Manifesto from Dagstuhl Perspectives Workshop 17442

From Evaluating to Forecasting Performance: How to Turn Information Retrieval, Natural Language Processing and Recommender Systems into Predictive Sciences

Edited by

Nicola Ferro[1], Norbert Fuhr[2], Gregory Grefenstette[3], Joseph A. Konstan[4], Pablo Castells[5], Elizabeth M. Daly[6], Thierry Declerck[7], Michael D. Ekstrand[8], Werner Geyer[9], Julio Gonzalo[10], Tsvi Kuflik[11], Krister Lindén[12], Bernardo Magnini[13], Jian-Yun Nie[14], Raffaele Perego[15], Bracha Shapira[16], Ian Soboroff[17], Nava Tintarev[18], Karin Verspoor[19], Martijn C. Willemsen[20], and Justin Zobel[21]

1 **University of Padova, Italy** ferro@dei.unipd.it
2 **University of Duisburg-Essen, Germany** norbert.fuhr@uni-due.de
3 **Institute for Human Machine Cognition, USA** ggrefenstette@ihmc.us
4 **University of Minnesota, Minneapolis, USA** konstan@umn.edu
5 **Autonomous University of Madrid, Spain** pablo.castells@uam.es
6 **IBM Research, Ireland** elizabeth.daly@ie.ibm.com
7 **DFKI GmbH, Saarbrücken, Germany** declerk@dfki.de
8 **Boise State University, USA** michaelekstrand@boisestate.edu
9 **IBM Research, Cambridge, USA** werner.geyer@us.ibm.com
10 **UNED, Spain** julio@lsi.uned.es
11 **The University of Haifa, Israel** tsvikak@is.haifa.ac.il
12 **University of Helsinki, Finland** krister.linden@helsinki.fi
13 **FBK, Trento, Italy** magnini@fbk.eu
14 **University of Montreal, Canada** nie@iro.umontreal.ca
15 **ISTI-CNR, Pisa, Italy** raffaele.perego@isti.cnr.it
16 **Ben-Gurion University of the Negev, Israel** bshapira@bgu.ac.il
17 **National Institute of Standards and Technology, USA** ian.soboroff@nist.gov
18 **Delft University of Technology, The Netherlands** n.tintarev@tudelft.nl
19 **The University of Melbourne, Australia** karin.verspoor@unimelb.edu.au
20 **Eindhoven University of Technology, The Netherlands** M.C.Willemsen@tue.nl
21 **The University of Melbourne, Australia** jzobel@unimelb.edu.au

Abstract

We describe the state-of-the-art in performance modeling and prediction for Information Retrieval (IR), Natural Language Processing (NLP) and Recommender Systems (RecSys) along with its shortcomings and strengths. We present a framework for further research, identifying five major problem areas: understanding measures, performance analysis, making underlying assumptions explicit, identifying application features determining performance, and the development of prediction models describing the relationship between assumptions, features and resulting performance.

Perspectives Workshop October 30 to November 03, 2017 – www.dagstuhl.de/17442
2012 ACM Subject Classification Information systems → Information retrieval, Information systems → Recommender systems, Computing methodologies → Natural language processing
Keywords and phrases Information Systems, Formal models, Evaluation, Simulation, User Interaction
Digital Object Identifier 10.4230/DagMan.7.1.96

From Evaluating to Forecasting Performance: How to Turn Information Retrieval, Natural Language Processing and Recommender Systems into Predictive Sciences, *Dagstuhl Manifestos*, Vol. 7, Issue 1, pp. 96–139
Editors: Nicola Ferro, Norbert Fuhr, Gregory Grefenstette, Joseph A. Konstan

Dagstuhl Manifestos
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Executive Summary

This workshop brought together experts from information retrieval (IR), recommender systems (RecSys), and natural language processing (NLP). Common to these three neighboring fields is the challenge of modeling and predicting algorithm performance under varying application conditions, measured in terms of result quality. A particular challenge is that these methods create or affect a human experience, and so performance ultimately depends on human judgment of the quality of experience and performance. Progress in performance modeling and prediction would allow us to better design such systems to achieve desired performance under given operational conditions.

In this manifesto, we first consider the state of prediction in the three research disciplines, and then describe a general framework for addressing the prediction problem.

Research in IR puts a strong focus on evaluation, with many past and ongoing evaluation campaigns. However, most evaluations utilize offline experiments with single queries only, while most IR applications are interactive, with multiple queries in a session. Moreover, context (e.g., time, location, access device, task) is rarely considered. Finally, the large variance of search topic difficulty make performance prediction especially hard.

NLP has always engaged in both intrinsic evaluation of the steps in the language processing pipeline (e.g., language identification, tokenization, morphological analysis, part-of-speech tagging, parsing, entity extraction, classification, etc.) and in extrinsic, application-oriented evaluation (such as information retrieval, machine translation, and so on). The different goals of different applications mean that there is no one best NLP processing system, and also call into doubt the usefulness of intrinsic evaluations alone, since the improvement of one pipeline step might have little influence on broader application performance. Added to this, the performance of an NLP system in a new language or domain can be hard to predict, as it may depend on the existence of language resources to implement these pipelines.

RecSys generate predictions and/or recommendations for a particular user from a set of candidate items, often for a particular context. Like the other two areas, the field has a legacy of metrics, user experimentation research, benchmarks, and datasets. At a general level, current RecSys research aims at distilling the current large body of empirical knowledge into more systematic foundational theories and at learning from cumulative research. More specific issues include topics like auto-tuning of systems, exploration vs. exploitation, coping with context-dependent performance, and algorithm vs. system performance.

For a general framework for performance prediction, we identified 5 problem areas:

1. **Measures**: We need a better understanding of the assumptions and user perceptions underlying different metrics, as a basis for judging about the differences between methods. Especially, the current practice of concentrating on global measures should be replaced by using sets of more specialized metrics, each emphasizing certain perspectives or properties. Furthermore, the relationships between system-oriented and user-/task-oriented evaluation measures should be determined, in order to obtain improved prediction of user satisfaction and attainment of end-user goals.
2. **Performance analysis**: Instead of regarding only overall performance figures, we should develop rigorous and systematic evaluation protocols focused on explaining performance differences. Failure and error analysis should aim at identifying general problems, avoiding idiosyncratic behavior associated with characteristics of systems or data under evaluation.
3. **Assumptions**: The assumptions underlying our algorithms, evaluation methods, datasets, tasks, and measures should be identified and explicitly formulated. Furthermore, we need strategies for determining how much we are departing from these assumptions in new cases and how much this impacts on system performance.

4. **Application features**: The gap between test collections and real-world applications should be reduced. Most importantly, we need to determine the features of datasets, systems, contexts, tasks that affect the performance of a system.
5. **Performance Models**: We need to develop models of performance which describe how application features and assumptions affect the system performance in terms of the chosen measure, in order to leverage them for prediction of performance.

These five problem areas call for a research and funding agenda where basic research efforts should address the first three items above by laying new foundations for the IR, NLP, and RecSys fields and adopting a multidisciplinary approach to bridge among algorithmics, data management, statistics, data analysis, human-computer interaction, and psychology. Once these foundations are laid, subsequent research efforts should leverage them and exploit, for example, machine learning and artificial intelligence techniques to address the last two items in the above list.

Overall, the above research agenda outlines a set of "high risk, high gain" research topics and promises to deliver a major paradigm shift for the IR, NLP, and RecSys fields, by embracing a new radical vision of what should be at the foundations of those fields and targeting a technological breakthrough able to change the way in which academia and industry invent, design and develop such kind of systems.

Table of Contents

1 Introduction

Predictability is a fundamental attribute of daily life: we expect familiar things to behave in familiar ways. In science, predictability has taken on more specific meanings; our understanding of a system, model, or method is validated by our ability to predict performance or outcomes, often in a quantified form. A particular challenge for the systems regarded here is that, ultimately, they create or affect a human experience.

Questions we might like to answer in this context include the following:

- How reliable will a system perform over different tasks?
- What test materials (and at what scale) are required to establish performance to standards that imply predictability?
- Will the current performance of a system be robust to changes in its data or use, and what parameters or limits would indicate whether there is a risk to performance?
- Can performance uncertainty be quantified?
- How can we plan a move from a laboratory prototype to a system in operation?
- To what extent do performance metrics match user perceptions and experiences?
- What resources or configuration might be required to adapt a system to a new context or a new application?
- What resources might be required to maintain a system or confirm that it is continuing to perform?

In this paper, we first discuss the state of performance prediction in the areas of Information Retrieval (IR), Recommender Systems (RecSys), and Natural Language Processing (NLP). Then we present a general framework for addressing the prediction problem, and point out the corresponding research challenges.

2 Information Retrieval

2.1 Motivations for Prediction in IR

An IR system is successful if it provides the information that a user needs to complete a task, supports them in learning, or helps the user accomplish a goal. That is, the purpose of an IR system is to have impact on a cognitive state, and thus the value or correctness of an outcome is inherently subjective. A related challenge is that, typically, a single system is often relied on by a user for a wide range of unrelated activities, and that similar interactions from different users may be the consequence of different intents. This is in part a consequence of the fact that tasks can be underspecified, or ill-formed; or may be fluid, shifting during the course of an interaction; or may be progressive.

Validation via users, and inconsistencies in that validation, are therefore an inherent component of prediction. These validations are inherently more complex than specific questions such as comprehensibility of a text or ease of use of an app. [49] describes prediction as a challenge for evolving IR to an engineering science, but the problem in IR is even more complex, referring to human judgment rather than to measurement of certain technical properties.

Several types of prediction may be relevant in IR. One case is that we have a system and a collection and we would like to know what happens when we move to a new collection, keeping the same kind of task. In another case, we have a system, a collection, and a kind of

task, and we move to a new kind of task. A further case is when collections are fluid, and the task must be supported over changing data.

Current approaches to evaluation mean that predictability can be poor, in particular:

- Assumptions or simplifications made for experimental purposes may be of unknown or unquantified validity; they may be implicit. Collection scale (in particular, numbers of queries) may be unrealistically small or fail to capture ordinary variability.
- Test collections tend to be specific, and to have assumed use-cases; they are rarely as heterogeneous as ordinary search. The processes by which they are constructed may rely on hidden assumptions or properties.
- Test environments rarely explore cases such as poorly specified queries, or the different uses of repeated queries (re-finding versus showing new material versus query exploration, for example). Characteristics such as "the space of queries from which the test cases have been sampled" may be undefined.
- Researchers typically rely on point estimates for the performance measures, instead of giving confidence intervals. Thus, we are not even able to make a prediction about the results for another sample from the same population. A related confound is that highly correlated measures (for example, Mean Average Precision (MAP) vs normalized Discounted Cumulated Gain (nDCG)) are reported as if they were independent; while, on the other hand, measures which reflect different quality aspects (such as precision and recall) are averaged (usually with a harmonic mean), thus obscuring their explanatory power.
- Current analysis tools are focused on sensitivity (differences between systems) rather than reliability (consistency over queries).
- Summary statistics are used to demonstrate differences, but the differences remain unexplained. Averages are reported without analysis of changes in individual queries.

Perhaps the most significant issue is the gap between offline and online evaluation. Correlations between system performance, user behavior, and user satisfaction are not well understood, and offline predictions of changes in user satisfaction continue to be poor because the mapping from metrics to user perceptions and experiences is not well understood.

2.2 Successes in Prediction in IR

The IR field has always had a strong evaluation focus. Because we are always trying to measure what we do, and furthermore working on analyzing the measures and the methodologies, we have a lot of experience in thinking about what we would like to predict. Also, IR is fundamentally about supporting people working to complete some kind of task. For example, modeling IR as a ranking problem already makes an assumption on how to present results and how users will access the output of the system. Even when evaluation is abstracted away from the actual user, we realize this measurement gap must be bridged.

Shared evaluation campaigns (TREC[1], CLEF[2], NTCIR[3], FIRE[4]) have always played a central role in IR research. They have produced huge improvements in the state-of-the-art and helped solidify a shared systematic methodology, achieving not only scholarly

[1] http://trec.nist.gov/
[2] http://www.clef-initiative.eu/
[3] http://research.nii.ac.jp/ntcir/
[4] http://fire.irsi.res.in/

impact [9, 103–105] but also economic impact [94]. The model has been adopted by other areas, and the IR field has successfully expanded into broader Information Access problems. Scalability has always been a major concern in the field that has been pushed by evaluation campaigns, and it is not as much a critical problem in prediction for IR systems as it is in other areas of Information Systems.

As a result of a strong evaluation focus, we have built a lot of datasets, and these datasets have closely related characteristics: common data types, common tasks, common experimental setups, common measures. This has let us appreciate the difficulty of predicting effectiveness on unseen data, tasks, or applications. There is extensive research on test collection building and evaluation methodologies, e.g. on robustness of the pooling methodology [117], the sensitivity and reliability of our measures [16,98], the impact of inter-assessor agreement [109], how many topics to use [97], just to name a few, although it is not easy to extract general lessons from it.

These test collections have allowed us to study what types of queries can be predicted to work well [27] and to discover other characteristics of queries (such a temporal distribution of the topic [66]) that can also be used to predict precision on some queries. Query performance prediction [19, 61] is thus concerned with predicting how difficult a query will be rather than the performance of a system for a given query but it can be a useful starting point for more advanced types of prediction.

Modeling score distribution, i.e. determining how relevant and not relevant documents are distributed, can be considered an another potential enabler for prediction, as also suggested by recent work which explicitly links it to query performance prediction [28].

On a more theoretical level, Axiomatics (the formal definition of constraints in a space of solutions for a problem) have been successfully used to predict the performance of IR models [34], to understand the properties and scales of evaluation measures [36–39] and to reduce the search space of available quality metrics [6–8].

