and for T15 are the bm25 model combined with the snowball stop list and the porter or snowball porter stemmers and the tfidf model combined with the snowball or smart stop lists and the snowball porter stemmer (*finding 3*).

6.4.1. Statistical Validation

Finding 1: inexpb2 is the most performing model across all measures and inb2 and inl2 are close to inexpb2.

The electronic appendix reports six Tukey HSD plots, one for each measure considered by CLAIRE analytical component, for T08 where the marginal means of the IR models are reported. It is possible to see that inexpb2 and inb2 are the only two models belonging to the top performing groups for all the measures and that inl2 belongs to the top group for the most impacting measures (i.e., AP, nDCG and P@10). This evidence confirms the CLAIRE findings.

Finding 2: the jskls model equipped with a snowball stop list and a porter stemmer is the most promising system for T08.

This can be partially validated by the three T08 Tukey HSD plots reported in the electronic appendix, where it is possible to see that the jskls model is amongst the first or the second top performing groups for all the measures, the snowball stop list is in the top performing group across all measures as well as porter for the stemmers. The statistical analysis does not provide information about third order interactions (stop list*stemmer*model), thus it is not possible to assess how these three components interact one with each other. It is possible to analyze second order interaction by referring to Figure 13 where it is possible to see that for the stemmer*model interaction plot (second row, third column), the jskls model and porter form the most performing pair of components; the stop list*model interaction plot (first row, third column) makes evdident that the jskls model and snowball stop list pair is amongst the top performing ones as well as the pair snowball stop list and porter stemmer (first row, second column plot). These results are consistent with the interaction plots for T08 adopting the other measures reported in the electronic appendix. Nevertheless, the statistical analysis does not allow for combining these evidences with the confidence intervals as well as to have a unique view of second order interactions across measures. Hence, CLAIRE provides a very deep intuition about third order interactions among components, about the behaviour of the systems (second and third order interactions) by considering more than one measure at a time and variance between topics across multiple measures, which otherwise could not be fully grasped by using any of the commonly available statistical tools.

Finding 3 could be partially validated in the same way as described above, but it requires to compare all the Tukey HSD plots for all the six considered test collections (i.e., 36 Tukey HSD plots) as well their 36 interaction plots (6 plots for 6 test collections). Nevertheless, this complex analysis would make evident only a hint (as discussed for the finding 2 above) about the most promising systems for each collection. Hence, CLAIRE proved to be highly effective and to overcome some of the limitations of the demanding statistical analyses that are usually adopted to make sense of the complex results of the experimental evaluation in IR.

7. Conclusions and future work

IR systems are the aggregation of several components that interact together to return the most relevant documents, within a given collection, to respond a user query. There is no viable method to estimate the performances of IR systems before implementing and testing them on several real-world scenarios. This process, though resource and time consuming, has been adopted since the 60s and proved to be an essential means to understand and improve IR systems. On the other hand, experimental evaluation allows for assessing the performances of IR systems as a whole and does not provide any insight about the performances and the interactions of single components. To this end, the common practice for large research laboratories and search engine companies is to experiment with all possible combinations of available components and then explore the very large resulting space of IR systems to individuate patterns and component interactions that may provide some insights about the internals of IR systems. The manual inspection of thousands of measures can be aided by the use of statistical analyses such as GLMM and multi-way ANOVA. These methods are resource demanding as well and require an extended knowledge to be interpreted.

We presented a relevant case implying the exploration of almost 1.5M data points – i.e., the GoP – corresponding to different performance measures of hundreds of IR systems. We detailed the characteristics of the GoP at hand, the process that lead to its creation and the statistical analyses we performed. The paper goal was to ease the GoP exploration and to make sense of this huge amount of data without a demanding and complex statistical analyses.

To this end, we developed a novel VA system, CLAIRE, that supports the analysis of a large set of IR systems. Distinguishing system features are its capability of presenting the user with both the solution space parameters and the associated measures and of shortening the statistical analysis providing a quick way for comparing different configurations and getting insights on the analyzed systems.

CLAIRE has been demonstrated against a comprehensive and representative set of open source components, collecting measures on six relevant and standard test collections widely used by the IR community both at the academic and industrial level. Statistical analyses have been conducted on such measures and used in the paper to validate the visual insights raising from the use of the system; this deep analysis led us to conclude that CLAIRE allows for visually discovering many insights that were determined with deep statistical analyses and also for getting additional insights not possible with traditional approaches so far.

As future work, we will extend the CLAIRE system by allowing users to upload their proprietary systems and components and compare them against the standard open-source baselines present in the CLAIRE GoP; something like this has been proposed also in Agosti et al. [1], Agosti and Ferro [2], Armstrong et al. [9], Di Nunzio and Ferro [18], Gollub et al. [29], Ioannakis et al. [36] even though in a traditional IR evaluation setting rather than for component-based evaluation. In this way, users will be able both to better break down the performance of their own systems and to understand whether their constituting components outperform standard open-source solutions. Moreover, we will extend the GoP in order to include other IR components such as parsers or query expansion modules. That will have a strong influence on the design of the visual system: adding just one additional dimension will produce a four dimensions GoP that requires a carefully design; possible solutions will rely on very simple 3D visualizations (occlusion problems strongly discourage this approach unless it is quite minimalist) or smart projection mechanisms, likely using the result of dimensionality reduction techniques as a steering mechanism to locate cluster of similar systems and project them in the actual CLAIRE fashion using the 3 most relevant components. Another extension of the paper in this direction is to study the effect of model parameters since they have a sizable impact on model performances; this will lead to an increased complexity and size of the GoP to be analysed and visualized.

References

[1] Agosti, M., Di Buccio, E., Ferro, N., Masiero, I., Peruzzo, S., Silvello, G., 2012. DIRECTions: Design and Specification of an IR Evaluation Infrastructure. In: Catarci, T., Forner, P., Hiemstra, D., Peñas, A., Santucci, G. (Eds.), Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics. Proceedings of the Third International Conference of the CLEF Initiative (CLEF 2012). Lecture Notes in Computer Science (LNCS) 7488, Springer, Heidelberg, Germany, pp. 88–99.

[2] Agosti, M., Ferro, N., 2009. Towards an Evaluation Infrastructure for DL Performance Evaluation. In: Tsakonas, G., Papatheodorou, C. (Eds.), Evaluation of Digital Libraries: An insight into useful applications and methods. Chandos Publishing, Oxford, UK, pp. 93–120.

[3] Amati, G., van Rijsbergen, C. J., 2002. Probabilistic Models of Information Retrieval based on measuring the Divergence From Randomness. ACM Transactions on Information Systems (TOIS) 20 (4), 357–389.

[4] Angelini, M., Ferro, N., Santucci, G., Silvello, G., August 2014. VIRTUE: A visual tool for information retrieval performance evaluation and failure analysis. Journal of Visual Languages & Computing (JVLC) 25 (4), 394–413.

[5] Angelini, M., Ferro, N., Santucci, G., Silvello, G., 2016. A Visual Analytics Approach for What-If Analysis of Information Retrieval Systems. In: [51], pp. 1081–1084.

[6] Angelini, M., Ferro, N., Santucci, G., Silvello, G., 2017. Visual Analytics for Information Retrieval Evaluation Campaigns. In: Sedlmair, M., Tominski, C. (Eds.), Proc. 8th International Workshop on Visual Analytics (EuroVA 2017). Eurographics Association, Goslar, Germany, pp. 25–30.

[7] Angelini, M., Santucci, G., october 2017. The dark side of progressive visual analytics. In: Ferreira, N., Nonato, L. G., Sadlo, F. (Eds.), Workshop on Visual Analytics, Information Visualization and Scientific Visualization (WVIS) in the 30th Conference on Graphics, Patterns and Images (SIBGRAPI'17). Niteri, RJ, Brazil.
URL `http://sibgrapi2017.ic.uff.br/`