Reproducibility is becoming a primary concern in many areas of science [48] and, in particular, in computer science as also witnessed by the recent ACM policy on result and artifact review and badging[5] [42]. Increasing attention is being paid to reproducibility also in IR [40, 118] where discussion is ongoing: use of private data in evaluation [18]; evaluation as a service [59]; reproducible baselines [75] and open runs [110]; considering it as part of the review process of major conferences and in dedicated tracks, such as the new ECIR Reproducibility Track; and, the inception of reproducibility tasks in the major evaluation campaigns[6] [43]. All these aspects contribute a better understanding and interpretation of experimental results and clarify implicit and explicit assumptions made during IR system development, which are key enablers for prediction.

2.3 Priorities for IR Experimentation

The considerations sketched out above, analyzed against existing successes, suggest four broad priorities that should be reflected in experimental methodologies: uncertainty, offline versus online, failure analysis, and reproducibility. Other aspects include use features (of topics, documents, and context), the roles of measures, domain adaptation, and more application-specific issues such as individual queries versus sessions. We consider each of these in turn below.

[5] https://www.acm.org/publications/policies/artifact-review-badging
[6] http://www.centre-eval.org/

2.3.1 Priorities

Uncertainty

Our measures typically produce a point estimate, without confidence intervals or effect sizes. Statistical significance is not predictive, and does not quantify uncertainty, although researchers use them that way. We need measures that indicate bounds as well as averages, where we can indicate confidence bounds in the performance of a system on unseen data.

Offline versus online

Offline metrics at best weakly predict online effectiveness and user satisfaction. We need to understand how online effectiveness can be predicted more reliably, and what factors are responsible for the inconsistency. A particular factor is the single-query nature of most offline evaluation, while online experiences are iterative or progressive, that is, involve a session. Thus, the result of the complete session is what matters for users.

Failure Analysis

Failure analysis typically focuses on individual tasks where performance is extremely bad. The RIA workshop [15] followed this approach, but did not arrive at general conclusions for improving the systems considered. Instead, it was acknowledged that there are topics of varying difficulty, and thus various approaches for estimating query difficulty have been proposed (see e.g. [91]). However, the core problem is still unsolved: which methods would be suitable for improving the results of 'difficult' queries?

Reproducibility and replicability

The ACM policy on Artifact Review and Badging[7] distinguishes between replicability (different team, same experimental setup) and reproducibility (different team, different experimental setup). Reproducibility is a key ingredient for prediction since not only it enables the systematic replication and understanding of experimental results – a key aspect to ensure robustness of prediction – but also studies how robust experimental results can be ported to new contexts and generalized. However, we still lack commonly agreed methodologies to ensure the replicability, reproducibility and generalizability of experimental results, as well as protocols and measures to verify and quantify the reproducibility of experimental results.

2.3.2 Other open issues

Measures and resources

It is clear that measures vary with regard to predictability. We need to develop good practice recommendations for selecting and using evaluation metrics: which metrics are suitable for a given task, scenario, or dataset? How should we interpret inconsistent quality signals? How should we deal with multiple, complementary quality signals (e.g. Precision and Recall)?

System comparisons are somewhat stable on typical small sets of topics, but concerns about the sampling population mean that to increase our understanding we need vastly

[7] https://www.acm.org/publications/policies/artifact-review-badging

larger topic sets; and arguably these should be characterized by kind of task and kind of interaction.

Finally, we also need a better understanding of what IR evaluation measures are and what their properties are. Indeed, we need to go beyond what we empirically [16, 17, 96] and theoretically [6, 7, 36, 38, 100] know today about IR evaluation measures and turn this into knowledge about evaluation measures affect prediction.

Contexts

What factors affect domain adaptation? Is it reasonable and effective to consider a domain as comprised of a number of tasks, where each has its own success criteria that, in turn, is reflected into a measure? What would constitute an actionable description (by means of features, tasks, collections, systems, and measures) of what is required to move from one domain to another? A specific example is the difference between language-dependent and language-independent factors, both at system level and at domain level, since they may require different kinds of prediction techniques.

3 Natural Language Processing

Current research in NLP emphasizes methods that are knowledge-free and lack explanatory power, but are demonstrably effective in terms of task performance. This has the consequence that small changes in the application scenario for an NLP system has an unpredictable impact on performance. We need to make the process of developing NLP systems more efficient.

3.1 Motivations for Prediction in NLP

We regard predictability of Natural Language Processing (NLP) system performance as the capacity to take advantage of known experiences (methodologies, techniques, data) to minimize the effort to develop new high performing systems. A key issue that impacts our ability to predict the performance of an NLP system is *portability*. Under this perspective we need to consider the following portability aspects:

- cross-language portability
- cross-corpus portability
- cross-domain portability
- cross-task portability

As an anecdotal example that motivates the interest in predictability, we can look at a project [52] for automatic classification of radiological reports in a hospital department. It was developed as a supervised system, which required a significant annotation effort by domain experts. The same technology was then proposed to the same department of another hospital, which asked for an estimate of the annotation effort, i.e. time of domain experts needed for adapting the system to a different classification schema. At this point it became clear that there was a lack of predictive methodologies and tools. At the end of the day, the new hospital was not convinced to invest in the technology due to the unclear investment that would be required.

Another anecdotal example with a more positive outcome is the Software Newsroom [65] which is a set of tools and applied methods for automated identification of potential news from textual data for an automated news search system. The purpose of the tool set is to analyze data collected from the Internet and to identify information having a high probability of containing new information. The identified information is summarized in order to help understanding of the semantic content of the data, and to assist the news editing process. The application had been developed for English and initially did not transfer well into Finnish. The problem was attributed to the fact that data was collected from Internet discussions and that the language was probably substandard. Attempts to fix this did not yield performance improvements. Later it became clear that words with certain syntactic and semantic properties are effective when building topic models for English, at which point it could be demonstrated that words with similar properties in Finnish are useful as well. Correctly extracting such words required knowledge about the special characteristics of the Finnish language.

A challenging aspect of typical NLP components, e.g. part of speech tagging, named entity recognition, parsing, semantic role labeling, is that they require a significant amount of human supervision, in the form of annotated data, to train reliable models. This an issue clearly impacts the portability of both individual components and more complex systems that depend on pipelines of such components. Several efforts are being made in the NLP field to reduce and to predict the amount of such supervision, moving towards less supervised algorithms. We mention a few of these research directions:

- the use of unannotated data and distributional representations of words, i.e. embeddings, as features for machine learning algorithms;
- distance learning approaches, exploiting the role of available resources, e.g. taxonomies, dictionaries, background knowledge to infer training examples;
- active learning techniques, in order to select instances to be manually annotated to optimize the performance of a system;
- projections of annotations across aligned corpora, from one source language (typically English) to a target language, to reduce the effort to develop training data.

Although the above research streams are producing significant advancements in terms of portability, we feel the need for fundamental research where predictability of NLP systems is addressed in the broader context of cross-language linguistic phenomena, characteristics of corpora, domain coverage and particular properties of the task.

3.2 Successes in Prediction in NLP

One traditional technique for predicting performance is to perform post-hoc data degradation. In TREC-4 (1995), the 'Confusion Task' compared performance of query retrieval using corrected OCR text against text with 10% and 20% recognition errors [67]. In this way, given an evaluation of the recognition rate of an OCRed collection, one could predict the performance degradation compared with a corrected collection. Similarly, TREC-9 analyzed the effect of spelling errors on retrieval performance, and the absence of word translations in cross-language information [79]. More recently, this method of post-hoc corpus degradation was used to show that at least 8 million words of text is needed to achieve published results in word embedding tasks, such as similarity and analogy [56].

This degradation technique provides a negative prediction of relative performance to a known system and known input testbed, but does not allow us to predict how well a given technique will work on a new language, or a new corpus, or a new domain.

Retrospective analysis can show that different domains have different measurable characteristics that correlate with some system performance. For example, biological texts have a greater entropy that correlates with a degraded performance of named entity recognition compared with performance on edited newspaper text [85]. Word sense disambiguation has been shown to degrade across a number of factors that can be calculated before experimentation [114].

The Software Newsroom [65] is an example of adapting a news discovery system from English to Finnish where there was a need to know linguistics and language technology to understand which parts were similar and which parts needed to be adapted. Similarly, evaluation of NLP tools applied across domains demonstrates that adaptation is required to port tools, e.g. from general English to a more specialized context such as biomedicine [107].

Work in adapting NLP technology to new languages, particularly to low-resource languages, begins with the problem of complex requirements for building NLP systems, including both annotated data sets and tools for analysis of linguistic data at various levels, such as the lexical, syntactic or semantic level. Recent research in *transfer* or *projection* learning has shown that it is possible to leverage data in one language to develop tools for the analysis of other, even quite linguistically distinct, languages [31]. However, this research has also demonstrated the need for resources to facilitate transfer, ranging from complementary resources such as parallel corpora and bilingual dictionaries to broad overarching frameworks such as the Universal POS Tagset [88] and the Universal Dependency representation [84]. In short, NLP system development requires either task-specific annotated data sets, a strategy for inferring annotations over data that can be leveraged, or a framework that facilitates model transfer through shared representation.

Current attempts at learning morphological reinflection for various languages have met some initial success using Deep Learning where it has been shown that approx. 10000 training cases can render a performance around 95% correct results for many languages [25]. Some of the systems benefited from additional unannotated data to boost performance.

In a keynote talk at GSCL 2017 (http://gscl2017.dfki.de/), Holger Schwenk (Facebook, Paris) presented recent advances in deep learning in the field of NLP, showing how Machine Translation could be understood as a cross-lingual document search. As deep learning is performing feature extraction and classification in an automatic fashion, this technology can be deployed in various NLP tasks, for example machine translation. Word embeddings, neural language models and sentence embeddings are leading to an application of multilingual joint sentence embeddings, supporting high quality translation. The further development of such approaches could lead to a better integration of NLP systems, using the generated vector spaces for cross-language, cross-corpus, cross-domain and cross-task information sharing.

3.3 Priority Next Steps in NLP Research

We believe that, in order to improve predictability of NLP systems, research in the next years should focus on innovative, fine-grained, shared, methodologies for error analysis. We advocate evaluation measures as well as techniques able to provide both quantitative and qualitative data that explain the behavior of the system under specific experimental conditions. We expect to move from ad-hoc and mainly manual error analysis to shared and automatic tests through which we determine reliable predictability indicators.

Particularly, it is expected that new error analysis methods can provide empirical evidence of system failures based on the whole complexity of the context in which the system operates,

including linguistic cross-language phenomena, the properties of the data (corpora, resources, knowledge), the domain characteristics, the specific task the system is supposed to addresses, and the role of the human users that interact with the system. Test suites, as discussed in Section 5.5.2, could support error analysis structured by cases.

The kind of expected error analysis has to be enough fine-grained to give precise insight about the causes for a system/tool not to deliver the expected results, including:

- Are the corpus/data sets or other (domain) resources appropriate?
- Has the right/adapted algorithm been selected?
- Is the selected/developed gold standard the relevant one?
- Are we using the right metrics/measures?
- Are we using the right amount of (linguistic) knowledge?
- Are we using the right type of representation of the data and the features, also in combination with data/tools that are not specific to NLP?
- Check the validity of the assumptions of what the system should deliver and at what level of quality, taking into consideration IT-performance, but not focusing only on the measures.

3.3.1 Desiderata

In the last few years, the increasing availability of large amounts of language data for a subset of natural languages as well as the availability of more powerful hardware and algorithmic solutions, has supported the re-emergence of machine learning methods that in certain applications, for example Neural Machine Translation (NMT), has relevantly improved performance in terms of objective measures. In the light of those developments, we need to re-think the way we develop and deploy NLP systems, taking into account not only linguistic knowledge but also technological parameters. Conversely, excitement about the performance of neural network approaches should not close our eyes on specific linguistic features and language properties. We need to embark on a new theory of the field of natural language processing.

To sum up, the NLP field is missing a comprehensive diagnostic theory for NLP systems. A consequence of using powerful diagnostic tools will be a substantial rethink of the way we develop and make NLP systems more adaptable to new languages, data, tasks, applications and scenarios, including when this involves other types of technologies. A long-term opportunity in this direction is that of NLP systems able to auto-adapt themselves to a changing environment, predicting adaptations on the base of diagnostic tools.

4 Recommender Systems

4.1 Motivations for Prediction in RecSys

Introduction and History. Even in the earliest days of recommender systems, predicting performance was seen as critical. The earliest recommender systems companies hired "sales engineers" whose job was to evaluate the potential gain prospective customers would have from deploying a recommender in their applications. A typical example was reported in talks by John Riedl. The recommender systems company Net Perceptions sent a team of sales engineers to work with a large catalog retailer. To make the sale, they had to import the retailer's database into their system and run side-by-side experiments with phone operators making suggestions from the legacy or new system. It was a multi-week, multi-person effort

that fortunately led to a successful sale and deployment. Not all such efforts were successful, highlighting the desirability of predictive models of performance that can more efficiently support deployment and tuning decisions. This section reviews examples of cases where such prediction is needed.

Case 1: ROI Improvement for Mobile News. A company develops start screens for mobile devices providing news items that are pushed to their users once they turn on their devices. They have a few million users and agreements with news agencies in various countries. Typically they provide a list of 8 items when the user turns on the device. Their business model is based on user clicks, so they are interested to improve the Click-Through Rate (CTR) and user engagement. They want to examine whether personalization of the provided list of items would improve these measures and if the investment in the development and implementation of personalization would be returned (ROI). Currently they provide the list based on the nationality of the user, recency of items and some notions of popularity. Several algorithms were considered: variations of content-based and collaborative algorithm, and diversity to expose more items and enhance the ranking of popular items. Performed off-line analysis yielded interesting results that were not always consistent for different parts of the data. A/B tests are very costly and can be done very selectively, since they don't have an experimental infrastructure, thus deploying algorithms and testing different variations places huge effort on production. The challenge is to predict which algorithm or the combination of algorithms would provide the expected ROI. Is it possible to predict for the company if they should invest in personalization. Can that be inferred from the offline test results, from the dataset, task, algorithm features, or from success and failure stories.