[8] Arguello, J., Crane, M., Diaz, F., Lin, J., Trotman, A., December 2015. Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). SIGIR Forum 49 (2), 107–116.
[9] Armstrong, T. G., Moffat, A., Webber, W., Zobel, J., 2009. EvaluatIR: an Online Tool for Evaluating and Comparing IR Systems. In: Allan, J., Aslam, J. A., Sanderson, M., Zhai, C., Zobel, J. (Eds.), Proc. 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009). ACM Press, New York, USA, p. 833.
[10] Bachthaler, S., Weiskopf, D., 2008. Continuous scatterplots. IEEE Transactions on Visualization and Computer Graphics 14 (6), 1428–1435.
[11] Berger, W., Piringer, H., Filzmoser, P., Gröller, E., 2011. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. Computer Graphics Forum 30 (3), 911–920.
[12] Booshehrian, M., Möller, T., Peterman, R. M., Munzner, T., 2012. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. Computer Graphics Forum 31 (3 PART 3), 1235–1244.
[13] Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P., 2009. Expected Reciprocal Rank for Graded Relevance. In: Cheung, D. W.-L., Song, I.-Y., Chu, W. W., Hu, X., Lin, J. J. (Eds.), Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009). ACM Press, New York, USA, pp. 621–630.
[14] Cleverdon, C. W., 1997. The Cranfield Tests on Index Languages Devices. In: Spärck Jones, K., Willett, P. (Eds.), Readings in Information Retrieval. Morgan Kaufmann Publisher, Inc., San Francisco, CA, USA, pp. 47–60.
[15] Clinchant, S., Gaussier, E., 2010. Information-Based Models for Ad Hoc IR. In: Crestani, F., Marchand-Maillet, S., Efthimiadis, E. N., Savoy, J. (Eds.), Proc. 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010). ACM Press, New York, USA, pp. 234–241.
[16] Crestani, F., Vegas, J., de la Fuente, P., 2004. A Graphical User Interface for the Retrieval of Hierarchically Structured Documents. Information Processing & Management 40 (2), 269–289.
[17] Croft, W. B., Metzler, D., Strohman, T., 2009. Search Engines: Information Retrieval in Practice. Addison-Wesley, Reading (MA), USA.
[18] Di Nunzio, G. M., Ferro, N., 2005. DIRECT: a System for Evaluating Information Access Components of Digital Libraries. In: Rauber, A., Christodoulakis, S., Min Tjoa, A. (Eds.), Proc. 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005). Lecture Notes in Computer Science (LNCS) 3652, Springer, Heidelberg, Germany, pp. 483–484.
[19] Eskelinen, P., Miettinen, K., Klamroth, K., Hakanen, J., 2010. Pareto navigator for interactive nonlinear multiobjective optimization. OR Spectrum 32 (1), 211–227.
[20] Ferro, N., February 2017. Reproducibility Challenges in Information Retrieval Evaluation. ACM Journal of Data and Information Quality (JDIQ) 8 (2), 8:1–8:4.
[21] Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., Zobel, J., June 2016. Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". SIGIR Forum 50 (1), 68–82.
[22] Ferro, N., Harman, D., 2010. CLEF 2009: Grid@CLEF Pilot Track Overview. In: Peters, C., Di Nunzio, G. M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (Eds.), Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments – Tenth Workshop of the Cross–Language Evaluation Forum (CLEF 2009). Revised Selected Papers. Lecture Notes in Computer Science (LNCS) 6241, Springer, Heidelberg, Germany, pp. 552–565.
[23] Ferro, N., Sabetta, A., Santucci, G., Tino, G., 2011. Visual Comparison of Ranked Result Cumulated Gains. In: Miksch, S., Santucci, G. (Eds.), Proc. 2nd International Workshop on Visual Analytics (EuroVA 2011). Eurographics Association, Goslar, Germany, pp. 21–24.
[24] Ferro, N., Silvello, G., 2016. A General Linear Mixed Models Approach to Study System Component Effects. In: [51], pp. 25–34.
[25] Ferro, N., Silvello, G., 2018. Towards an Anatomy of IR System Component Performances. Journal of the American Society for Information Science and Technology (JASIST) 69 (2), 187–200.
[26] Ferro, N., Silvello, G., Keskustalo, H., Pirkola, A., Järvelin, K., 2016. The Twist Measure for IR Evaluation: Taking User's Effort Into Account. Journal of the American Society for Information Science and Technology (JASIST) 67 (3), 620–648.
[27] Fowler, R. H., Lawrence-Fowler, W. A., Wilson, B. A., 1991. Integrating Query, Thesaurus, and Documents Through a Common Visual Representation. In: Fox, E. A. (Ed.), Proc. 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1991). ACM Press, New York, USA.
[28] Fuhr, N., December 2012. Salton Award Lecture: Information Retrieval As Engineering Science. SIGIR Forum 46 (2), 19–28.
[29] Gollub, T., Stein, B., Burrows, S., Hoppe, D., 2012. TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. IEEE Computer Society, pp. 151–155.
[30] Hanbury, A., Müller, H., 2010. Automated Component-Level Evaluation: Present and Future. In: Agosti, M., Ferro, N., Peters, C., de Rijke, M., Smeaton, A. (Eds.), Multilingual and Multimodal Information Access Evaluation. Proceedings of the International Conference of the Cross-Language Evaluation Forum (CLEF 2010). Lecture Notes in Computer Science (LNCS) 6360, Springer, Heidelberg, Germany, pp. 124–135.
[31] Harman, D. K., 2011. Information Retrieval Evaluation. Morgan & Claypool Publishers, USA.
[32] Hearst, M. A., 2009. Search User Interfaces, 1st Edition. Cambridge University Press, New York, NY, USA.
[33] Hearst, M. A., November 2011. "Natural" Search User Interfaces. Communications of the ACM (CACM) 54 (11), 60–67.
[34] Hiemstra, D., 2000. Using Language Models for Information Retrieval. Ph.D. thesis, CTIT Ph.D. Thesis Series No. 01-32, Centre for Telematics and Information Technology, The Netherlands.
[35] Hochberg, Y., Tamhane, A. C., 1987. Multiple Comparison Procedures. John Wiley & Sons, USA.
[36] Ioannakis, G., A.Koutsoudis, Pratikakis, I., Chamzas, C., 2018. RETRIEVAL - An Online Performance Evaluation Tool

for Information Retrieval Methods. IEEE Trans. Multimedia 20 (1), 119–127.
[37] Järvelin, K., Kekäläinen, J., October 2002. Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems (TOIS) 20 (4), 422–446.
[38] Koshman, S., 2005. Testing User Interaction with a Prototype Visualization-Based Information Retrieval System. Journal of the American Society for Information Science and Technology (JASIST) 56 (8), 824–833.
URL http://dx.doi.org/10.1002/asi.20175
[39] Krovetz, R., April 2000. Viewing morphology as an inference process. Artificial Intelligence 118 (1–2), 277–294.
[40] Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C., Vigna, S., 2016. Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge. In: Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., Silvello, G. (Eds.), Advances in Information Retrieval. Proc. 38th European Conference on IR Research (ECIR 2016). Lecture Notes in Computer Science (LNCS) 9626, Springer, Heidelberg, Germany, pp. 357–368.
[41] Lipani, A., Lupu, M., Hanbury, A., 2017. Visual Pool: A Tool to Visualize and Interact with the Pooling Method. In: Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A. P., White, R. W. (Eds.), Proc. 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). ACM Press, New York, USA.
[42] Liu, R., Chakrabarti, A., Samanta, T., Ghosh, J. K., Ghosh, M., 06 2014. On Divergence Measures Leading to Jeffreys and Other Reference Priors. Bayesian Anal. 9 (2), 331–370.
URL https://doi.org/10.1214/14-BA862
[43] Lovins, J. B., January/February 1971. Error Evaluation for Stemming Algorithms as Clustering Algorithms. Journal of the American Society for Information Science (JASIS) 22 (1), 28–40.
[44] Mansmann, S., Scholl, M. H., 2008. Visual olap: A new paradigm for exploring multidimensional aggregates. In: MCCSIS'08 - IADIS Multi Conference on Computer Science and Information Systems; Proceedings of Computer Graphics and Visualization 2008 and Gaming 2008: Design for Engaging Experience Soc. Interaction. pp. 59–66.
[45] Matkovic, K., Gracanin, D., Jelovic, M., Hauser, H., 2008. Interactive visual steering - rapid visual prototyping of a common rail injection system. IEEE Transactions on Visualization and Computer Graphics 14 (6), 1699–1706.
[46] Maxwell, S., Delaney, H. D., 2004. Designing Experiments and Analyzing Data. A Model Comparison Perspective, 2nd Edition. Lawrence Erlbaum Associates, Mahwah (NJ), USA.
[47] Moffat, A., Zobel, J., 2008. Rank-biased Precision for Measurement of Retrieval Effectiveness. ACM Transactions on Information Systems (TOIS) 27 (1), 2:1–2:27.
[48] Morse, E. L., Lewis, M., Olsen, K. A., 2002. Testing Visual Information Retrieval Methodologies Case Study: Comparative Analysis of Textual, Icon, Graphical, and Spring Displays. Journal of the American Society for Information Science and Technology (JASIST) 53 (1), 28–40.
[49] Olejnik, S., Algina, J., December 2003. Generalized Eta and Omega Squared Statistics: Measures of Effect Size for Some Common Research Designs. Psychological Methods 8 (4), 434–447.
[50] Padua, L., Schulze, H., Matkovi, K., Delrieux, C., 2014. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. Computers and Graphics (Pergamon) 41 (1), 99–113.
[51] Perego, R., Sebastiani, F., Aslam, J., Ruthven, I., Zobel, J. (Eds.), 2016. Proc. 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016). ACM Press, New York, USA.
[52] Piringer, H., Berger, W., Krasser, J., 2010. Hypermoval: Interactive visual validation of regression models for real-time simulation. Computer Graphics Forum 29 (3), 983–992.
[53] Porter, M. F., July 1980. An algorithm for suffix stripping. Program 14 (3), 130–137.
[54] Roberts, J. C., 2007. State of the art: Coordinated & multiple views in exploratory visualization. In: Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on. IEEE, pp. 61–71.
[55] Robertson, S. E., 1981. The methodology of information retrieval experiment. In: Spärck Jones, K. (Ed.), Information Retrieval Experiment. Butterworths, London, United Kingdom, pp. 9–31.
[56] Robertson, S. E., Zaragoza, U., 2009. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval (FnTIR) 3 (4), 333–389.
[57] Roelleke, T., 2013. Information Retrieval Models. Foundations and Relationships. Morgan & Claypool Publishers, USA.
[58] Rutherford, A., 2011. ANOVA and ANCOVA. A GLM Approach, 2nd Edition. John Wiley & Sons, New York, USA.
[59] Salton, G., McGill, M. J., 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York, USA.
[60] Sanderson, M., 2010. Test Collection Based Evaluation of Information Retrieval Systems. Foundations and Trends in Information Retrieval (FnTIR) 4 (4), 247–375.
[61] Schulz, H.-J., Angelini, M., Santucci, G., Schumann, H., 2016. An enhanced visualization process model for incremental visualization. IEEE transactions on visualization and computer graphics 22 (7), 1830–1842.
[62] Shaffer, C. A., Knill, D. L., Watson, L. T., 1998. Visualization for multiparameter aircraft designs. In: Proceedings of the IEEE Visualization Conference. pp. 491–494.
[63] Shireen, N., 2016. Paraxplore interfaces: Parametric interfaces for parallel exploration in design. In: Companion Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces: Nature Meets Interactive Surfaces, ISS 2016. pp. 7–12.
URL www.scopus.com
[64] Su, H., Nelder, J. A., Wolbert, P., Spence, R., 1996. Application of generalized linear models to the design improvement of an engineering artefact. Quality and Reliability Engineering International 12 (2), 101–112.
URL http://dx.doi.org/10.1002/(SICI)1099-1638(199603)12:2<101::AID-QRE991>3.0.CO;2-9
[65] Trotman, A., Clarke, C. L. A., Ounis, I., Culpepper, J. S., Cartright, M.-A., Geva, S., December 2012. Open Source