Case 2. To Personalize at All? An online education company has a large and expanding library of courses, and currently has no personalized mechanisms for recommending courses to their learners. Their system is based on three forms of discovery: (a) search for courses that match relevant keywords, (b) lists of most-popular courses, both overall and within broad top-level categories (e.g., "most popular computer science courses"), and (c) marketer-selected lists of courses to promote in themes (e.g., the April theme was "new beginnings") with a set of promoted introductory courses in different categories. The company is interested in determining whether there would be significant benefits to adding a personalized recommender system to their site.

Given the characteristics of the education company's dataset (number of learners and courses, distributions of courses-taken and learners-enrolled, etc.), can we model and predict the performance of a recommender system for this application? Today, we cannot. Our choices are to offer "advice from experience" or to offer instead to go through the data engineering effort to implement the recommender in order to justify its feasibility. Neither is a particularly satisfying alternative for a company (or expert) hoping to make an informed decision to invest without incurring substantial cost.

Case 3. How Much Value Do We Have from Certain Data? Data collection is both expensive and potentially interfering with the privacy of clients. With increasing regulation on which data and how data is stored, such as General Data Protection Regulation (GDPR) in the EU, this may also be difficult in practice. In some cases, however adding the right amount and kind of data can improve the quality of predictions. At the moment, we do not have a formal way of assessing how much, and what kind, of data will translate to a specific return. While there is some heuristic consensus on which dimensions may be relevant, and that these depend on dimensions of the domain, client, and tasks, this knowledge is not systematically structured or cataloged.

4.2 Successes in Prediction in RecSys

This section outlines key areas of success with regard to prediction in recommender systems, outlining the state-of-the-art and gaps to motivate the priority areas in Section 4.3.

This section discusses the following topics.

- Noise and inconsistency
- Data Sets
- Metrics and Evaluation Protocols
- Toolkits
- Subjective evaluation
- Meta-learning

4.2.1 Noise and inconsistency

Accuracy of prediction is limited by the by noise (e.g., so called *shilling malicious ratings* [74]), anchoring effects due to original ratings [2], as well as by the inconsistency of rating by end-users [4].

Progress has been made in terms of detecting noise, and in the development of de-noising techniques both in terms of algorithms [5] and interface design [1]. There is an understanding that prediction accuracy may be restricted by the upper bounds of such factors. However, the nature of noise and its role in relation to prediction accuracy is not completely understood. For example, it is still not clear to which extent changes in rating behavior are due to inconsistency, versus how much reflects a genuine change in opinion.

4.2.2 Data Sets

Recommender research been advanced by many public data sets containing user consumption data, suitable for training collaborative filters and evaluating recommender algorithms of various forms. These have enabled direct comparison of algorithms in somewhat standardized environments. These include:

- EachMovie [78], movie ratings from a movie recommender system operated by the Digital Equipment Corporation (DEC).
- MovieLens [60], a series of movie rating data sets released from the MovieLens movie recommender system operated by GroupLens Research at the University of Minnesota. Recent versions include the Tag Genome [108], a dense matrix of inferred relevance scores for movie-tag pairs.
- Jester [54], a set of user-provided ratings of jokes.
- NetFlix [13] (no longer available), user-provided ratings of movies in the NetFlix DVD-by-mail system; this data set was the basis for the NetFlix Prize, which awarded $1M for a 10% improvement in prediction accuracy over NetFlix's internal recommendation algorithm.
- BookCrossing [116], book ratings harvested from an online book community.
- Yahoo! Research publishes a number of data sets, including movie ratings and music ratings.
- CiteULike provided access to user bibliographies of research papers.
- Amazon rating data collected by [62].
- Yelp regularly provides data sets of business ratings [115].
- The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. [14].

In addition, the RecSys Challenge has regularly made new data sets on news, jobs, music, and other domains available to the research community, and there are suitable data sets through a number of Kaggle competitions and other sources, such as the NewsReel labs within CLEF [70, 76].

The "challenge" format around many data sets has provided a boost of energy in recommender systems research, with teams competing to provide the best performance around a single data set.

4.2.3 Toolkits

The recommender systems research community has a long history of publicly available or open-source software for supporting research and development. Early work on item-based collaborative filtering was supported by SUGGEST [30, 69]. Throughout the last decade, a number of open-source packages have been developed. Currently-maintained packages that are used in recommender research include LibRec [58], RankSys [99], LensKit [32], and rrecsys [119, 120]; Rival [95] provides cross-toolkit evaluation capabilities. These toolkits provide varying capabilities: some focus on algorithms or evaluation, while others provide both; some support primarily offline operation and batch evaluation while others have direct support for live use in online systems. There have also been a number of toolkits in the past that are no longer being maintained, such as MyMediaLite [51], and others that have pivoted away from a focus on recommendation such as Apache Mahout [10], in addition to algorithm-specific packages such as SVDFeature [22] and non-recommender-specific software such as XGBoost, Torch, and TensorFlow.

4.2.4 Subjective Evaluation

The goal of a recommender system is to provide personalized support for users in finding relevant content or items. If we want to predict or model whether that goal is actually achieved, researchers have realized that we should move beyond accuracy metrics to see if algorithmic improvements actually change the experience users have with the system [80]. This has led to several conceptual models that also provide subjective measures and scales of users' quality perceptions and evaluations of recommender systems [89, 113]. Building on this earlier work and other work on technology acceptance and attitude models, [73] argue that their user-centric framework [72] provide an "EP type" theory than can [E]xplain and [P]redict user behavior given the specific conditions of the recommender system under investigation. In other words, the framework goes beyond user studies that only qualitatively inspect user satisfaction or large scale A/B tests that only quantitatively look at the impact of a system change on user behavior. It aims at determining the factors why particular experimental conditions (i.e. a change in objective diversity of the algorithm) can change user experience (i.e. choice satisfaction) and user interaction behavior (increased engagement) by looking at the intermediate concept of subjective system perceptions (e.g., subjective perceptions of diversity and accuracy). For example a study [112] shows that user perceptions of accuracy and diversity mediated the effect of the diversification of recommender output on the experienced choice difficulty and user satisfaction showing that only if these subjective perceptions are changing, we can predict user experience to change.

4.2.5 Meta-Learning

Recommender system algorithms can be complex. Usually, they are configured specifically for specific data, users, and tasks and are optimized for specific desired measures. The construction and tuning of RecSys algorithms is typically done manually by human experts through try-and-err testings. It is desired to automatically explore the vast space of possible algorithms with the vision of enabling the prediction of which algorithms will perform well for a given dataset, a set of users, task, and performance metric using meta-learning techniques. A combination of features extracted from a given dataset, task, users, along with algorithm configuration, and the discretized result of a specified performance metric would make a labeled meta-training learning instance. One major challenge would be to learn the set of features of the dataset, users, tasks, that can affect the results. Another major challenge would be to collect enough instances for a large variety of datasets, algorithms, tasks and measures that would enable valuable learning. For this challenge the following possible sources can be considered: 1) a corpus of datasets and the corresponding learning results that will be provided collected by the community as a joint effort that should be promoted 2) data and information extracted from machine learning competitions (e.g. Kaggle). It may be possible to extract relevant information also from academic paper results. Nevertheless, to address the challenge of learning in the vast search space of possible algorithm and their specific configurations, ML techniques should be designed to allow a system to learn and capture insights and experiences in order to guide the selection of algorithms. Previous research showed that meta-learning can be successfully used for selecting the best model for decomposing large learning tasks such as [93], selecting the best setting for multi-label classification tasks and even recently for selecting the best collaborative filtering model for recommender systems However, to have the supervised meta-learning successful, a joint community effort to collect learning instances could be beneficial.

4.3 Priority Next Steps in RecSys Research

This section outlines the identified priority next steps. These are organized into three broad categories:

- Developing the Foundations for Rigor
- Learning from Cumulative Research
- Specific Challenges

4.3.1 Developing the Foundations for Rigor

There are several prerequisites that the field needs to achieve in order to place the remainder of the research we propose on a rigorous foundation.

4.3.1.1 Taxonomizing Cases

Today recommender systems are used in many domains and for different purposes. For example, in addition to recommending content for consumption, researchers have also started using those systems to incentivize user to create content e.g. [53]. Given this broad spectrum of use cases and applications, performance evaluation protocols are often tailored or anchored in the context of the recommender system use case. In order to learn from these cases, we need a rigorous and consistent way of describing them.

There are two levels of description needed in order to facilitate rigorous learning from cases. The first is agreement on the things necessary to describe a *case*, which we take to mean a set of research findings in a particular recommendation situation. The case consists of at least the following properties:

- The domain
- The system or experiment goals
- The target users
- The user task(s) within the domain
- The user's characteristics considered for recommendation
- The data
- The algorithms
- The experimental design and evaluation protocol (online or offline)
- The metrics and statistical analysis

The second level is the means of describing each of these properties. Significant research, detailed in later sections, will be needed in order to make this possible. For example, we need to know what properties of a user task and characteristics need to be captured in order to facilitate generalization and learning. Different tasks can have very different requirements, changing even the direction of certain optimization criteria; while recommending songs in a music recommender that a user has listened to before will usually have a positive impact on user satisfaction, recommending old news in a news recommender system will have a negative impact on satisfaction. We believe it is critical for the future success of recommender systems to develop a taxonomy of tasks that will lend itself to create task models connecting user goals of recommender systems with their objective, subjective evaluation metrics and output metrics. Herlocker et. al. [63] developed an initial description of end user goals and tasks of recommender systems (e.g. annotation in context, find good items, recommend sequence, ...). We suggest evaluating this research as a starting point for the development of a more formal taxonomy and evaluate literature since then to also develop different dimensions of a taxonomy of tasks (e.g. domain, time-sensitivity, content creation, or cost to consume).

Developing this shared understanding will enable the field to move forward and develop predictive knowledge from the current and future body of research findings.

4.3.1.2 Standardizing Evaluations

To ensure the replicability and comparability of results, the field needs to establish standards for measures and evaluation protocols. There are too many different ways to calculate what is labeled as the same measure. Even well-understood metrics such as Root Mean Square Error (RMSE) have important differences in their application: how unscorable items are handled and whether scores are averaged per-user or globally. The community needs to establish standard definitions and protocols for both metrics and for best practices in managing the broader evaluation (e.g., how to split data for cross-fold validation to evaluate performance on common tasks). Further research is needed to establish some of the standards, for example how best to mitigate popularity bias [12] and misclassified decoys [26, 33] and the role of time in splitting data sets. As metrics are standardized, the community should also provide standard test cases for use in acceptance tests to validate new implementations. A standard resource for recomputing or labelling historical results could also be used to assess new individual implementations.

There will likely always be the need to occasionally deviate from standard calculations to answer particular research questions. A standardization effort, however, can lay out the

option space and describe defaults and best practices for standard tasks; authors should provide clear justification when they deviate from the defaults the community agrees upon.

The standardization also needs to lay the decision points in experimental designs, such as what happens when an algorithm cannot produce recommendations for a user: does it get 'forgiveness' and have that user ignored in its assessment, does it get penalized as if it recommended nothing useful, or does it use some fallback method to ensure all users receive recommendations? There is not necessarily a best answer to these questions. Community guidelines should lay them out so that authors are aware that they need to make, describe, and justify a decision about each of them instead of relying on accidental properties of implementation details, and guidance on sensible defaults for common recommendation scenarios would help future researchers.

As the community decides upon standards, toolkit implementers should implement them and ideally make them the default operation, with appropriate thought given to backwards compatibility for their users.

There are two additional immediate first steps to promote rigor even before standardization is achieved. **Paper authors** should report sufficiently complete details on their evaluation protocols, algorithm configuration, and tuning to enable readers to reproduce them with exact replication of original decisions. **Toolkit implementers** should document the expected evaluation results of well-tuned versions of their algorithms on common data sets to provide a reference point for authors and reviewers to assess claimed baseline performance.

4.3.2 Learning from cumulative research

To enable us to learn from previous and ongoing research, we are assuming that there is a definition of a case. Such a case may include a description of the case, a description of the dataset (underlying assumptions, algorithm parameters, outputs, etc.) as well as a link to the resulting paper).

This repository should be both collaborative and machine readable. Users may add and remove content, with moderation of who can edit the repository, and how information is removed, to ensure completeness and consistency. A not trivial aspect in building this repository is the analysis of the existing body of research in order to transform it to standard case descriptions; this can be done either manually, requiring quite a lot of effort, or automatically, even if this is going to be a challenge. However, once this repository has been bootstrapped, adding new cases will require just a low-cost (for humans) and standardized procedure to be followed.

The repository will open up several new interesting possibilities:

- Meta-analysis is a well-understood technique in domains such as medical studies, where statistical confidence accrues through aggregating the evidence from numerous research studies. Medical research has the benefits of both more controlled studies and a long tradition of publishing results in a manner that explicitly supports such meta-analysis. Recommender systems will need to evolve its research to support these techniques, including such steps as:
  - developing standard templates for reporting results from recommender system evaluations and experiments and a disciplinary culture of reporting these with research results
  - developing characteristic parameters for datasets, users, and tasks
  - developing and testing predictive models over the diverse characteristics of these
- Creating a mechanisms for comparing commonalities and variabilities among cases, in order to support researchers in making decisions. This will help us identify recurring successful

cases, and failure cases, and the characteristics of both kinds of cases. Moreover, analysis of previous cases will also allow us to identify aspects of cases that are not sufficiently covered by current research. With time, this mechanism may become increasingly automated.

- As we aim at being able to predict the level of success of a recommender system before using it and even before developing it, failure analysis becomes of major importance as a tool to fix our prediction models. Given the actual results vs. the expected we should be able to reason about the causes of the mismatch or failure. Indeed, failure analysis/correction can be done either by analyzing the model that was discussed or by adding it as a case for deep-learning based meta-analysis where cases of systems that include data characteristics, task characteristics, algorithmic characteristics objective and subjective measures are analyzed to identify "successful" or "to be" systems.

Finally, to keep the repository a primary tool for research, we need to envision mechanisms to encourage contributions to it such as dedicated workshops and tutorials, both physical and online. These workshops will focus on cases analysis in the areas identified as gaps or failures. These workshops will also enable us to share what can be learned from the repository, as well as continue to help grow the case base.

Properties of data sets and algorithms

Different recommender algorithms attempt to exploit different properties of the data, e.g. user-based collaborative filtering leverages the assumption that people who are similar will like similar things. Content-based filtering uses textual features to represent items with the hopes of finding related items. However, the performance of these individual algorithms will depend heavily on the data set and the distribution of the properties being exploited and how they relate to each other. For example sparse datasets mean neighborhood algorithms perform poorly unless some latent factorization model is employed. Similarly, if content filtering is using a textual representation of items or people but if those base feature vectors don't accurately represent or reflect the items or user preferences then no tuning of the similarity measure or ranking will create significant additional value for the user.

Connecting this back to a more formal task taxonomy, we need to create a clear list of measures and distributions that will help guide the identifying the right family of algorithms based on the properties of their dataset and the class of task to be evaluated. By linking the performance of individual classes of algorithms, simply looking at the distribution and measures of the data can help predict performance and identify possible weaknesses before requiring a complete end-to-end evaluation.

Predictability limitations in data sets

Differing assumptions, measures and testing strategies lead to wildly varying performance across datasets. One issue which can be addressed is exploring how much information can realistically be exploited by any algorithm based on the properties of the data. This issue has started to be addressed in the complex networks community, for example Song et. al. explored the **limits of predictability** in human mobility to test the assumption on whether prediction is truly possible [101]. More recently Marting *et al* [77] have asked the same question for social systems in general highlighting that "*the central question of this paper — namely to what extent errors in prediction represent inadequate models and/or data versus intrinsic uncertainty in the underlying generative process — remains unanswered*". If a clear process was in place to test the limits of predictability in the underlying data, then circumstances where experiments demonstrate unrealistically high prediction rate can be

identified as having some flawed experiment design properties. For example if a recommender algorithm can accurately predict a users movie choice 99% of the time but the underlying data shows almost random behavior, then something is awry.

4.3.3 Specific Challenges

This section identifies a set of specific challenges that are central to advancing the goal of predicting system performance or to achieving the above priorities related to rigor and learning.

Auto-tuning. Adding to the challenge of recommender system selection and performance prediction is the difficulty of tuning the underlying algorithms. Nearest neighbor algorithms include a variety of parameters related to neighborhood size, weighting based of extent of commonality, and in the case of model-based approaches the size and truncation of the model. Similarly, latent factor models have extensive parameters in both training and use.

A consequence of the challenge of tuning is that researchers often fail to compare like with like. As parameters can depend on data distributions, it is increasingly important to identify standard ways to tune algorithms–and particularly baseline algorithms for comparison. Fortunately, tuning can be framed as a combination of parameter space exploration and understanding the response curves and sensitivity to parameters. With a systematic exploration of algorithms, we should also explore empirically-tested auto-tuning to ensure both fair comparisons and more efficient exploration.

Exploration vs. Exploitation. Two key challenges in recommender systems are discovering changes or inaccuracies in models of user preference and the cold-start item problem of learning to recommend new items. Both of these can be addressed interactively by presenting users with some set of "exploratory" recommendations – recommendations based not on their current tastes but on the system's need for information. For example, a music player may identify a target set of users to whom a new song should be played to identify the right audience for that song. Or a news recommender may periodically recommend a randomly selected news article to validate or update the user's preference model.

The trade-off between exploration and exploitation–both algorithmically and as a matter of user experience–needs further study. In particular, we may need to create metrics of "realistic user experience" that incorporate system-wide exploration as well as targeted exploitation of profiles.

Temporality and Dependency. Little work has been done in the recommender system community to address changes of user preferences over time, for example, Moore et al. [82] modeled temporal changes of preferences in music recommendation. We need to more systematically explore how we can detect patterns of change and exploit those in our performance prediction models in multiple domains across our use cases. Closely related to the challenge of temporal changes of user preferences is the ability to understand how recommendations based on user preferences influence and change preferences, i.e. we need to take this dependency into account in our performance models. For example, low level measures such as diversity or exploratory recommendations will have an impact on preferences.

Underlying theoretical assumptions in recommender algorithms. Recommender algorithms (and evaluation protocols) are built on statistical and mathematical models that incorporate underlying theoretical assumptions about the distributions and patterns of data. These range from the high-level assumptions of all collaborative filtering algorithms (stable

or predictable preferences, past agreement predicts future agreement) to more complex distributional assumptions (e.g., exponential popularity distributions to support neighbor-finding) to issues of temporal stability (e.g., whether offline evaluation has to be ordered vs, random). These theoretical limitations are often at most informally expressed, and they are rarely explicitly checked or analyzed. Rather, experiments are put in place, and empirical evidence is drawn from them confirming or refuting the effectiveness of recommendation approaches at the end of the algorithmic development pipeline.

An explicit and precise identification and a deeper understanding of the essential assumptions would help assess and document the scope of algorithmic performance evidence and predictions. In practice, recommender algorithms have often "worked well enough" when assumptions are violated, but such boundaries should be tested and understood.

We find it a worthy endeavor to research what the precise (core, simplifying or otherwise) assumptions in the algorithms really are; finding means for checking them in particular cases (data, tasks, users, etc.); and understanding the impact in the algorithm effectiveness to the extent that the assumptions are not met, or not fully. This should help enable and guide principled algorithmic development, diagnosis, deployment and innovation, beyond just assumption-blind trial and error.

Likewise, we should understand whether, to what extent or in what direction the biases may distort the experimental measurements. Further implicit assumptions are made on the purpose for which a recommender system is to be deployed when evaluation metrics are developed and chosen. Understanding and analyzing the consistency between metrics and the ultimate goals the system is conceived for are key to make sure the right thing is being measured.

Algorithm vs. System Performance. One of the major issues in evaluating the performance of a RecSys in a realistic setting, by real users, is the users' inability to distinguish between the system as a whole and the recommendation algorithm itself. Indeed, many of the most successful advances in studying algorithm differences have come from individual research systems where the same system interface could be used with different algorithms.

A challenge today, however, is general lack of access to such systems for the typical researcher. While industry has access to large user bases, companies rarely will allow external researchers to experiment with those users.

We therefore propose a community-wide effort to build and maintain a high-quality, usable recommender system specifically to support the research community. This system would have the ability to integrate different algorithms, and would include instrumentation to allocate users to experimental conditions, record user interaction, log system performance, and administer user experience surveys where needed. Most important, it would report metrics and export data according the to the community-agreed standard.

Such an effort could be launched *ab initio* or could involve creating a consortium to enhance, open, and maintain pre-existing recommender systems for the research community.

5 Cross-Discipline Themes

In order to predict performance, a number of research issues has to be addressed central to all domains (IR/NLP/RecSys), which are sketched in figure 1. First we have to choose the performance criteria and define corresponding *measures*. When performing experiments with different test collections and observing the system's output and the measured performance, we will carry out a *performance analysis*. For that, we will look at violations of the *assumptions*

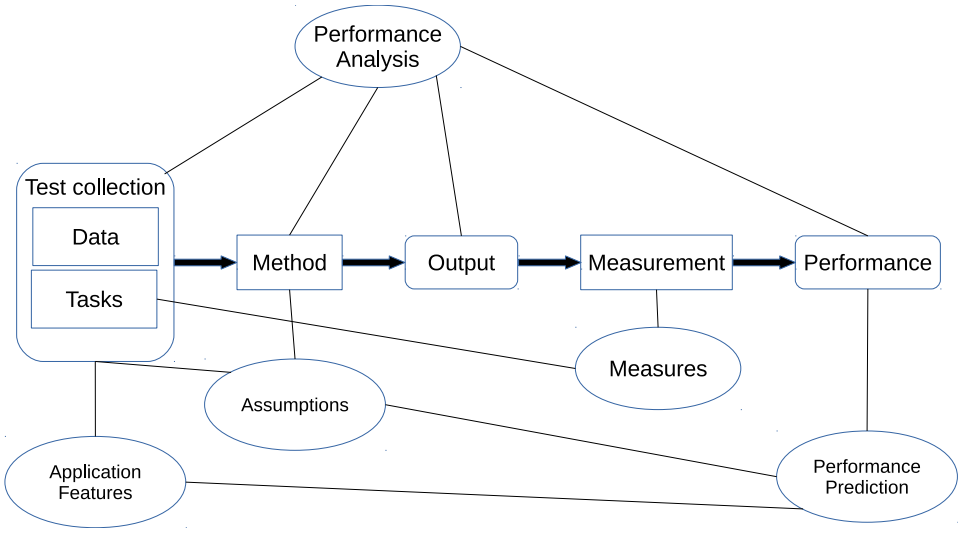

Figure 1 An overarching performance prediction framework.

underlying the method applied. Also, characteristics of the data and tasks will have an important effect on the outcome. Finally, we aim at developing a *performance prediction model* that takes these factors into account.

Different fields have traditionally focused on evaluating specific aspects in this framework, but we believe that understanding the relations between these tasks is essential in achieving adequate performance prediction. Moreover, we have mentioned several times the importance of reproducibility for improving our experimental evaluation practices. It should be understood that reproducibility is just another side of the coin when it comes to performance prediction. Indeed, the possibility of replicating the same results in the same experimental condition, the capability of reproducing them in different conditions, and the ability to generalize them to new tasks and scenarios are just another way of formulating the performance prediction problem.

Once these tasks are well understood we can begin to try and predict performance in an unseen situation if enough of the above still hold. Expecting to be able to test and tune all aspects of this pipeline is a limiting factor for exploring new ideas and solutions. It is our hope that by abstracting stages within the framework, recurrent patterns will emerge to support prediction for unseen cases (combinations of the above aspects).

In the following, we discuss each of these aspects in detail. Besides describing open research issues, we will also point to out some cases where weak current scholarly practices impair our understanding of the matter.

5.1 Measures

5.1.1 What and Why

In this section, we focus on two aspects, namely the definition of the low-level metrics, and the link between low-level and system-level performance evaluation, meant as the connection between more objective and engineering-like measures with more subjective ones, ultimately representing the user satisfaction and experience with a system.

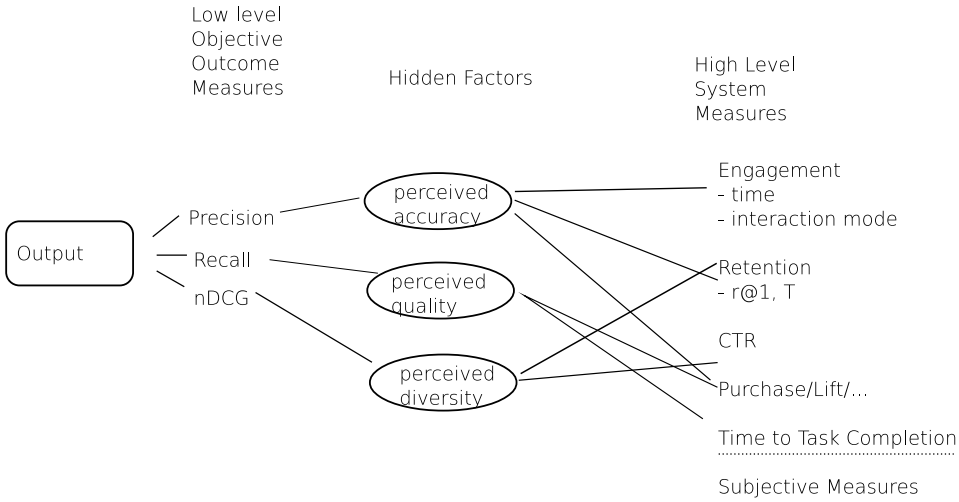

Figure 2 Low- and high-level performance measures.

Metrics definition

The definition of a metric relies on several alternatives and decisions, which happen before the actual measurement takes place, also to avoid any post-hoc bias.

We first have to choose the criteria that reflect the goals of our evaluation, for instance relevance, diversity, or novelty. However, as said before, the performance of a system is not only a matter of goals but also of the "utility" delivered to users. Therefore, we need to identify/choose a prototypical user behavior; for example, when ranking is involved, a stopping point, after which no more recommended items or retrieved documents are considered, introduces a clear separation between seen and unseen items, where only the former influence the measurement outcome. Instead of a deterministic behavior, we might also assume a stochastic model for this aspect as done, for example, by [21, 35, 81]. We also have to define the user's preferences concerning the items seen, like e.g. the total number of useful items, or the ratio between useful and useless items.

Finally, we have to choose an aggregation method like arithmetic or geometric mean, where the former focuses on absolute differences, and the latter on relative changes, paying attention that the aggregation method is admissible when considering the scale properties – ordinal, interval, ratio scales – of a measure [39, 50]

Overall, current research often neglects the fact that each metric represents a specific user standpoint, and often the standpoint may be context-dependent. Thus, an evaluation focusing on a single low-level metric will either ignore many user standpoints, or represent an intransparent mixture of different standpoints.

From low-level to system-level performance

Figure 2 shows a closer look at the measurement aspect, where we distinguish between low-level evaluation measures and expected high-level system outcomes. We see the range of inputs and performance measurements that reflect the performance of a diversity of systems, including IR, NLP, and Recommender Systems. In this section we focus on the link between low-level and system-level performance evaluation, and in turn on the challenge of not just building predictive models but also incorporating and building a deeper understanding of the factors that lead to system-level results.