Information Retrieval: a Report on the SIGIR 2012 Workshop. ACM SIGIR Forum 46 (2), 95–101.
[66] Trotman, A., Puurula, A., Burgess, B., 2014. Improvements to BM25 and Language Models Examined. In: Culpepper, J. S., Park, L., Zuccon, G. (Eds.), Proc. 19th Australasian Document Computing Symposium (ADCS 2014). ACM Press, New York, USA, pp. 58–65.
[67] Tufte, E. R., 2001. The visual display of quantitative information.
[68] Tweedie, L., Spence, B., Dawkes, H., Su, H., 1995. The influence explorer. In: Conference Companion on Human Factors in Computing Systems. CHI '95. ACM, New York, NY, USA, pp. 129–130.
URL http://doi.acm.org/10.1145/223355.223464
[69] Tweedie, L., Spence, R., Dawkes, H., Su, H., 1996. Externalising abstract mathematical models. In: Conference on Human Factors in Computing Systems - Proceedings. pp. 406–412.
[70] Zhai, C., 2008. Statistical Language Models for Information Retrieval. A Critical Review. Foundations and Trends in Information Retrieval (FnTIR) 2 (3), 137–213.
[71] Zhang, J., 2001. TOFIR: A Tool of Facilitating Information Retrieval - Introduce a Visual Retrieval Model. Information Processing & Management 37 (4), 639–657.
[72] Zhang, J., 2008. Visualization for Information Retrieval. Springer-Verlag, Heidelberg, Germany.

Electronic appendix

In this appendix we report the detailed statistical analyses conducted on the *Grid of Points (GoP)* defined on the six test collections considered in the paper: TREC 07 Adhoc, TREC 08 Adhoc, TREC 09 Web, TREC 10 Web, TREC 14 Terabyte and TREC 15 Terabyte. This appendix is meant to be used as a statistical counterpart of the validation use cases presented in the paper.

7.1. GLMM and effect size

We use a three-way GLMM to carry out the analysis of IR GoP. In this three factors design we manipulate factors A, B and C corresponding to the stop lists, the *Lexical Unit Generator (LUG)* and the IR models respectively; with this design we can also study the interaction between component pairs as well as the third order interaction between them even though we cannot visualize them by means of main and interaction effect plots.

The systems are decomposed into the three main constituents delined in the paper: (i) factor A (stop lists) with p levels where, for instance, A_1 corresponds to the absence of a stop list, A_2 to the indri stop list, A_3 to the terrier stop list and so on; (ii) factor B (stemmer) with q levels where B_1 corresponds to the absence of a stemmer, B_2 to the Porter stemmer, B_3 to the Krovetz stemmer and so on; (iii) factor C (IR models) with r levels where C_1 corresponds to BM25, C_2 to TF*IDF and so on.

The full GLMM for the described factorial ANOVA design for repeated measures with three fixed factors (A, B, C) and a random factor ($T^{'}$) is:

$$\begin{aligned} Y_{ijkl} = & \underbrace{\mu_{....} + \tau_i + \alpha_j + \beta_k + \gamma_l}_{\text{Main Effects}} + \\ & \underbrace{\alpha\beta_{jk} + \alpha\gamma_{jl} + \beta\gamma_{kl} + \alpha\beta\gamma_{jkl}}_{\text{Interaction Effects}} + \underbrace{\varepsilon_{ijkl}}_{\text{Error}} \end{aligned} \tag{4}$$

where: Y_{ijkl} is the score of the i-th subject in the j-th, k-th, and l-th factors; $\mu_{....}$ is the grand mean; τ_i is the effect of the i-th subject $\tau_i = \mu_{i...} - \mu_{....}$ where $\mu_{i...}$ is the mean of the i-th subject; $\alpha_j = \mu_{.j..} - \mu_{....}$ is the effect of the j-th factor, where $\mu_{.j..}$ is the mean of the j-th factor; $\beta_k = \mu_{..k.} - \mu_{....}$ is the effect of the k-th factor, where $\mu_{..k.}$ is the mean of the k-th factor; and, $\gamma_l = \mu_{...l} - \mu_{....}$ is the effect of the l-th factor where $\mu_{...l}$ is the mean of the l-th factor; ε_{ijkl} is the error committed by the model in predicting the score of the i-th subject in the three factors j, k, l. It consists of all the interaction terms between the random subjects and the fixed factors, such as $(\tau\alpha)_{ij}$, $(\tau\beta)_{ik}$ and so on, plus the error ϵ_{ijkl} which is any additional error due to uncontrolled sources of variance. As in the single factor design to calculate interaction effects with the subjects, you need to have replicates; when there is only one score per subject per factor the factor ϵ_{ijkl} cannot be separated from the interaction effects with the random subjects.

The estimators of the main effects can be derived by extension from those of the single factor design; for instance, the grand mean is $\hat{\mu}_{....} = \frac{1}{rqpn}\sum_{l=1}^{r}\sum_{k=1}^{q}\sum_{j=1}^{p}\sum_{i=1}^{n} Y_{ijkl}$, the mean of the k-th effect is $\hat{\mu}_{..k.} = \frac{1}{rpn}\sum_{l=1}^{r}\sum_{j=1}^{p}\sum_{i=1}^{n} Y_{ijkl}$ and its estimator is $\hat{\beta}_k = \hat{\mu}_{..k.} - \hat{\mu}_{....}$.

The estimators of the interaction factors are calculated as follows, let us consider $(\alpha\beta)_{jk}$:

$$\widehat{\alpha\beta}_{jk} = \hat{\mu}_{.jk.} - (\hat{\mu}_{....} + \hat{\alpha}_j + \hat{\beta}_k) \tag{5}$$

where $\hat{\mu}_{.jk.} = \frac{1}{nr}\sum_{i=1}^{n}\sum_{l=1}^{r} Y_{ijkl}$; $\hat{\alpha}_j = \hat{\mu}_{.j..} - \hat{\mu}_{....}$; and, $\hat{\beta}_k = \hat{\mu}_{..k.} - \hat{\mu}_{....}$.

Similarly, we calculate the estimators for all the other interaction factors – i.e. $\widehat{\alpha\gamma}_{jl}$ and $\widehat{\beta\gamma}_{kl}$; $\widehat{\alpha\beta\gamma}_{jkl}$ is calculated by extending equation (5):

$$\widehat{\alpha\beta\gamma}_{jkl} = \hat{\mu}_{.jkl} - (\hat{\mu}_{....} + \hat{\alpha}_j + \hat{\beta}_k + \hat{\gamma}_l) \tag{6}$$

where $\hat{\mu}_{.jkl} = \frac{1}{n}\sum_{i=1}^{n} Y_{ijkl}$ and $\hat{\gamma}_l = \hat{\mu}_{...l} - \hat{\mu}_{....}$.

In this design the error $\varepsilon_{ijkl} = Y_{ijkl} - \hat{Y}_{ijkl}$ contains the variance not explained by the main and interaction effects discussed above and it is composed by all the interactions of the subjects τ_j with the other factors in the model, in addition to the uncontrolled sources of variance.

We are not only interested in determining whether a factor effect is significant, but also which proportion of the variance is due to it, that is we need to estimate its *effect-size measure* or *Strength of Association (SOA)*. The SOA is a "standardized index and estimates a parameter that is independent of sample size and quantifies the magnitude of the difference between populations or the relationship between explanatory and response variables" [49].

$$\hat{\omega}^2_{\langle fact \rangle} = \frac{df_{fact}(F_{fact} - 1)}{df_{fact}(F_{fact} - 1) + N} \tag{7}$$

is an unbiased estimator of the variance components associated with the sources of variation in the design, where F_{fact} is the F-statistics and df_{fact} are the degrees of freedom for the factor while N is the total number of samples.

The common rule of thumb [58] when classifying $\hat{\omega}^2_{\langle fact \rangle}$ effect size is: 0.14 and above is a *large size effect*, 0.06–0.14 is a *medium size effect*, and 0.01–0.06 is a *small size effect*. $\hat{\omega}^2_{\langle fact \rangle}$ values could happen to be negative and in such cases they are considered as zero.