This deeper understanding is essential to crafting complete systems. For example, to have more effective automated summarization, we need to understand what low-level properties of these summarizations lead users to perceive fluent text. Similarly, this enables us to understand how diversity of a search or recommender result affects the user's confidence in having found the correct result, or help them learn about the scope of possible results.

The model behind this understanding combines statistical analysis with existing theory to posit and evaluate hidden aspects (e.g., perceptions of coverage or fluency) and to measure the relationships among the low-level measurements, the hidden aspects, and the system-level performance measurements. A good model will highlight the most significant links, identify causation when possible, and provide mechanism to both predict the impact of system changes and reverse direction to identify target system changes to achieve desired system-level results.

5.1.2 How

5.1.2.1 Low level performance measures

Most of the research in algorithmic evaluation has focused on low level performance measures such as precision and recall. Our model tries to link low level measures to system level measures, potentially explained by the hidden aspects will tell us what low level measures can best identify system level successes (or failures). Some low level measures might directly relate to system level performances, for example, we might find that a measure of precision directly influences click through rates, but in many cases the relations between low and system level measures might be opaque and can only be understood by unraveling the underlying hidden aspects.

5.1.2.2 Hidden aspects

Hidden aspects are aspects that cannot directly be measured objectively, but that can be measured subjectively from users via surveys or observations. Hidden aspects allow us to understand relations between low level performance measures and system level measures. For example, a particular low-level measure (e.g. novelty) might relate to several hidden aspects in directionally different ways (e.g. improving the perceived diversity but reducing the perceived accuracy of a recommendation or search result). These hidden aspects in turn might either positively or negatively influence system-level performance measures. What hidden aspects to account for and how to measure these is a question for which theoretical understanding of the problem is crucial.

5.1.2.3 System-level performance measures

On the system side, many measures can reflect system performance. We must consider behavioral measures, which can be short-term effects such as click through rates, measures of what items users access or read and when they consume the items, as well as long-term effects reflecting the system's success such as long-term retention or users unsubscribing from a service or reduced or increased usage of the service. Subjective measures such as satisfaction that cover the user experience are also important to measure and predict system-level performance and can be both short-term (are you satisfied with the choice you just made) as well as long-term when measuring overall user satisfaction using infrequent higher-level surveys (e.g. are you happy with our service in the last three months). Sometimes subjective measures can even outperform behavioral measures in predicting, for example user segments of a website [55]. The challenge will be to understand which direct short-term measures best

predict long-term satisfaction and system success. Optimizing the system for any one single short-term measure may in fact harm long-term performance. For example maximizing the number of clicks from a user in a news reading service may actually not reflect they are reading more, it may mean they are trying to find something of interest to read and are failing, so a combination of number of clicks and pause time on the page in combination are better predictions of user satisfaction. Subjective measures such as surveys maybe used to provide training data for machine learning models to capture the complex relationship between system level measures and predicting good user retention and satisfaction.

5.1.2.4 Understanding the relations between the measures via path modeling

Our model explicitly models the hidden aspects as intermediate concepts relating the low-level aspects to the high-level measures. Statistical methods such as path modeling and structural equation modeling allow us to model structural relations between the variables in one single model. This approach in essence just regresses system-level measures on hidden aspects and low-level aspects, and shows whether effects of low level on system-level measures are direct, or indirect and thus mediated by the hidden aspects. As the model fits all relations at once, their relative contributions can be better understood and estimated then one could do by just correlating all measures without a clearly-defined underlying structure. Moreover, the path modeling will allow us to test which underlying structure provides the best explanation and find missing relations. For example, if hidden aspects relate to system level measures but no low level measures directly affect these particular hidden aspects, this will indicate that either better low level measures might be needed or that other external factors might influence our system level measures that are not under our control. Such external factors can be partially controlled for by testing the model in a controlled experiment in which we only manipulate a particular aspect of our algorithms and keep the rest of the system the same, such that we can tease out the one aspect we are interested in. Structural equation modeling moreover allows for hidden aspects to be latent constructs that can be measured through several questionnaire items, rather than by single indicators. This is important when measuring psychological constructs such as perceived diversity or satisfaction, because single items lack 'content validity'. Each user might interpret an item differently and by measuring concepts with several slightly differently phrased items a better measurement of the underlying latent construct can be achieved.

5.1.3 Next Steps

Here we suggest some key next steps that should be undertaken and supported in order to cover the full range from low-level and system-oriented measures to high-level and user-oriented ones:

- Creating a dictionary of higher-level and hidden-aspect measures, including validated and reusable measures that can support comparative research and accumulation of results across studies. We note that certain sub-areas have a longer tradition of higher-level evaluation metrics, and that other sub-areas will need to be engaged to understand the key success measures for their systems.
- Building a library of case studies–examples with constructed models. These cases should be collected in a standard format to promote further analysis and meta-analysis.
- Encourage the study of complex cases–including cases that span more than one technology. To build our understanding of how user perceptions affect performance of complex systems, we need to study a wide range of increasingly complex cases.

- Perform both manual and automated analyses to seek patterns in the case library. By exploring common subgraph patterns, we can develop evidence-based theory to govern system design. The most interesting patterns will likely be ones where the subgraphs depend on specific system attributes (e.g., certain relationships may exist in systems where users have a particular goal in mind, but not in ones where users are simply exploring).
- Ensure the availability of long-standing user cohorts, who can assess over time the systems and whose outcomes can be traced to validate predictions. We will require different user cohorts for different domains/contexts, so that it becomes possible to develop a matrix arrangements of systems across cohorts which can be leveraged for cross-predictability either column-wise, i.e. changing domain/cohort, or row-wise, i.e. changing systems within different rounds of the same cohort. An open question is how to map user cohorts to real users? How good is their external validity?

5.2 Performance Analysis

Reporting of averages and average improvements is often unhelpful, and is uninformative in terms of explaining what system elements contributed to success, what data and queries the method is applicable to, and for which data and queries the method fails. That is, instead of focusing on statistically significant differences in the average from control to treatment, we need to move to understanding the changes in specific tasks and task types, and to understanding the contributions of individual system components.

In this context, researchers also should no longer ignore the problems of multiple testing and sequential testing: When performing multiple significance tests on the same data set, they must adjust the significance level accordingly, using e.g. Bonferroni's method[8] [50]. Even more problematic is the sequential testing case, where the same data is used by other researchers, who have learned from previous results on the same test collection, and then perform significance tests for their new method(s), not considering the large number of tests carried out before. As shown in [20], this usually leads to totally random results. As a consequence, statistically meaningful results cannot be expected from heavily re-used test collections. A similar statement might also hold for multiple qualitative analyses on the same data set. Thus, re-use of a test collection is problematic, which leads to the need for more (and more diverse) collections.

Another important viewing angle is the consideration of measures representing different user standpoints: instead of focusing on universal performance, more emphasis should be put on performance differences wrt. different metrics. E.g., in retrieval, many users will look at the top ranking documents only (e.g. in Web search), while others are aiming at locating all potentially relevant documents (e.g. patent search). Thus, instead of looking at overall performance only, it is more interesting to identify methods that support specific user standpoints.

Classic failure analysis inspects individual tasks where performance is significantly altered, but other data interrogation methods, such as systematic addition of noise, can illustrate the robustness and vulnerabilities of the system that is under investigation.

[8] https://en.wikipedia.org/wiki/Multiple_comparisons_problem

5.3 Documenting and Understanding Assumptions

5.3.1 Role of assumptions

Any method or model is based on certain assumptions, some of which are explicit; usually, there is an even larger number of implicit assumptions. The performance of a method in an application depends mainly on the extent to which these assumptions are true in this setting. Thus, we have to solve three problems:

- Identify the underlying assumptions (make implicit ones explicit).
- Devise methods for determining if or to what extent these assumptions are fulfilled in an application.
- Develop a model that tells us how the violation of an assumption affects performance.

Only when we have answers to these three questions, we are able to make reasonable predictions.

5.3.2 Assumption categories

Assumptions come into play at many points in the design and evaluation of recommender, IR, and NLP systems. At each point, there are at least two broad categories of assumptions: *fundamental* assumptions and *convenience* assumptions. These two categories are transversal to different kinds of assumptions which we can distinguish according to the role they play in data, algorithms/techniques, evaluation protocols and metrics, and their implications on system performance and the validity of research findings. Overall, they determine a taxonomy which we can use to systematically check and make them explicit.

Convenience assumptions are simplifications (or approximations) intended to make problems tractable, reduce their complexity and/or enable evolving some starting point theoretical expression (e.g. a probability) into a computable form (counting things and doing math upon numbers). Examples include the mutual feature independence assumption in Naïve Bayes (of which pairwise word independence in text IR can be seen as a particular case), whereby joint probabilities are decomposed into products of simpler distributions; user, time and context independence as a means to eliminate variables from IR and recommendation problems; or document relevance independence assumption, which enables the definition of simple and easy to compute metrics such as precision. Convenience assumptions may be violated, and yet the algorithm or the metric may still work reasonably. On the other hand, performance differences between collections may be traced back to the violation of certain assumptions.

Convenience assumptions typically represent an opportunity to define new research problems consisting of the elimination of a particular simplifying assumption and dealing with the corresponding complexity. An example is personalized IR, which takes the user variable back into the problem and copes with it; or IR diversity, which removes the document relevance independence assumption; or time-aware or context-aware IR, which do the same with time and context.

By fundamental assumptions we mean hypotheses that algorithms or metrics themselves build upon – they are intrinsic to the underlying model. For instance, content-based recommendation assumes item features can partially explain user choices; IR language model algorithms assume language similarity is related to relevance; most text IR models assume term frequency matters; proximity search algorithms assume word order matters too; metrics like precision assume users want to get relevant documents; average search length (rank of first relevant document) assumes users need just one relevant document or item; recommendation diversity metrics may assume people enjoy variety; novelty metrics assume users wish to be surprised; an experimental protocol may assume each and every user has a non-empty set of

training (or test) observations. When fundamental assumptions fail to be met, the algorithm or the metric may no longer be effective or valid. Content-based recommendation is as good as random if user choices are unrelated to item features; a novelty metric is irrelevant if users are just willing to stick to familiar experiences; lack of data for a single user may result in an undefined evaluation outcome.

Becoming aware of and understanding fundamental assumptions enables a better and more consistent use of the tool (algorithm, metric, protocol) that builds upon them, and may prevent unintentional misuse. It can also help detect spurious confounders (biases that cause the hypothesis to hold for misleading reasons) and experimental flaws that can easily go unnoticed (e.g. a recommendation algorithm's accuracy skyrockets simply because we forgive it refusing to deliver recommendations to certain users; depending on the characteristics of these users, this may result in discriminatory quality of service).

5.3.3 Understanding Violations to Assumptions

A critical aspect in explaining and predicting performance is to understand whether and to what extent the assumptions our methods are based upon have been complied with or violated.

This understanding should happen at both theoretical and experimental level. At theoretical level, among the various assumptions, we should be able to differentiate those that are crucial for a method and whose violations seriously hamper its application from those that are somehow desirable. At experimental level, we should have techniques for assessing each assumption and understand whether and how much it has been violated.

We need to develop commonly agreed *scales* to quantify how much an assumption has been violated. However, given the wide range and diversity in the type of assumptions we have, we should aim at developing assumption checking methods and scales that hold, at best, for families of related assumptions rather than hoping for a single general solution where one-fits-all.

Then, we need to research on the relationship between the severity of departures from assumptions, quantified in the above mentioned scales, and the observed and predicted performances. The final goal is to understand how much resilient are our methods to such violations and how much this impacts on explanation, first, and prediction, after.

Violations of algorithm or technique assumptions are perhaps the easiest to assess: run the algorithm on a data set that violates its assumption(s) and measure its performance and behavior. Violations of evaluation and data assumptions are more challenging, as they undermine the tools by which we measure the behavior of the system in the first place. To assess the impact of these assumptions, we need techniques that allow us to peek behind the curtain and understand the behavior of these components of the experimental process under a range of possible truths, in order to relate their output to our confidence about the relationship of the data and evaluation to the underlying truth and intended task.

An area we can take inspiration from is statistics and the notion of *robustness* in statistical testing, meant as "insensitivity to small deviations from the assumptions" [64]. Robustness is developed both a theoretical level, e.g. by studying it under a null and an alternative hypothesis [68], and defining indicators such as, for example, the breakdown point, i.e. the proportion of incorrect observations an estimator can handle before giving an incorrect result.

Furthermore, simulation and resampling are particularly promising tools for quantifying the importance of assumptions to components of the information processing and evaluation pipeline. Measuring results on different data sets is useful, but only provides a few data points regarding the behavior of a method or evaluation technique, and does not change the

relevant variables in a controlled fashion; further, the data set's relationship to underlying ground truth cannot be controlled and may not be known. Simulation and resampling allows a range of possibilities – some within assumptions, some outside – to be tested, and the relationship of data to truth to be controlled, allowing us to precisely characterize the system response to targeted violations of its assumptions. These experiments can take multiple forms, including wholly-synthetic data, resampling of traditional data sets, and sampling of specialized data sets such as ratings collected on complete or uniformly sampled sets of items. As one example, [106] employed simulation to study the robustness of information retrieval evaluations to violations of statistical assumptions about the underlying data sets and their topic distributions.

5.3.4 Increasing awareness in our communities

We note that across our communities there is a large variance on how assumptions are managed and on the perception itself of their importance. A general recommendation is pushing in any possible way our communities to a greater awareness of the need for making assumptions explicit and clear. Inserting in all scientific works a clear statement permitting a precise identification and a deeper understanding of the essential assumptions made and their scope of validity should become a universal practice. To this regard we recall the effort currently conducted in the IR community toward reproducibility of results [40, 41]: after a consciousness campaign last several years, we now have a reproducibility track in one of the main IR conferences (ECIR), and reproducibility tasks have been just launched in the major evaluation campaigns[9].