A *Type I* error occurs when a true null hypothesis is rejected and the significance level α is the probability of committing a Type I error. When performing multiple comparisons, the probability of committing a Type I error increases with the number of comparisons and we keep it controlled by applying the Tukey HSD test [35] with a significance level $\alpha = 0.05$. Tukey's method is used in ANOVA to create confidence intervals for all pairwise differences between factor levels, while controlling the family error rate; it is an effective method generally more powerful than other popular statistical methods like the Bonferroni one [46]. Two levels u and v of a factor are considered significantly different when

$$|t| = \frac{|\hat{\mu}_u - \hat{\mu}_v|}{\sqrt{MS_{error}\left(\frac{1}{n_u} + \frac{1}{n_v}\right)}} > \frac{1}{\sqrt{2}} q_{\alpha,k,N-k} \tag{8}$$

where $\hat{\mu}_u$ and $\hat{\mu}_v$ are the marginal means, i.e. the main effects, of the two factors; n_u and n_v are the sizes of levels u and v; $q_{\alpha,k,N-k}$ is the upper $100 * (1 - \alpha)$th percentile of the studentized range distribution with parameter k and $N - k$ degrees of freedom; k is the number of levels in the factor and N is the total number of observations.

A *Type 2* error occurs when a false null hypothesis is accepted and it is concerned with the capability of the conducted experiment to actually detect the effect under examination. Type 2 errors are often overlooked because if they occur, although a real effect is missed, no misdirection occurs and further experimentation is very likely to reveal the effect.

7.2. Detailed results of the statistical analyses

We present a table for each collection reporting the estimated ω^2 SOA for all the main and interaction effects and, within parentheses, the p-values for all the ANOVA three-way tests we conducted (see Table 1–6); the color coding of the cells is the following: not significant effects are in light grey; small size effects are in light blue; medium size effects are in blue; and large size effects are in dark blue.

Moreover, we report the main effect plots (see Figure 19 for the main effect plots based on AP for the different test collections and Figure 20 for the main effect plots of different measures over TREC 08 test collection) and the interaction plots (see Figures 21, 22, 23, 24, 25 and 26) for all the considered test collections based on AP. Moreover in Figures 27, 28 and 29 we report the Tukey plot of the marginal means for the models, stemmers and stop lists over TREC 08.

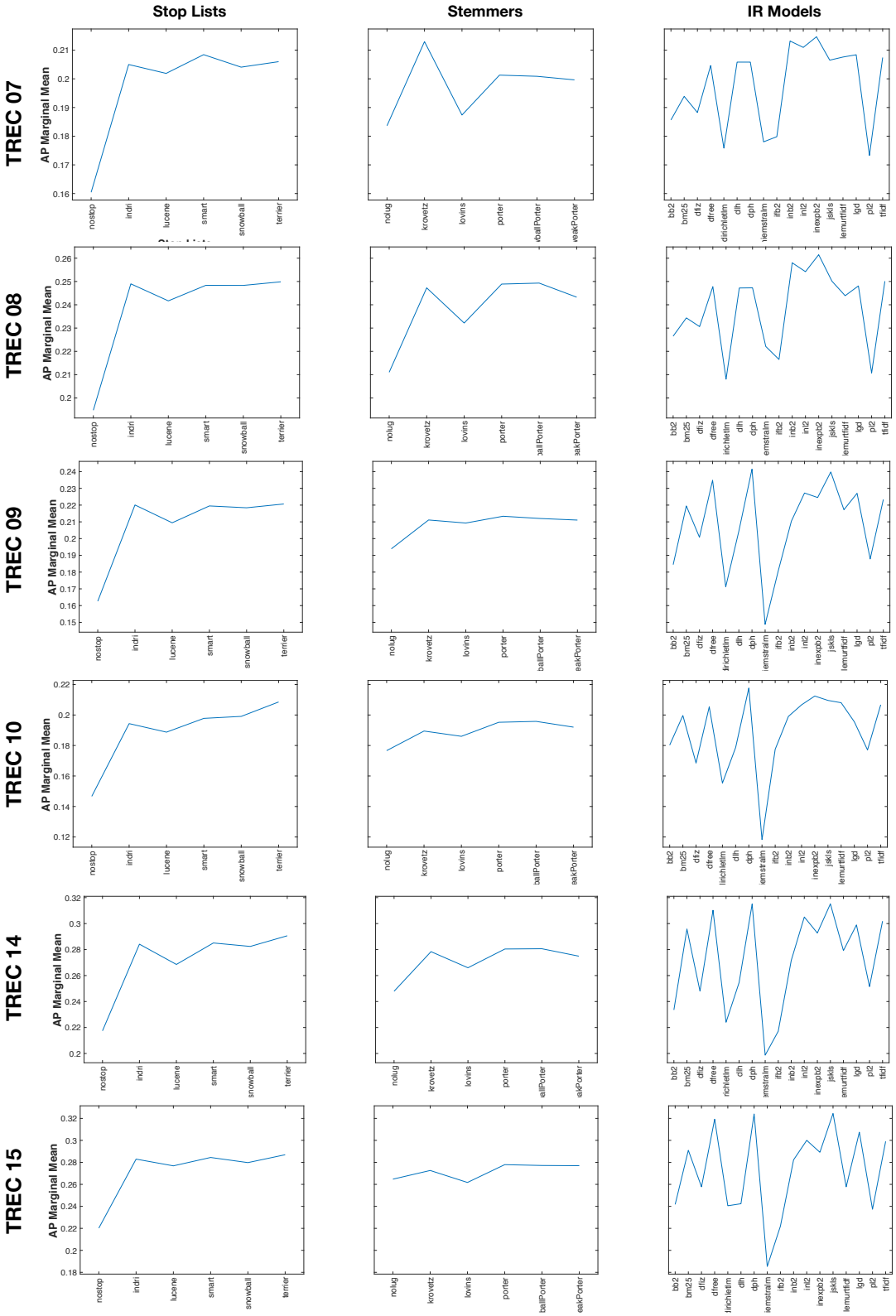


Figure 19: Main effects for the AP measure on TREC 07, TREC 08, TREC 09, TREC 10, TREC 14, TREC 15 test collections.