The awareness on such an important aspect impacting the validity and reproducibility of results can be disseminated and increased in several ways. A first recommendation is adding an explicit reference to the clarity and completeness of assumptions made in the call for paper and the paper review forms of all conferences. This can have the double effect of educating the reviewers to reserve a particular attention to assumption clarity and, on the other hand, to increase author's awareness on them. Papers claiming results involving assumptions that are not explicitly voiced or understood should not be deemed as solid since no strong conclusion can be drawn from them. As a second step, after a systematization of assumptions and a greater understanding have been reached, the emerging best practices can give origin to commonly accepted requirements to be integrated in the call for papers of specific tracks.

It is significantly harder to test the importance of assumptions in user-facing aspects of the system, such as the presentation of results or the task model, as it is prohibitively expensive to simulate arbitrarily many versions of a system and put them before users. System utility can be remarkably robust to violations of core assumptions – for example, e-commerce vendors obtain great value from collaborative filtering techniques that assume items are functionally interchangeable even when they clearly are not – but rigorous empirical data on this robustness is difficult to obtain. However, measuring hidden factors (cf. Figure 2) might help explain why particular versions of a system perform better, directly testing underlying assumptions.

[9] http://www.centre-eval.org/

5.4 Application features

One common feature of Natural Language Processing, Information Retrieval and Recommender Systems is the wide space of data and task characteristics that have to be accounted for when designing a system. Adapting existing systems to a new domain, a new data set, or a new task, and then predicting their performance in this new setting is particularly challenging in our research fields because there is always some degree of mismatch between testing and development conditions (either in laboratory or real-world settings) used to create the existing systems and new application area.

As a result, measuring only effect sizes and statistical significance is of little help for predicting out-of-the-lab performance. Even moving between two test collections which apparently share the same features often results in different experimental outcomes. In order to have predictive power, evaluation methodologies need a much higher emphasis on explanatory analysis: why, where and how systems fail is more relevant than effect sizes on average measures.

In this section we begin by reviewing a few measurable characteristics that make prediction possible but challenging in our research fields, and we then move to advocating explanatory analysis.

5.4.1 Task & Data features

How will an existing method, algorithm or system perform under conditions different from the ones in which it was tested? There are some easily identifiable features related to the data or the task that, if changed, may affect predictability.

With respect to the task, some relevant characteristics are the **language** involved in the task. Will the task be performed using monolingual or multilingual data. Will the output be in a different language from the input (cross-linguistic)? Or is the task language independent? Are there the necessary language resources for the task? Does the task involve some dialect for which these resources have to be adapted? Is the data based on speech, on written text?

Another characteristic of the task is it **dynamicity**: are we dealing with a static collection, or a stream of data? Is the task a one-off, ad-hoc task, or a long standing task, such as filtering a news stream with a static query? Is the task offline, or online, performed with an active user? Does the task change over time as the user performs it?

Task **context** also plays an important role in many situations: Current Web search engines consider already user history, location, time and end device when computing the search result. The same might be true for other types of tasks.

We can characterize the data as **curated**, for example scientific papers, or edited news stories, or as naturalistic, for example, stemming from social media, or transcribed speech. In the latter case, one can sometimes measure the expected error rate, such as the frequency of spelling errors, or transcription errors. Many language processing tools were developed for curated language, without such errors.

Another dimension of data is its **connectedness** or **structure**. Can each data item be considered as a separate item, or are there links between items? For example, web pages link to other web pages. A collection of movies can contain a series of implicitly linked sequels. Users in a social network have both explicit and implicit connections to other users. Each data item can have internal structure (metadata such as timestamps, hand-assigned classification codes, numerical data; or internal structure, such as abstract, body, supplemental material).

With respect to (textual) data, some measurable features are: readability and comprehensibility; domain; users' expertise; how source and target data correlates; verifiability

of answers; dependence on assumptions to construct ground truth; richness of features; external validations; existence of corner cases and stress factors; parameterizations that impact performance; quality of domain resources (ontologies, dictionaries, taggers, etc.)

These characteristics, however, are not likely to be sufficiently predictive: even when they are the same in the new application as in laboratory conditions, often components of the systems perform differently. One of the main shortcomings of our experimental methodology is the lack of adequate explanatory methods.

5.4.2 Bias and Scaling

Test collections are often not a representative sample of a larger population. Instead, they have been compiled under certain restrictions (e.g. in IR test collections, rather specific or too general topics are not considered). We need to understand the limitations and bias of our sampling methodology across topics, documents, and systems. Can we determine when differences are due to bias, or when we are sampling from separate distributions?

Another problem to be investigated is the effect of scale: methods doing well on small test collections might not work on collections orders of magnitude larger, and vice versa.

5.5 Modeling Performance

Trying to explain and model the performance of systems over different datasets and tasks is a preliminary yet indispensable step towards envisioning how to predict the performance of such systems. However, this is often difficult to do due the lack of appropriate analysis techniques and the need for careful experimental designs and protocols, which may be complex and demanding to carry out.

There is therefore a need for further research providing us with the methods for analyzing and decomposing the performance into those of the affecting factors, such as system components, datasets, tasks, and more. These explanatory models will then constitute the basis for developing predictive models.

Performance prediction can take different forms. We commonly wish to make an *ordinal* prediction, of which of two systems will be superior for a kind of task over a class of collections. For a single system, we might aim at an *interval* prediction, giving us a confidence interval for a certain metric; the most simple case would be a prediction for another sample from the same population. While these two approaches target at average performance, we may alternatively wish to estimate risk or *uncertainty*, that is, predict a likelihood of failure.

5.5.1 Performance factor analysis in IR

In the case of IR, over the years, there have been examples of attempts to decompose performance into constituting factors, based on the use of General Linear Mixed Models (GLMM) and ANalysis Of VAriance (ANOVA).

[11, 102] have shown how to break down the performance of an IR system into a Topic and a System effect, finding that the former has a much bigger impact than the latter.

By using a specific experimental design, [45, 46] also broke down the System effect into those of its components – namely stop lists, stemmers, and IR models. They further demonstrated that we are not actually evaluating these components alone, even when we change only one of them and keep all the rest fixed. Rather, we are evaluating whole pipelines

where these components are inserted and with which they may have positive (or negative) interaction, boosting (or depressing) their estimated impact.

The difficulty in estimating the Topic*System interaction effects is the lack of replicates for each (topic, system) pair in a standard experimental setting. Therefore, [92] used simulation based on distributions of relevant and not relevant documents to demonstrate the importance of the Topic*System interaction effect. Very recently, [111] exploited random partitions of the document corpus to obtain more replicates of each (topic, system) pair, obtaining an estimation of the Topic*System interaction effect which allowed for improved precision in determining the System effect.

Finally, [44] conducted preliminary studies on the effect of Sub-Corpora and the System*Sub-Corpus, showing their impact and how they can be exploited to improve the estimation of the System effect.

All these GLMMs are not connected yet, meaning that they tackle the problem separately from different viewpoints but there is not yet a single model integrating all these facets. So a first required step toward performance prediction is to unify all these explanatory models into a single one. Then, the next step is to turn these models into predictive ones, e.g. by using some of the features discussed in Section 5.4 to learn how to predict the factors described in these models.

5.5.2 Controlled experimentation in NLP

NLP components are often combined into more complex NLP systems (e.g. part of speech detection, entity recognition etc. being used as part of a summarization task). The need to understand both the individual performance of components of pipeline NLP systems, as well as interactions between them, has resulted in evaluation methods involving controlled experimentation with systems in terms of these component parts.

Systematic component evaluation. Evaluating adaptivity by "decomposing" and evaluating it in a "piece-wise" manner can also be adapted from evaluations of interactive adaptive systems [87]. This can be done in a number of ways such as component substitution, ablation, and oracle input data.

Component Substitution: One strategy for doing this is to perform experiments that involve substitution of alternative components for a single component of the pipeline, to measure the impact on that component on the overall system performance.

Ablation: A related approach is the use of "ablation" (also called *lesion*) studies, in which sets of features, or combinations of feature sets, are systematically removed, in order to determine the most effective representation for a given task [24, 83]. This is commonly used in evaluation of machine learning-based methods which make use of substantial feature engineering.

Oracle input data: Pipelining components introduces a ceiling for each component that limits the performance of the overall system. To focus evaluation on a specific component in isolation, the performance of a target component can be measured by assuming that perfect input data is derived from earlier stages of processing. In user studies, this is sometimes also called a "wizard-of-oz" approach, where the component being evaluated is facing an end-user. Most commonly in these systems, the oracle is a human operating as system.

As an example, the performance of a relation extraction system that depends on a named entity recognition system as a precursor step will be limited by the performance of that earlier step. In the BioNLP-Shared Task relation extraction evaluations, gold standard entity

annotations are provided as input for the relation extraction systems [90] to control for this problem. While this represents idealized conditions for the overall system, it allows isolation of the relation extraction algorithms from performance effects resulting directly from the entity recognition step.

Test suites: Test suites have long been used by the NLP community to structure the evaluation of the functionality of specific tools [86], and also to structure efficient development of NLP systems [47]. In this approach, specific test cases are created based on controlled variation of pre-determined phenomena.

Recently, this approach has seen some revival, on the basis that articulation of specific cases identified in linguistic data can be used to guide finer-grained evaluation of systems that process that data, and that evaluation of purely natural data is dominated by high-frequency, possibly "simple" cases [23, 29, 57].

It can be argued that producing meaningful test cases is itself a challenging, resource-intensive activity [71], and also that the corner cases are not possible to define in advance. Nevertheless, this approach may provide a useful strategy to consider for deeper characterization of system performance and performance predictability, by characterizing the types of data that are expected to be seen by a system, and their varying distributions in natural data sets.

6 Conclusion

Performance prediction in the areas of IR, NLP and RecSys is a research problem that has been ignored for many years. In this manifesto, we have presented a framework for starting research in this area. Some problems might require substantial resources before they can be addressed. For instance, the analysis of performance-determining application features requires a large number of testbeds. Most of the problems, however, require primarily a more analytic approach. Instead of focusing only on performance improvement/system tuning, researchers should aim at improving our understanding of why, how and when the investigated methods work.

This manifesto should not only be regarded as a useful account of an important research challenge. We hope that it will also produce valuable fall-outs, such as bringing these issues in the research agenda of the involved communities (as it recently happened in the case of IR [3]), helping funding agencies in envisioning appropriate funding instruments for addressing these challenges, and spurring researchers on to overcome today's limitations.

7 Participants

- Pablo Castells
Autonomous University of Madrid, Spain
- Elizabeth M. Daly
IBM Research – Dublin, Ireland
- Thierry Declerck
DFKI GmbH – Saarbrücken, Germany
- Michael D. Ekstrand
Boise State University, USA
- Nicola Ferro
University of Padova, Italy
- Norbert Fuhr
Universität Duisburg-Essen, Germany
- Werner Geyer
IBM TJ Watson Research Center – Cambridge, USA
- Julio Gonzalo
UNED – Madrid, Spain
- Gregory Grefenstette
IHMC – Paris, France
- Joseph A. Konstan
University of Minnesota – Minneapolis, USA
- Tsvi Kuflik
Haifa University, Israel
- Krister Lindén
University of Helsinki, Finland
- Bernardo Magnini
Bruno Kessler Foundation – Trento, Italy
- Jian-Yun Nie
University of Montreal, Canada
- Raffaele Perego
CNR – Pisa, Italy
- Bracha Shapira
Ben Gurion University – Beer Sheva, Israel
- Ian Soboroff
NIST – Gaithersburg, USA
- Nava Tintarev
TU Delft, The Netherlands
- Karin Verspoor
The University of Melbourne, Australia
- Martijn Willemsen
TU Eindhoven, The Netherlands
- Justin Zobel
The University of Melbourne, Australia

Acknowledgements

We thank Schloss Dagstuhl for hosting us.

References

1 Gediminas Adomavicius, Jesse Bockstedt, Shawn Curley, and Jingjing Zhang. De-biasing user preference ratings in recommender systems. In *RecSys 2014 Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS 2014)*, pages 2–9, Foster City, CA, USA, 2014. URL: http://ceur-ws.org/Vol-1253/paper1.pdf.

2 Gediminas Adomavicius, Jesse C Bockstedt, Shawn P Curley, and Jingjing Zhang. Do recommender systems manipulate consumer preferences? A study of anchoring effects. *Information Systems Research*, 24(4):956–975, 2013. doi:10.1287/isre.2013.0497.

3 James Allan, Jaime Arguello, Leif Azzopardi, Peter Bailey, Tim Baldwin, Krisztian Balog, Hannah Bast, Nick Belkin, Klaus Berberich, Bodo Billerbeck, Jamie Callan, Rob Capra, Mark Carman, Ben Carterette, Charles L. A. Clarke, Kevyn Collins-Thompson, Nick Craswell, W. Bruce Croft, J. Shane Culpepper, Jeff Dalton, Gianluca Demartini, Fernado Diaz, Laura Dietz, Susan Dumais, Carsten Eickhoff, Nicola Ferro, Norbert Fuhr, Shlomo Geva, Claudia Hauff, David Hawking, Hideo Joho, Gareth Jones, Jaap Kamps, Noriko Kando, Diane Kelly, Jaewon Kim, Julia Kiseleva, Yiqun Liu, Xiaolu Lu, Stefano Mizzaro, Alistair Moffat, Jian-Yun Nie, Alexandra Olteanu, Iadh Ounis, Filip Radlinski, Maarten de Rijke, Mark Sanderson, Falk Scholer, Laurianne Sitbon, Mark Smucker, Ian Soboroff, Damiano Spina, Torsten Suel, James Thom, Paul Thomas, Andrew Trotman, Ellen Voorhees, Arjen P. de Vries, Emine Yilmaz, and Guido Zuccon. Research Frontiers in Information Retrieval – Report from the Third Strategic Workshop on Information Retrieval in Lorne (SWIRL 2018). *SIGIR Forum*, 52(1):34–90, June 2018. doi:10.1145/3274784.3274788.