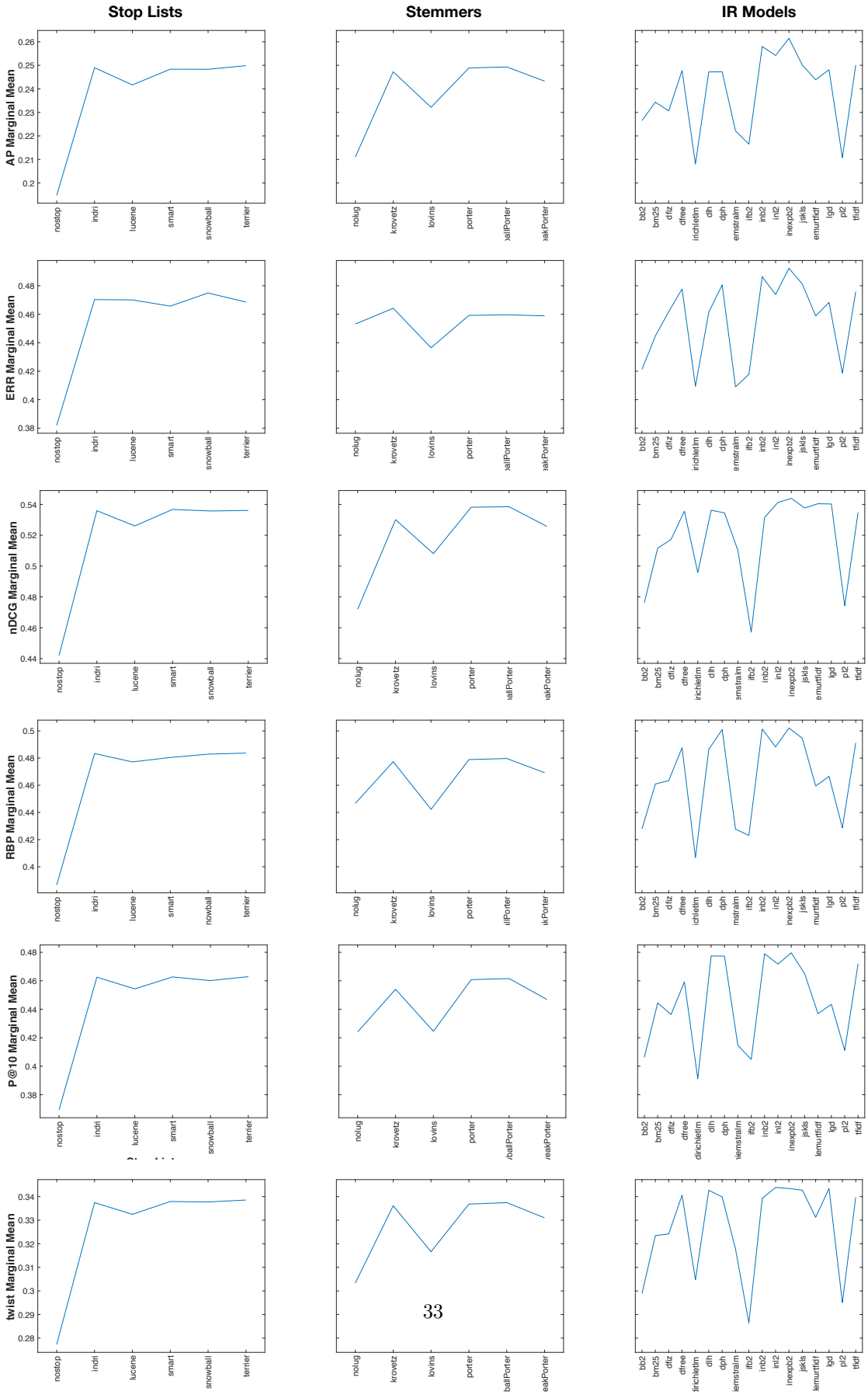


Figure 20: Main effects for the all the considered measures (AP, ERR, nDCG, RBP, P@10, twist) on TREC 08, 1999, Adhoc test collection; we can see that the `dirichletlm` model is amongst the worst performing models for all the measures.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\omega^2_{\langle \text{Stop Lists} \rangle}$	0.0564 (0.00)	0.0611 (0.00)	0.0740 (0.00)	0.0704 (0.00)	0.0750 (0.00)	0.1502 (0.00)	0.0564 (0.00)	0.0569 (0.00)	0.1010 (0.00)
$\omega^2_{\langle \text{Stemmers} \rangle}$	0.0195 (0.00)	0.0126 (0.00)	0.0164 (0.00)	0.0123 (0.00)	0.0120 (0.00)	0.0432 (0.00)	0.0036 (0.00)	0.0036 (0.00)	0.0306 (0.00)
$\omega^2_{\langle \text{IR Models} \rangle}$	0.0393 (0.00)	0.0418 (0.00)	0.0509 (0.00)	0.0475 (0.00)	0.0505 (0.00)	0.0877 (0.00)	0.0426 (0.00)	0.0428 (0.00)	0.0638 (0.00)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0002 (0.79)	0.0001 (0.30)	-0.0001 (0.71)	-0.0002 (0.75)	0.0001 (0.31)	-0.0000 (0.55)	-0.0002 (0.86)	-0.0002 (0.85)	-0.0003 (0.88)
$\omega^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.0987 (0.00)	0.1316 (0.00)	0.1512 (0.00)	0.1479 (0.00)	0.1579 (0.00)	0.2897 (0.00)	0.1280 (0.00)	0.1299 (0.00)	0.2161 (0.00)
$\omega^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0012 (1.00)	0.0007 (0.04)	-0.0012 (1.00)	0.0001 (0.38)	-0.0010 (1.00)	-0.0017 (1.00)	0.0018 (0.00)	0.0018 (0.00)	-0.0004 (0.83)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0127 (1.00)	-0.0117 (1.00)	-0.0124 (1.00)	-0.0126 (1.00)	-0.0123 (1.00)	-0.0123 (1.00)	-0.0120 (1.00)	-0.0120 (1.00)	-0.0120 (1.00)

Table 1: Topic/Component Effects on TREC 07, 1998, Adhoc. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

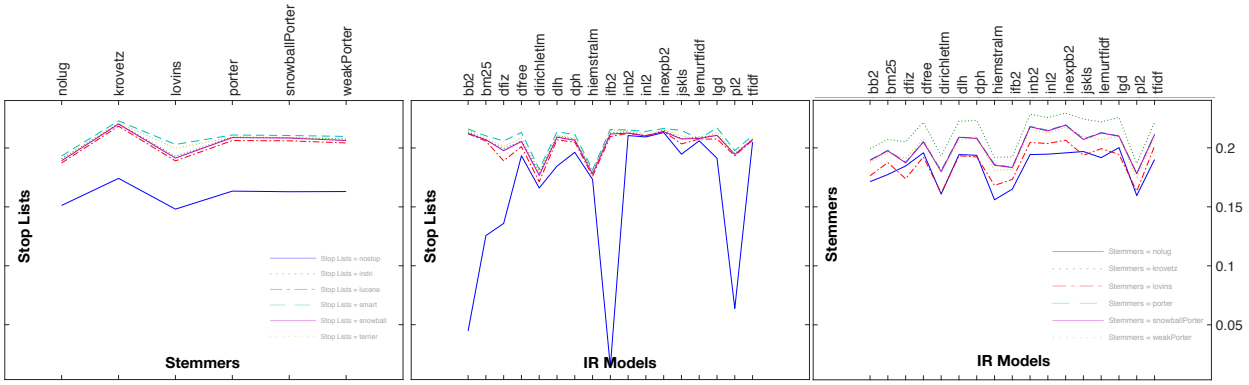


Figure 21: Interaction effects for the AP measure on TREC 07, 1998, Adhoc test collection.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\omega^2_{\langle \text{Stop Lists} \rangle}$	0.0796 (0.00)	0.0744 (0.00)	0.0899 (0.00)	0.0835 (0.00)	0.0786 (0.00)	0.1340 (0.00)	0.0603 (0.00)	0.0610 (0.00)	0.1063 (0.00)
$\omega^2_{\langle \text{Stemmers} \rangle}$	0.0394 (0.00)	0.0165 (0.00)	0.0344 (0.00)	0.0168 (0.00)	0.0255 (0.00)	0.0656 (0.00)	0.0047 (0.00)	0.0047 (0.00)	0.0372 (0.00)
$\omega^2_{\langle \text{IR Models} \rangle}$	0.0542 (0.00)	0.0561 (0.00)	0.0695 (0.00)	0.0648 (0.00)	0.0711 (0.00)	0.0832 (0.00)	0.0452 (0.00)	0.0449 (0.00)	0.0784 (0.00)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0004 (0.98)	-0.0005 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0005 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0005 (1.00)
$\omega^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.1337 (0.00)	0.1544 (0.00)	0.1728 (0.00)	0.1749 (0.00)	0.1722 (0.00)	0.2606 (0.00)	0.1362 (0.00)	0.1379 (0.00)	0.2165 (0.00)
$\omega^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0016 (1.00)	-0.0004 (0.86)	-0.0012 (1.00)	-0.0012 (1.00)	-0.0011 (1.00)	-0.0015 (1.00)	-0.0013 (1.00)	-0.0013 (1.00)	-0.0012 (1.00)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0124 (1.00)	-0.0116 (1.00)	-0.0122 (1.00)	-0.0124 (1.00)	-0.0122 (1.00)	-0.0122 (1.00)	-0.0117 (1.00)	-0.0117 (1.00)	-0.0120 (1.00)

Table 2: Topic/Component Effects on TREC 08, 1999, Adhoc. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\omega^2_{\langle \text{Stop Lists} \rangle}$	0.0603 (0.00)	0.0448 (0.00)	0.0443 (0.00)	0.0551 (0.00)	0.0795 (0.00)	0.1907 (0.00)	0.0334 (0.00)	0.0338 (0.00)	0.0723 (0.00)
$\omega^2_{\langle \text{Stemmers} \rangle}$	0.0063 (0.00)	0.0013 (0.00)	0.0016 (0.00)	0.0014 (0.00)	0.0016 (0.00)	0.0122 (0.00)	0.0007 (0.00)	0.0008 (0.00)	0.0027 (0.00)
$\omega^2_{\langle \text{IR Models} \rangle}$	0.0865 (0.00)	0.0738 (0.00)	0.0691 (0.00)	0.0938 (0.00)	0.0804 (0.00)	0.1847 (0.00)	0.0734 (0.00)	0.0728 (0.00)	0.0688 (0.00)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0005 (0.99)	-0.0003 (0.96)	-0.0005 (0.99)	-0.0007 (1.00)	-0.0006 (1.00)	-0.0004 (0.98)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0005 (0.99)
$\omega^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.0957 (0.00)	0.1099 (0.00)	0.0953 (0.00)	0.1306 (0.00)	0.1409 (0.00)	0.3350 (0.00)	0.0861 (0.00)	0.0879 (0.00)	0.1371 (0.00)
$\omega^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0015 (1.00)	-0.0015 (1.00)	-0.0009 (0.99)	-0.0015 (1.00)	-0.0006 (0.92)	-0.0011 (1.00)	-0.0009 (0.99)	-0.0008 (0.99)	-0.0009 (0.99)
$\omega^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0125 (1.00)	-0.0118 (1.00)	-0.0116 (1.00)	-0.0125 (1.00)	-0.0121 (1.00)	-0.0127 (1.00)	-0.0117 (1.00)	-0.0117 (1.00)	-0.0113 (1.00)

Table 3: Topic/Component Effects on TREC 09, 2000, Web. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\hat{\omega}^2_{\langle \text{Stop Lists} \rangle}$	0.0643 (0.00)	0.0505 (0.00)	0.0511 (0.00)	0.0569 (0.00)	0.0624 (0.00)	0.1391 (0.00)	0.0295 (0.00)	0.0300 (0.00)	0.0742 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \rangle}$	0.0072 (0.00)	0.0025 (0.00)	0.0053 (0.00)	0.0018 (0.00)	0.0044 (0.00)	0.0259 (0.00)	0.0027 (0.00)	0.0027 (0.00)	0.0063 (0.00)
$\hat{\omega}^2_{\langle \text{IR Models} \rangle}$	0.0948 (0.00)	0.0777 (0.00)	0.0576 (0.00)	0.1001 (0.00)	0.0877 (0.00)	0.1314 (0.00)	0.0674 (0.00)	0.0669 (0.00)	0.0801 (0.00)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0007 (1.00)	-0.0004 (0.99)	-0.0006 (1.00)	-0.0005 (1.00)	-0.0005 (1.00)	-0.0004 (0.96)	-0.0001 (0.63)	-0.0001 (0.58)	-0.0004 (0.99)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.0862 (0.00)	0.1027 (0.00)	0.0839 (0.00)	0.1160 (0.00)	0.1344 (0.00)	0.2607 (0.00)	0.0863 (0.00)	0.0880 (0.00)	0.1444 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0006 (0.95)	-0.0006 (0.94)	-0.0001 (0.62)	-0.0011 (1.00)	-0.0014 (1.00)	-0.0008 (0.98)	-0.0013 (1.00)	-0.0013 (1.00)	-0.0000 (0.49)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0126 (1.00)	-0.0117 (1.00)	-0.0116 (1.00)	-0.0125 (1.00)	-0.0121 (1.00)	-0.0127 (1.00)	-0.0114 (1.00)	-0.0114 (1.00)	-0.0116 (1.00)

Table 4: Topic/Component Effects on TREC 10, 2001, Web. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\hat{\omega}^2_{\langle \text{Stop Lists} \rangle}$	0.0951 (0.00)	0.0568 (0.00)	0.1027 (0.00)	0.0657 (0.00)	0.0749 (0.00)	0.1476 (0.00)	0.0327 (0.00)	0.0329 (0.00)	0.1111 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \rangle}$	0.0220 (0.00)	0.0045 (0.00)	0.0194 (0.00)	0.0048 (0.00)	0.0039 (0.00)	0.0224 (0.00)	0.0019 (0.00)	0.0018 (0.00)	0.0179 (0.00)
$\hat{\omega}^2_{\langle \text{IR Models} \rangle}$	0.1810 (0.00)	0.1261 (0.00)	0.1721 (0.00)	0.1429 (0.00)	0.1604 (0.00)	0.2030 (0.00)	0.1001 (0.00)	0.0994 (0.00)	0.1666 (0.00)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0001 (0.66)	-0.0004 (0.98)	-0.0002 (0.84)	-0.0004 (0.97)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0005 (1.00)	-0.0005 (1.00)	-0.0002 (0.82)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.1408 (0.00)	0.1322 (0.00)	0.1946 (0.00)	0.1486 (0.00)	0.1713 (0.00)	0.2647 (0.00)	0.1051 (0.00)	0.1067 (0.00)	0.2000 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	0.0017 (0.00)	0.0004 (0.19)	0.0014 (0.00)	0.0004 (0.18)	0.0004 (0.17)	0.0008 (0.04)	0.0009 (0.03)	0.0009 (0.02)	0.0012 (0.01)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0127 (1.00)	-0.0115 (1.00)	-0.0124 (1.00)	-0.0121 (1.00)	-0.0119 (1.00)	-0.0127 (1.00)	-0.0109 (1.00)	-0.0109 (1.00)	-0.0114 (1.00)

Table 5: Topic/Component Effects on TREC 14, 2004, Terabyte. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

Effects	ap	p10	rprec	rbp	ndcg20	ndcg	err20	err	twist
$\hat{\omega}^2_{\langle \text{Stop Lists} \rangle}$	0.0804 (0.00)	0.0454 (0.00)	0.0933 (0.00)	0.0462 (0.00)	0.0594 (0.00)	0.1701 (0.00)	0.0326 (0.00)	0.0329 (0.00)	0.0940 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \rangle}$	0.0064 (0.00)	0.0022 (0.00)	0.0045 (0.00)	0.0035 (0.00)	0.0026 (0.00)	0.0172 (0.00)	0.0038 (0.00)	0.0038 (0.00)	0.0057 (0.00)
$\hat{\omega}^2_{\langle \text{IR Models} \rangle}$	0.1949 (0.00)	0.1165 (0.00)	0.1686 (0.00)	0.1126 (0.00)	0.1432 (0.00)	0.2233 (0.00)	0.0876 (0.00)	0.0866 (0.00)	0.1726 (0.00)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \rangle}$	-0.0006 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0006 (1.00)	-0.0005 (1.00)	-0.0005 (1.00)	-0.0005 (1.00)	-0.0007 (1.00)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{IR Models} \rangle}$	0.1552 (0.00)	0.1362 (0.00)	0.1960 (0.00)	0.1407 (0.00)	0.1653 (0.00)	0.3512 (0.00)	0.0943 (0.00)	0.0963 (0.00)	0.2294 (0.00)
$\hat{\omega}^2_{\langle \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0001 (0.58)	-0.0008 (0.99)	-0.0003 (0.77)	-0.0011 (1.00)	-0.0003 (0.74)	-0.0005 (0.89)	-0.0008 (0.99)	-0.0008 (0.99)	0.0007 (0.05)
$\hat{\omega}^2_{\langle \text{Stop Lists} \times \text{Stemmers} \times \text{IR Models} \rangle}$	-0.0128 (1.00)	-0.0121 (1.00)	-0.0126 (1.00)	-0.0126 (1.00)	-0.0125 (1.00)	-0.0126 (1.00)	-0.0119 (1.00)	-0.0119 (1.00)	-0.0118 (1.00)

Table 6: Topic/Component Effects on TREC 15, 2005, Terabyte. Each cell reports the ω^2 SoA for the speficied effects and, within parentheses, the p-value.

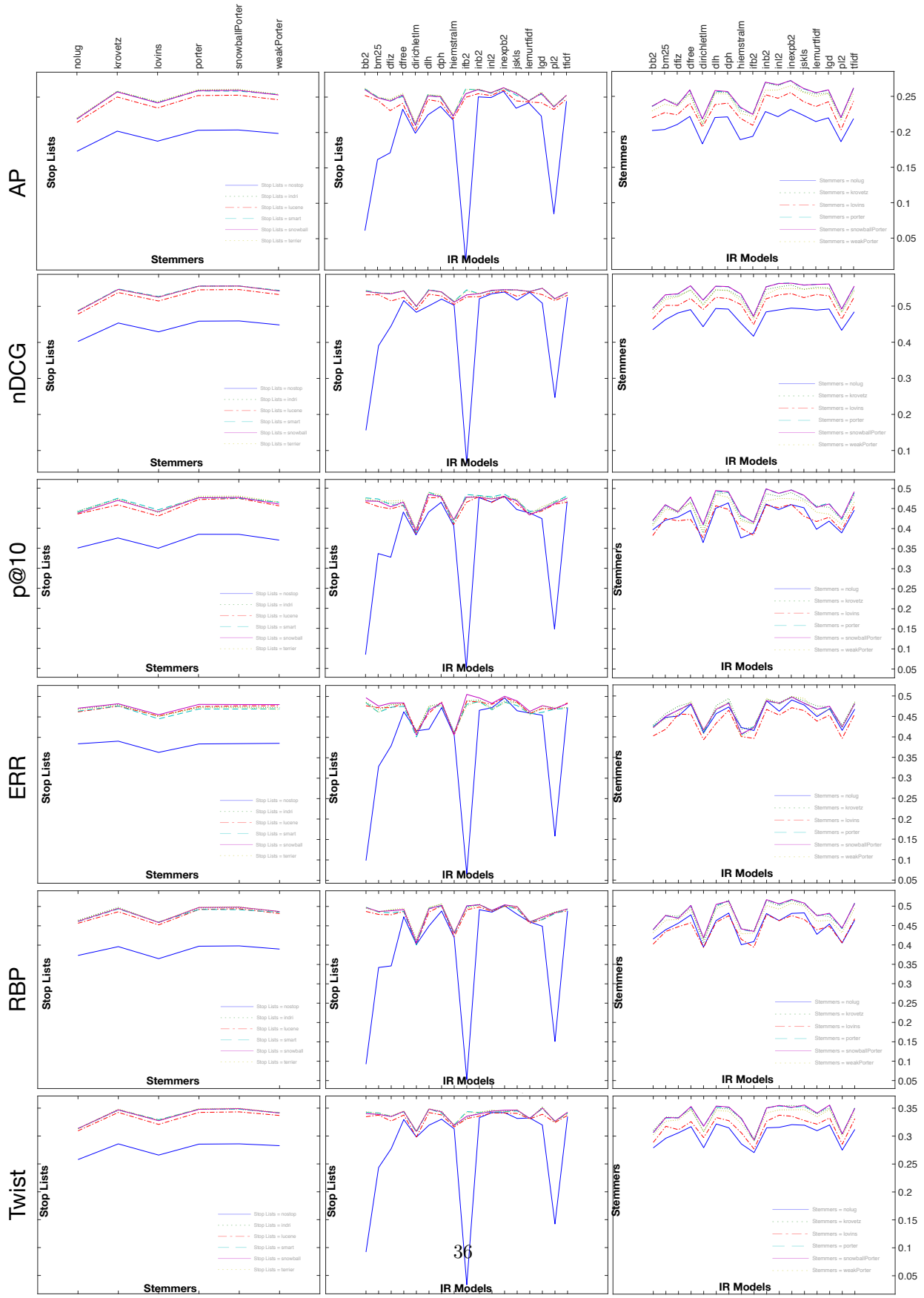


Figure 22: Interaction effects for all the considered measures on TREC 08, 1999, Adhoc test collection.