4 Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. I like it... I like it not: Evaluating user ratings noise in recommender systems. In *17th International Conference on User Modeling, Adaptation, and Personalization, , UMAP 2009*, volume 5535 of *Lecture Notes in Computer Science*, pages 247–258. Springer, 2009. doi:10.1007/978-3-642-02247-0_24.

5 Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009*, pages 173–180. ACM, 2009. doi:10.1145/1639714.1639744.

6 Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009. doi:10.1007/s10791-008-9066-8.

7 Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. A general evaluation measure for document organization tasks. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 643–652. ACM, 2013. doi:10.1145/2484028.2484081.

8 Enrique Amigó, Damiano Spina, and Jorge Carrillo de Albornoz. An Axiomatic Analysis of Diversity Evaluation Metrics: Introducing the Rank-Biased Utility Metric. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *Proc. 41th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*, pages 625–634. ACM Press, New York, USA, 2018. doi:10.1145/3209978.3210024.

9 Marco Angelini, Nicola Ferro, Birger Larsen, Henning Müller, Giuseppe Santucci, Gianmaria Silvello, and Theodora Tsikrika. Measuring and Analyzing the Scholarly Impact of Experimental Evaluation Initiatives. In Maristella Agosti, Tiziana Catarci, and Floriana Esposito, editors, *Proc. 10th Italian Research Conference on Digital Libraries (IRCDL 2014)*, volume 38, pages 133–137. Procedia Computer Science, Vol. 38, 2014. doi:10.1016/j.procs.2014.10.022.

10 Apache Software Foundation. Apache Mahout 0.12.2, June 2016. URL: https://mahout.apache.org/.

11 David Banks, Paul Over, and Nien-Fan Zhang. Blind Men and Elephants: Six Approaches to TREC data. *Information Retrieval*, 1(1-2):7–34, May 1999. doi:10.1023/A:1009984519381.

12 Alejandro Bellogín, Pablo Castells, and Iván Cantador. Statistical Biases in Information Retrieval Metrics for Recommender Systems. *Information Retrieval*, 20(6):606–634, December 2017. URL: http://ir.ii.uam.es/pubs/irj2017.pdf.

13 James Bennett and Stan Lanning. The Netflix Prize. In *Proc. of KDD Work on Large-Scale Rec. Sys.*, 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.6998.

14 Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011. URL: http://ismir2011.ismir.net/papers/OS6-1.pdf.

15 Chris Buckley and Donna Harman. Reliable information access final workshop report. *ARDA Northeast Regional Research Center Technical Report*, 3, 2004.

16 Chris Buckley and Ellen M. Voorhees. Evaluating Evaluation Measure Stability. In Emmanuel J. Yannakoudakis, Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pages 33–40. ACM Press, New York, USA, 2000. doi:10.1145/345508.345543.

17 Chris Buckley and Ellen M. Voorhees. Retrieval Evaluation with Incomplete Information. In Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza, editors, *Proc. 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 25–32. ACM Press, New York, USA, 2004. doi:10.1145/1008992.1009000.

18 Jamie Callan and Alistair Moffat. Panel on Use of Proprietary Data. *SIGIR Forum*, 46(2):10–18, December 2012. doi:10.1145/2422256.2422258.

19 David Carmel and Elad Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Morgan & Claypool Publishers, USA, 2010. doi:10.2200/S00235ED1V01Y201004ICR015.

20 Ben Carterette. The Best Published Result is Random: Sequential Testing and Its Effect on Reported Effectiveness. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 747–750, New York, NY, USA, 2015. ACM. doi:10.1145/2766462.2767812.

21 Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected Reciprocal Rank for Graded Relevance. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009)*, pages 621–630. ACM Press, New York, USA, 2009. doi:10.1145/1645953.1646033.

22 Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. SVDFeature: A toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research: JMLR*, 13(1):3619–3622, December 2012. URL: http://dl.acm.org/citation.cfm?id=2503308.2503357.

23 K. Bretonnel Cohen, Christophe Roeder, William A. Baumgartner Jr., Lawrence E. Hunter, and Karin Verspoor. Test Suite Design for Biomedical Ontology Concept Recognition Systems. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2010/summaries/31.html.

24 Paul R. Cohen and Adele E. Howe. How evaluation guides AI research: The message still counts more than the medium. *AI magazine*, 9(4):35, 1988. URL: http://www.aaai.org/ojs/index.php/aimagazine/article/view/952.

25 Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection in 52 Languages. *CoRR*, abs/1706.09031, 2017. arXiv:1706.09031.

26 Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 39–46, New York, NY, USA, 2010. ACM. doi:10.1145/1864708.1864721.

27 Stephen Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2002. doi:10.1145/564376.564429.

28 Ronan Cummins. Document Score Distribution Models for Query Performance Inference and Prediction. *ACM Transactions on Information System (TOIS)*, 32(1):2:1–2:28, January 2014. doi:10.1145/2559170.

29 Dina Demner-Fushman. Adapting Naturally Occurring Test Suites for Evaluation of Clinical Question Answering. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 21–22, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W/W08/W08-0505.

30 Mukund Deshpande and George Karypis. Item-based top-N Recommendation Algorithms. *ACM Transactions on Information and System Security*, 22(1):143–177, January 2004. doi:10.1145/963770.963776.

31 Long Duong. *Natural language processing for resource-poor languages*. PhD thesis, The University of Melbourne, 2017. URL: http://hdl.handle.net/11343/192938.

32 Michael D. Ekstrand. *Towards Recommender Engineering: Tools and Experiments in Recommender Differences*. PhD thesis, University of Minnesota, Minneapolis, MN, July 2014. URL: http://hdl.handle.net/11299/165307.

33 Michael D. Ekstrand and Vaibhav Mahant. Sturgeon and the Cool Kids: Problems with Top-N Recommender Evaluation. In *Proceedings of the 30th Florida Artificial Intelligence Research Society Conference*. AAAI Press, May 2017. URL: https://aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/viewPaper/15534.

34 Hui Fang, Tao Tao, and Chengxiang Zhai. Diagnostic Evaluation of Information Retrieval Models. *ACM Trans. Inf. Syst.*, 29(2):7:1–7:42, April 2011. doi:10.1145/1961209.1961210.

35 Marco Ferrante, Nicola Ferro, and Maria Maistro. Injecting User Models and Time into Precision via Markov Chains. In Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin, editors, *Proc. 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2014)*, pages 597–606. ACM Press, New York, USA, 2014. doi:10.1145/2600428.2609637.

36 Marco Ferrante, Nicola Ferro, and Maria Maistro. Towards a Formal Framework for Utility-oriented Measurements of Retrieval Effectiveness. In James Allan, W. Bruce Croft, Arjen P. de Vries, and Chengxiang Zhai, editors, *Proc. 1st ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR 2015)*, pages 21–30. ACM Press, New York, USA, 2015. doi:10.1145/2808194.2809452.

37 Marco Ferrante, Nicola Ferro, and Silvia Pontarollo. An Interval-Like Scale Property for IR Evaluation Measures. In Nicola Ferro and Ian Soboroff, editors, *Proc. 8th International*

Workshop on Evaluating Information Access (EVIA 2017), pages 10–15. CEUR Workshop Proceedings (CEUR-WS.org), 2017. URL: http://ceur-ws.org/Vol-2008/paper_11.pdf.

38 Marco Ferrante, Nicola Ferro, and Silvia Pontarollo. Are IR Evaluation Measures on an Interval Scale? In Jaap Kamps, Evangelos Kanoulas, Maarten de Rijke, Hui Fang, and Emine Yilmaz, editors, *Proc. 3rd ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR 2017)*, pages 67–74. ACM Press, New York, USA, 2017. doi:10.1145/3121050.3121058.

39 Marco Ferrante, Nicola Ferro, and Silvia Pontarollo. A General Theory of IR Evaluation Measures. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2018. doi:10.1109/TKDE.2018.2840708.

40 Nicola Ferro. Reproducibility Challenges in Information Retrieval Evaluation. *ACM Journal of Data and Information Quality (JDIQ)*, 8(2):8:1–8:4, February 2017. doi:10.1145/3020206.

41 Nicola Ferro, Norbert Fuhr, Kalervo Järvelin, Noriko Kando, Matthias Lippold, and Justin Zobel. Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". *SIGIR Forum*, 50(1):68–82, June 2016. doi:10.1145/2964797.2964808.

42 Nicola Ferro and Diane Kelly. SIGIR Initiative to Implement ACM Artifact Review and Badging. *SIGIR Forum*, 52(1):4–10, June 2018. doi:10.1145/3274784.3274786.

43 Nicola Ferro, Maria Maistro, Tetsuya Sakai, and Ian Soboroff. Overview of CENTRE@CLEF 2018: a First Tale in the Systematic Reproducibility Realm. In Patrice Bellot, Chiraz Trabelsi, Josiane Mothe, Fionn Murtagh, Jian-Yun Nie, Laure Soulier, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Nineth International Conference of the CLEF Association (CLEF 2018)*, volume 11018 of *Lecture Notes in Computer Science*, pages 239–246. Springer, 2018. doi:10.1007/978-3-319-98932-7_23.

44 Nicola Ferro and Mark Sanderson. Sub-corpora Impact on System Effectiveness. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *Proc. 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 901–904. ACM Press, New York, USA, 2017. doi:10.1145/3077136.3080674.

45 Nicola Ferro and Gianmaria Silvello. A General Linear Mixed Models Approach to Study System Component Effects. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proc. 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*, pages 25–34. ACM Press, New York, USA, 2016. doi:10.1145/2911451.2911530.

46 Nicola Ferro and Gianmaria Silvello. Toward an Anatomy of IR System Component Performances. *Journal of the American Society for Information Science and Technology (JASIST)*, 69(2):187–200, February 2018. doi:10.1002/asi.23910.

47 Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000. doi:10.1017/S1351324900002370.

48 Juliana Freire, Norbert Fuhr, and Andreas Rauber. Report from Dagstuhl Seminar 16041: Reproducibility of Data-Oriented Experiments in e-Science. *Dagstuhl Reports*, 6(1):108–159, 2016. doi:10.4230/DagRep.6.1.108.

49 Norbert Fuhr. Salton Award Lecture Information Retrieval As Engineering Science. *SIGIR Forum*, 46(2):19–28, December 2012. doi:10.1145/2422256.2422259.

50 Norbert Fuhr. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum*, 51(3):32–41, December 2017. doi:10.1145/3190580.3190586.

51 Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the Fifth ACM Conference*

on Recommender Systems, RecSys '11, page 305–308, New York, NY, USA, 2011. ACM. doi:10.1145/2043932.2043989.

52 Alfonso Emilio Gerevini, Alberto Lavelli, Alessandro Maffi, Roberto Maroldi, Anne-Lyse Minard, Ivan Serina, and Guido Squassina. Automatic Classification of Radiological Reports for Clinical Care. In *Artificial Intelligence in Medicine - 16th Conference on Artificial Intelligence in Medicine, AIME 2017, Vienna, Austria, June 21-24, 2017, Proceedings*, pages 149–159, 2017. doi:10.1007/978-3-319-59758-4_16.

53 Werner Geyer, Casey Dugan, David R. Millen, Michael Muller, and Jill Freyne. Recommending Topics for Self-descriptions in Online User Profiles. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 59–66, New York, NY, USA, 2008. ACM. doi:10.1145/1454008.1454019.

54 Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information retrieval*, 4(2):133–151, July 2001. doi:10.1023/A:1011419012209.

55 Mark P. Graus, Martijn C. Willemsen, and Kevin Swelsen. Understanding Real-Life Website Adaptations by Investigating the Relations Between User Behavior and User Experience. In Francesco Ricci, Kalina Bontcheva, Owen Conlan, and Séamus Lawless, editors, *User Modeling, Adaptation and Personalization*, number 9146 in Lecture Notes in Computer Science, pages 350–356. Springer International Publishing, June 2015. doi:10.1007/978-3-319-20267-9_30.

56 Gregory Grefenstette. Exploring the Richness and Limitations of Web Sources for Comparable Corpus Research. In *Ninth Workshop on Building and Using Comparable Corpora*, page 26, Portoro, Slovenia, May 2016. ELDA. URL: https://comparable.limsi.fr/bucc2016/pdf/BUCC05.pdf.

57 Tudor Groza and Karin Verspoor. Automated Generation of Test Suites for Error Analysis of Concept Recognition Systems. In *Proceedings of the Australasian Language Technology Association Workshop 2014*, pages 23–31, Melbourne, Australia, 2014. URL: https://aclanthology.info/papers/U14-1004/u14-1004.

58 Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. LibRec: A java library for recommender systems. In *UMAP Workshops*, volume 1388 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL: http://ceur-ws.org/Vol-1388/demo_paper1.pdf.

59 Allan Hanbury, Henning Müller, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Ivan Eggel, Tim Gollub, Frank Hopfgartner, Jayashree Kalpathy-Cramer, Noriko Kando, Anastasia Krithara, Jimmy J. Lin, Simon Mercer, and Martin Potthast. Evaluation-as-a-Service: Overview and Outlook. *CoRR*, abs/1512.07454, December 2015. arXiv:1512.07454.

60 F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, December 2015. doi:10.1145/2827872.

61 Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. A Survey of Pre-Retrieval Query Performance Predictors. In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *Proc. 17th International Conference on Information and Knowledge Management (CIKM 2008)*, pages 1419–1420. ACM Press, New York, USA, 2008. doi:10.1145/1458082.1458311.