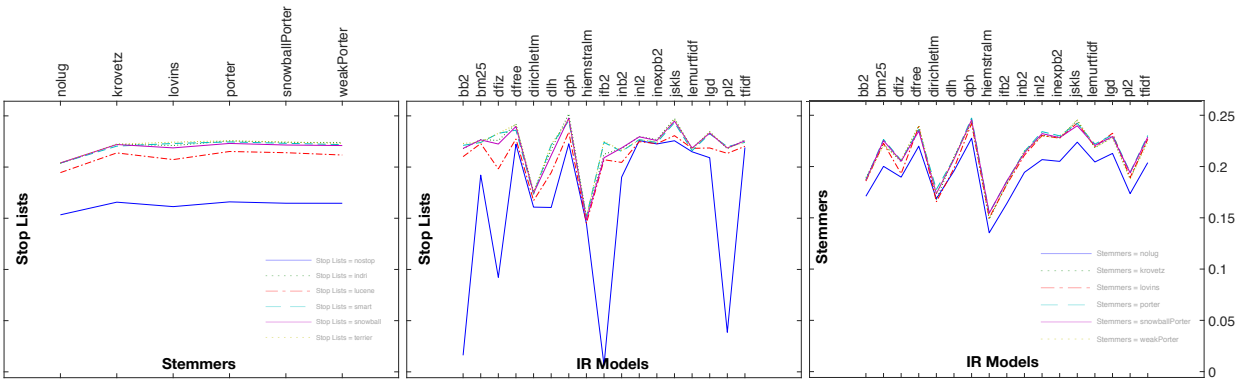


Figure 23: Interaction effects for the AP measure on TREC 09, 2000, Web test collection.

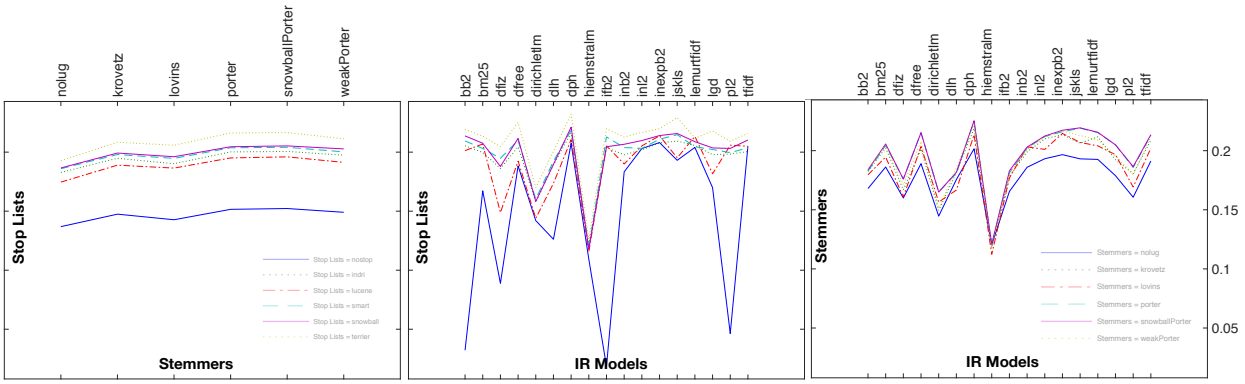


Figure 24: Interaction effects for the AP measure on TREC 10, 2001, Web test collection.

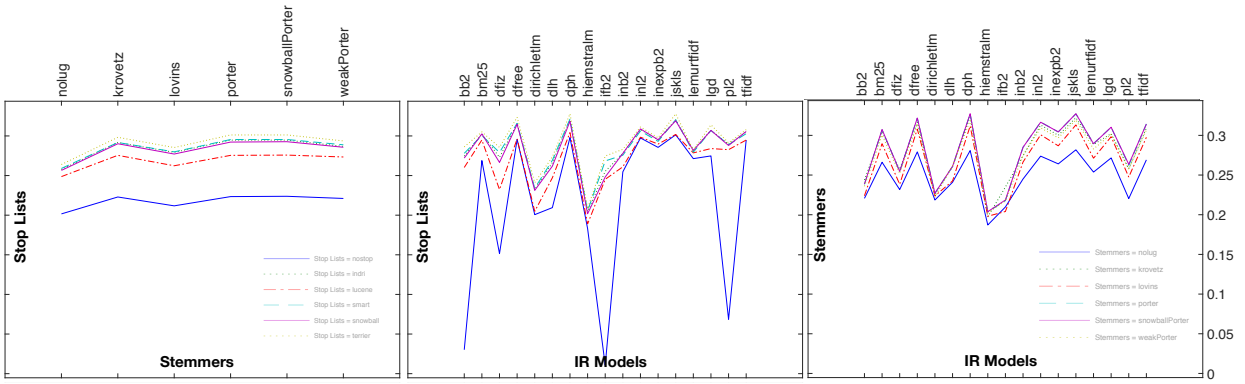


Figure 25: Interaction effects for the AP measure on TREC 14, 2004, Terabyte test collection.

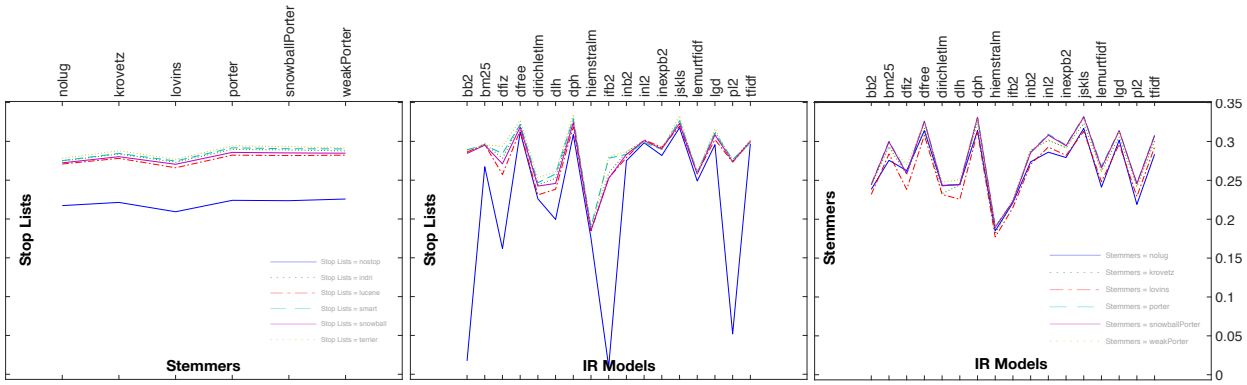


Figure 26: Interaction effects for the AP measure on TREC 15, 2005, Terabyte test collection.

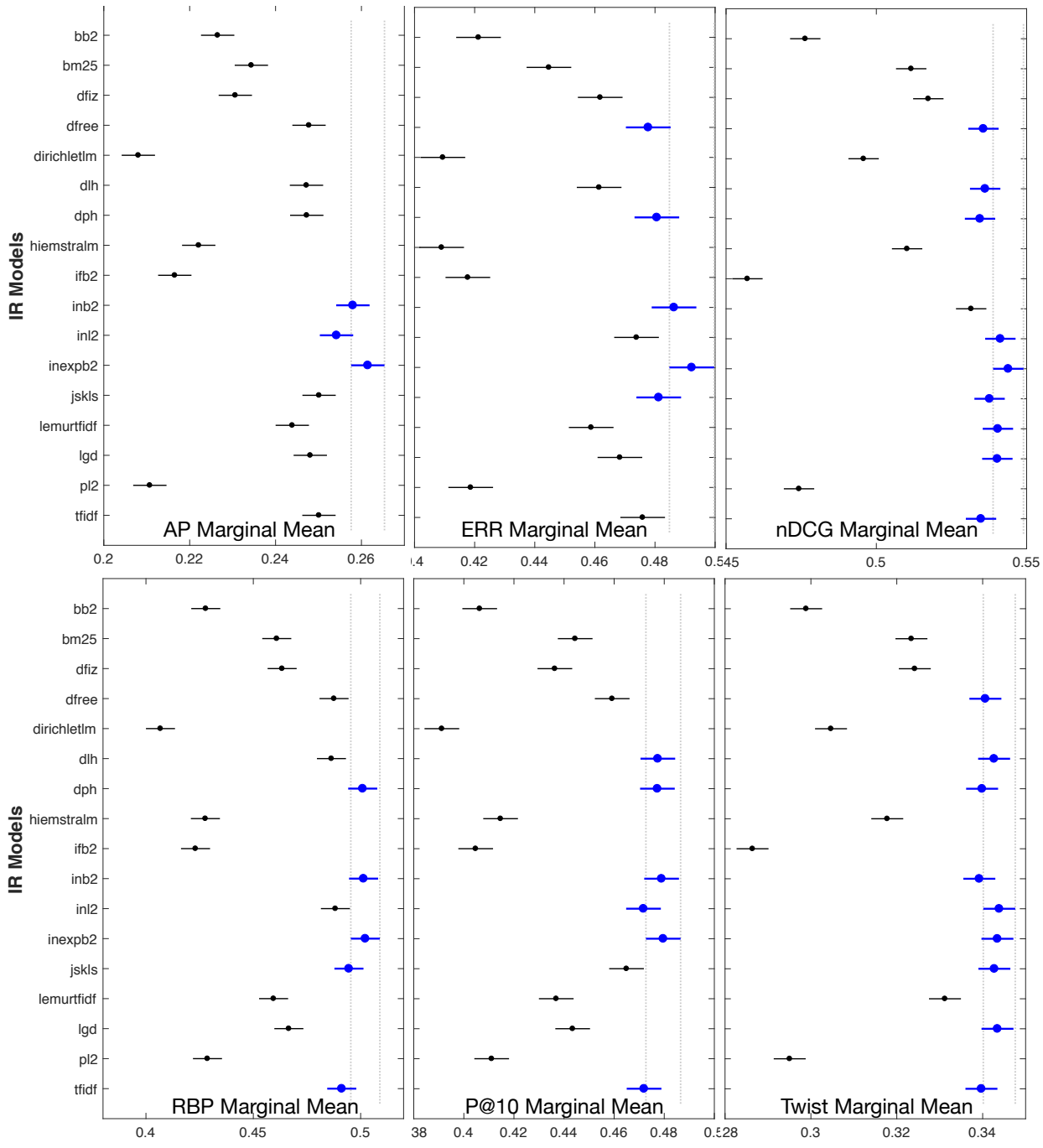


Figure 27: Tukey HSD test plots of the models for the TREC 08 test collection. This shows that `dirichletlm` model is always amongst the worst performing models and that `inexpb2` and `inb2` always belong to the top performing group.