62 Ruining He and Julian McAuley. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee, April 2016. doi:10.1145/2872427.2883037.

63 Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004. doi:10.1145/963770.963772.

64 Peter J. Huber and Elvezio M. Ronchetti. *Robust Statistics*. John Wiley & Sons, USA, 2nd edition, 2009. URL: https://www.wiley.com/en-us/Robust+Statistics%2C+2nd+Edition-p-9781118210338.

65 Juhani Huovelin, Oskar Gross, Otto Solin, Krister Linden, Sami Petri Tapio Maisala, Tero Oittinen, Hannu Toivonen, Jyrki Niemi, and Miikka Silfverberg. Software newsroom–an approach to automation of news search and editing. *Journal of Print Media Technology research*, 2(3):141–156, 2013. URL: http://hdl.handle.net/10138/42754.

66 Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3):14, 2007. doi:10.1145/1247715.1247720.

67 Paul B. Kantor and Ellen M. Voorhees. Report on the TREC-5 Confusion Track. In *Proceedings of The Fifth Text REtrieval Conference, TREC 1996*, volume Special Publication 500-238. National Institute of Standards and Technology (NIST), 1996. URL: http://trec.nist.gov/pubs/trec5/papers/confusion_track.ps.gz.

68 Takeaki Kariya and Bimal K. Sinha. *Robustness of Statistical Tests*. Academic Press, USA, 1989. doi:10.1016/C2013-0-10934-8.

69 George Karypis. SUGGEST recommendation engine, November 2000. URL: http://glaros.dtc.umn.edu/gkhome/suggest/overview.

70 Benjamin Kille, Andreas Lommatzsch, Gebrekirstos G. Gebremeskel, Frank Hopfgartner, Martha Larson, Jonas Seiler, Davide Malagoli, András Serény, Torben Brodt, and Arjen P. de Vries. Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-Time Stream-Recommendation Algorithms. In Norbert Fuhr, Paulo Quaresma, Teresa Gonçalves, Birger Larsen, Krisztian Balog, Craig Macdonald, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventh International Conference of the CLEF Association (CLEF 2016)*, volume 9822 of *Lecture Notes in Computer Science*, pages 311–331. Springer, 2016. doi:10.1007/978-3-319-44564-9_27.

71 Margaret King. Evaluating Natural Language Processing Systems. *Commun. ACM*, 39(1):73–79, January 1996. doi:10.1145/234173.234208.

72 Bart Knijnenburg, Martijn Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, October 2012. doi:10.1007/s11257-011-9118-4.

73 Bart P. Knijnenburg and Martijn C. Willemsen. *Evaluating Recommender Systems with User Experiments*, pages 309–352. Springer US, Boston, MA, 2015. doi:10.1007/978-1-4899-7637-6_9.

74 Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proceedings of the 13th international conference on World Wide Web*, pages 393–402. ACM, 2004. doi:10.1145/988672.988726.

75 Jimmy J. Lin, Matt Crane, Andrew Trotman, Jamie Callan, Ishan Chattopadhyaya, John Foley, Grant Ingersoll, Craig MacDonald, and Sebastiano Vigna. Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge. In Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, editors, *Advances in Information Retrieval. Proc. 38th European Conference on IR Research (ECIR 2016)*, volume 9626 of *Lecture Notes in Computer Science*, pages 408–420. Springer, 2016. doi:10.1007/978-3-319-30671-1_30.

76 Andreas Lommatzsch, Benjamin Kille, Frank Hopfgartner, Martha Larson, Torben Brodt, Jonas Seiler, and Özlem Özgöbek. CLEF 2017 NewsREEL Overview: A Stream-Based

Recommender Task for Evaluation and Education. In Gareth J. F. Jones, Séamus Lawless, Julio Gonzalo, Liadh Kelly, Lorraine Goeuriot, Thomas Mandl, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Eighth International Conference of the CLEF Association (CLEF 2017)*, volume 10456 of *Lecture Notes in Computer Science*, pages 239–254. Springer, 2017. doi:10.1007/978-3-319-65813-1_23.

77 Travis Martin, Jake M. Hofman, Amit Sharma, Ashton Anderson, and Duncan J. Watts. Exploring limits to prediction in complex social systems. In *Proceedings of the 25th International Conference on World Wide Web*, pages 683–694. ACM, 2016. doi:10.1145/2872427.2883001.

78 Paul McJones. Eachmovie collaborative filtering data set. *DEC Systems Research Center*, 249:57, 1997.

79 Paul McNamee and James Mayfield. Comparing cross-language query expansion techniques by degrading translation resources. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 159–166. ACM, 2002. doi:10.1145/564376.564406.

80 Sean M. McNee, John Riedl, and Joseph A. Konstan. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, New York, NY, USA, 2006. ACM. doi:10.1145/1125451.1125659.

81 Alistair Moffat and Justin Zobel. Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):2:1–2:27, 2008. doi:10.1145/1416950.1416952.

82 Joshua L. Moore, Shuo Chen, Douglas Turnbull, and Thorsten Joachims. Taste Over Time: The Temporal Dynamics of User Preferences. In Alceu de Souza Britto Jr., Fabien Gouyon, and Simon Dixon, editors, *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013*, pages 401–406, 2013. URL: http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/220_Paper.pdf.

83 Allen Newell. A tutorial on speech understanding systems. *Speech recognition*, pages 3–54, 1975.

84 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May 2016. European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2016/summaries/348.html.

85 Chikashi Nobata, Nigel Collier, and Jun'ichi Tsujii. Comparison between tagged corpora for the named entity task. In *Proceedings of the workshop on Comparing corpora*, pages 20–27. Association for Computational Linguistics, 2000. doi:10.3115/1117729.1117733.

86 Stephan Oepen, Klaus Netter, and Judith Klein. TSNLP - test suites for natural language processing. In John Nerbonne, editor, *Linguistic Databases*, chapter 2, pages 13–36. CSLI Publications, 1998.

87 Alexandros Paramythis, Stephan Weibelzahl, and Judith Masthoff. Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction*, 20(5):383–453, 2010. doi:10.1007/s11257-010-9082-4.

88 Slav Petrov, Dipanjan Das, and Ryan McDonald. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*

(LREC-2012). European Language Resources Association (ELRA), 2012. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf.

89 Pearl Pu, Li Chen, and Rong Hu. A User-centric Evaluation Framework for Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM. doi:10.1145/2043932.2043962.

90 Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. *BMC Bioinformatics*, 13(11):S2, June 2012. doi:10.1186/1471-2105-13-S11-S2.

91 Fiana Raiber and Oren Kurland. Query-performance prediction: setting the expectations straight. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 13–22. ACM, 2014. doi:10.1145/2600428.2609581.

92 Stephen E. Robertson and Evangelos Kanoulas. On Per-topic Variance in IR Evaluation. In William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *Proc. 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pages 891–900. ACM Press, New York, USA, 2012. doi:10.1145/2348283.2348402.

93 Lior Rokach. Decomposition methodology for classification tasks: a meta decomposer framework. *Pattern Analysis and Applications*, 9(2):257–271, October 2006. doi:10.1007/s10044-006-0041-y.

94 Brent R. Rowe, Dallas W. Wood, Albert N. Link, and Diglio A. Simoni. *Economic Impact Assessment of NIST's Text REtrieval Conference (TREC) Program*. RTI Project Number 0211875, RTI International, USA, July 2010. URL: http://trec.nist.gov/pubs/2010.economic.impact.pdf.

95 Alan Said and Alejandro Bellogin. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In *Proceedings of the Eighth ACM Conference on Recommender Systems (RecSys '14)*, RecSys '14, page 129–136, New York, NY, USA, October 2014. ACM Press. doi:10.1145/2645710.2645746.

96 Tetsuya Sakai. Evaluating Evaluation Metrics based on the Bootstrap. In Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin, editors, *Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 525–532. ACM Press, New York, USA, 2006. doi:10.1145/1148170.1148261.

97 Tetsuya Sakai. Topic set size design. *Information Retrieval*, 19(3):256–283, June 2016. doi:10.1007/s10791-015-9273-z.

98 Mark Sanderson and Justin Zobel. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *Proc. 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 162–169. ACM Press, New York, USA, 2005. doi:10.1145/1076034.1076064.

99 Saúl Vargas Sandoval. *Novelty and diversity evaluation and enhancement in recommender systems*. PhD thesis, Universidad Autonama Demadrid, Madrid, Spain, 2015. URL: http://saulvargas.es/phd-thesis.pdf.

100 Fabrizio Sebastiani. An Axiomatically Derived Measure for the Evaluation of Classification Algorithms. In James Allan, W. Bruce Croft, Arjen P. de Vries, and Chengxiang Zhai, editors, *Proc. 1st ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR 2015)*, pages 11–20. ACM Press, New York, USA, 2015. doi:10.1145/2808194.2809449.

101 Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010. doi:10.1126/science.1177170.

102 Jean Tague-Sutcliffe and James Blustein. A Statistical Analysis of the TREC-3 Data. In Donna K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 385–398. National Institute of Standards and Technology (NIST), Special Publication 500-225, Washington, USA, 1994.

103 Clare Thornley, Andrea C. Johnson, Alan F. Smeaton, and Hyowon Lee. The Scholarly Impact of TRECVid (2003–2009). *Journal of the American Society for Information Science and Technology (JASIST)*, 62(4):613–627, April 2011. doi:10.1002/asi.21494.

104 Theodora Tsikrika, Alba Garcia Seco de Herrera, and Henning Müller. Assessing the Scholarly Impact of ImageCLEF. In Pamela Forner, Julio Gonzalo, Jaana Kekäläinen, Mounia Lalmas, and Maarten de Rijke, editors, *Multilingual and Multimodal Information Access Evaluation. Proceedings of the Second International Conference of the Cross-Language Evaluation Forum (CLEF 2011)*, volume 6941 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2011. doi:10.1007/978-3-642-23708-9_12.

105 Theodora Tsikrika, Birger Larsen, Henning Müller, Stefan Endrullis, and Erhard Rahm. The Scholarly Impact of CLEF (2000–2009). In Pamela Forner, Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. Proceedings of the Fourth International Conference of the CLEF Initiative (CLEF 2013)*, volume 8138 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013. doi:10.1007/978-3-642-40802-1_1.

106 Julián Urbano. Test collection reliability: a study of bias and robustness to statistical assumptions via stochastic simulation. *Information Retrieval Journal*, 19(3):313–350, December 2015. doi:10.1007/s10791-015-9274-y.

107 Karin Verspoor, Kevin Bretonnel Cohen, Arrick Lanfranchi, Colin Warner, Helen L Johnson, Christophe Roeder, Jinho D Choi, Christopher Funk, Yuriy Malenkiy, Miriam Eckert, et al. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC bioinformatics*, 13(1):1, 2012.

108 Jesse Vig, Shilad Sen, and John Riedl. Computing the Tag Genome. Technical report, GroupLens Research, University of Minnesota, 2010. URL: http://www.grouplens.org/node/447.

109 Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management*, 36(5):697–716, September 2000. doi:10.1016/S0306-4573(00)00010-8.

110 Ellen M. Voorhees, Shahzad Rajput, and Ian Soboroff. Promoting Repeatability Through Open Runs. In Emine Yilmaz and Charles L. A. Clarke, editors, *Proc. 7th International Workshop on Evaluating Information Access (EVIA 2016)*, pages 17–20. National Institute of Informatics, Tokyo, Japan, 2016. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings12/pdf/evia/04-EVIA2016-VoorheesE.pdf.

111 Ellen M. Voorhees, Daniel Samarov, and Ian Soboroff. Using Replicates in Information Retrieval Evaluation. *ACM Transactions on Information Systems (TOIS)*, 36(2):12:1–12:21, September 2017. doi:10.1145/3086701.

112 Martijn C. Willemsen, Mark P. Graus, and Bart P. Knijnenburg. Understanding the role of latent feature diversification on choice difficulty and satisfaction. *User Modeling and User-Adapted Interaction*, 26(4):347–389, October 2016. doi:10.1007/s11257-016-9178-6.

113 Bo Xiao and Izak Benbasat. E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Quarterly*, 31(1):137–209, 2007. URL: http://www.jstor.org/stable/25148784.

114 David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310, 2002. doi:10.1017/S135132490200298X.

115 Yelp. Yelp Dataset, September 2017. Accessed: 2017-11-2. URL: https://www.yelp.com/dataset/challenge.

116 Cai-Nicolas Ziegler, Sean McNee, Joseph A Konstan, and Georg Lausen. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web*, pages 22–32, Chiba, Japan, 2005. ACM. doi:10.1145/1060745.1060754.

117 Justin Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proc. 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 307–314. ACM Press, New York, USA, 1998. doi:10.1145/290941.291014.

118 Justin Zobel, William Webber, Mark Sanderson, and Alistair Moffat. Principles for Robust Evaluation Infrastructure. In Maristella Agosti, Nicola Ferro, and Costantino Thanos, editors, *Proc. Workshop on Data infrastructurEs for Supporting Information Retrieval Evaluation (DESIRE 2011)*, pages 3–6. ACM Press, New York, USA, 2011. doi:10.1145/2064227.2064247.

119 Ludovik Çoba and Markus Zanker. rrecsys: An R-package for Prototyping Recommendation Algorithms. In *RecSys Posters*, volume 1688 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1688/paper-12.pdf.

120 Ludovik Çoba and Markus Zanker. Replication and Reproduction in Recommender Systems Research - Evidence from a Case-Study with the rrecsys Library. In *Advances in Artificial Intelligence: From Theory to Practice*, Lecture Notes in Computer Science, pages 305–314. Springer, Cham, June 2017. doi:10.1007/978-3-319-60042-0_36.