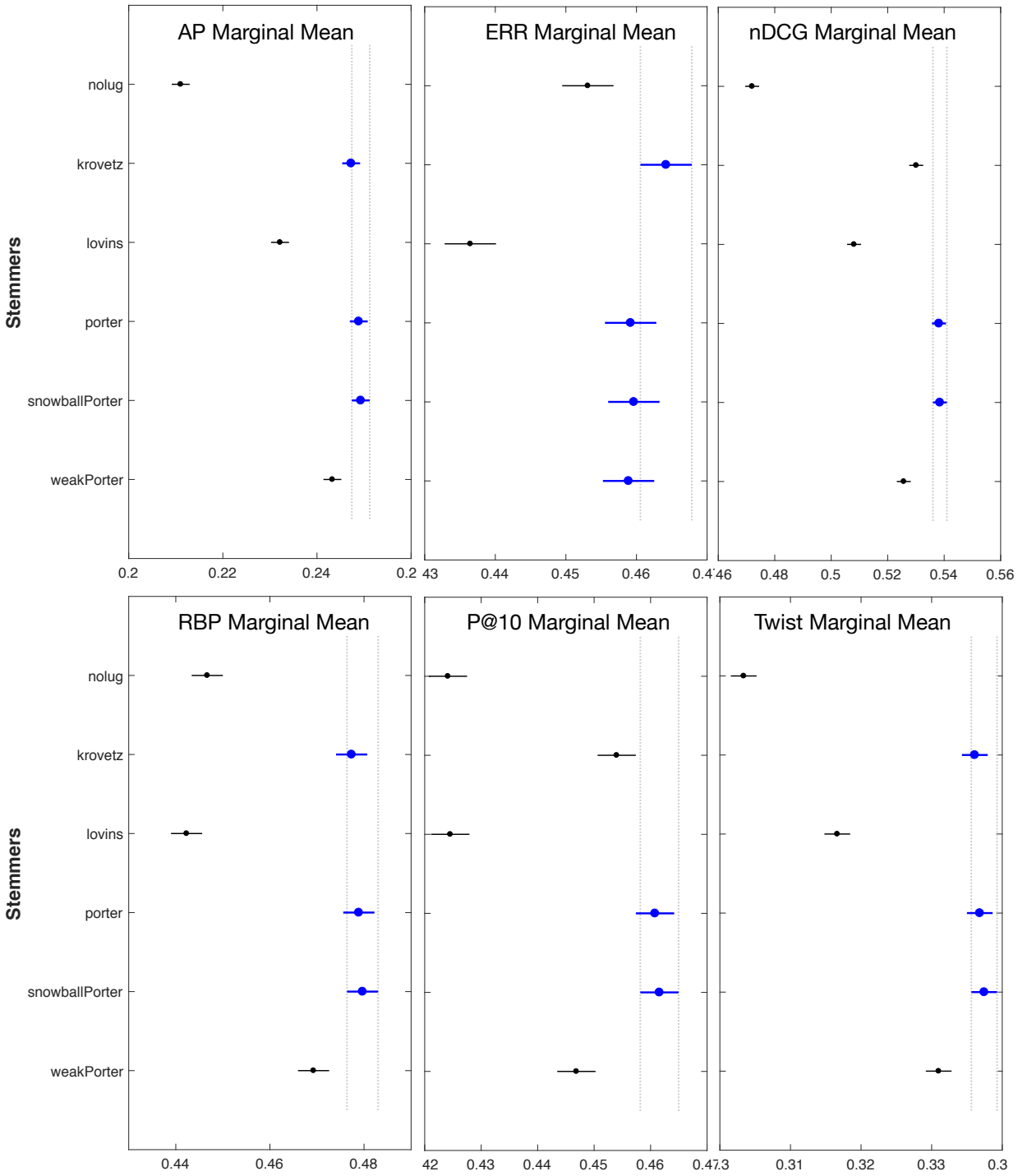


Figure 28: Tukey HSD test plots of the stemmers for the TREC 08 test collection. This shows that the `porter` stemmer is amongst the top performing stemmers for all the considered measures.

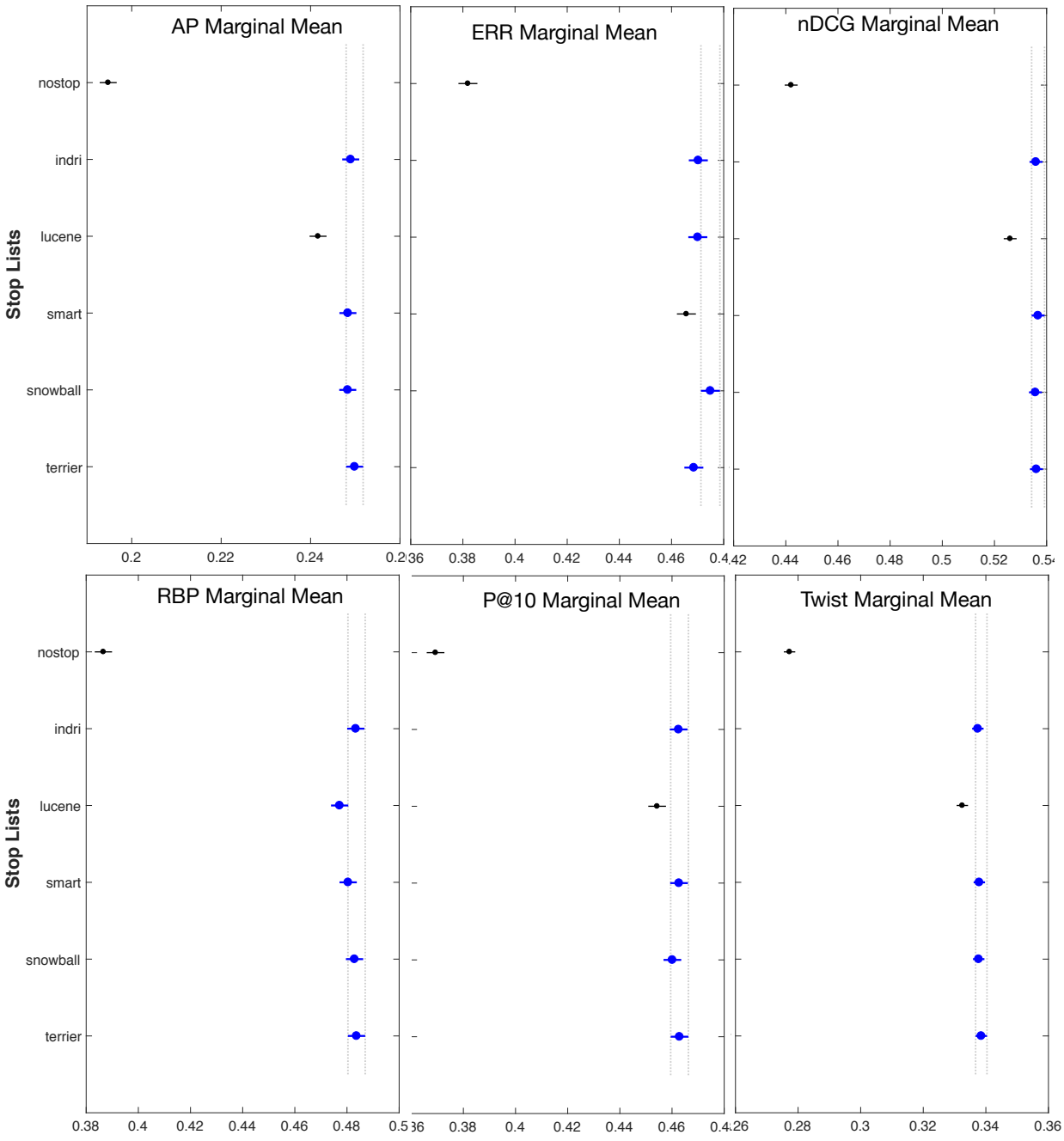


Figure 29: Tukey HSD test plots of the stop lists for the TREC 08 test collection. This shows that the `snowball` stop list is amongst the top performing stop lists for all the considered measures.