Salton Award Lecture
Information Retrieval as Engineering Science

Norbert Fuhr
Faculty of Engineering Sciences
University of Duisburg-Essen
Duisburg, Germany
norbert.fuhr@uni-due.de

Abstract

This paper gives my personal view on the field of information retrieval (IR), by first presenting my own definition of the field, and then pointing out why IR isn't an engineering science yet. For the latter, IR lacks appropriate theoretic foundations, which would not only give us a better understanding of current systems, but also would provide a generally valid basis for building systems, and enable us to make predictons about the performance of these systems. For reaching this goal, we need a well-balanced interplay between theory building, hypothesis generation and testing via experimentation.

1 Introduction

First, let me say how pleased I am to receive the Gerard Salton Award from SIGIR. Salton's pioneering work strongly influenced our field for several decades. I had lively discussions with him at several occasions, especially when he visited our research group in Darmstadt for several weeks in 1988. We were following different theoretic approaches—he as inventor of the vector space model, myself as a young researcher being enthusiastic about probabilistic models—but the discussions with him improved my understanding of the commonalities and differences of both approaches, such as the role of representations, underlying assumptions and theoretic justifications.

Like my predecessors, I want to give my personal view on information retrieval, by first presenting my own definition of the field, and then pointing out some areas for future research.

2 My definition of IR

For me, IR deals with *vagueness and imprecision in information systems*. Vagueness means that users are not able to give a precise specification of the objects s/he is looking for, like e.g. "How do I get to Portland?" or "I am looking for a high-end Android smartphone at a reasonable price." Typically, vagueness leads to an iterative retrieval process. Vice versa, even for highly structured data (like e.g. in a relational database), if we frequently observe

iterative querying, then this calls for IR methods that are able to cope with vagueness. For example, comparison shopping web sites usually sort offers by ascending prices, but often also show customer's rating of the corresponding shops; thus, many users may look for a compromise between cheap offer and satisfying service - so both the conditions 'best price' and ' high customer satisfaction' are vague criteria, where each user might have a different interpretations of these two conditions, as well as how to find the optimum balance between them [9].

Imprecision is mainly caused by the imperfection in the representation of the semantics and pragmatics of the objects stored, which are typically (multimedia) documents. Another cause might be imprecise or missing attribute values (e. g., in a product database listing new and upcoming smartphones, prices might be give as ranges, and the release date might be more or less precise). Traditionally, IR has been very strong in handling imprecision of the first kind, by developing appropriate weighting schemes. However, these schemes were restricted to propositional forms of representation. For dealing with multimedia data, spatial or temporal relationships also have to be considered. This calls for extending the weighting schemes to relational representations, which, in turn, are also able to cope with imprecise attributes.

Overall, we see that IR is not restricted to unstructured (or semi-structured) media (which used to be the traditional definition of our field), and the classic distinction between database (DB) and IR systems is no longer valid. In fact, in a way, I view IR as a generalization of the former: The logical DB view interprets query processing as the task of finding those objects o for a query q for which the implication $o \rightarrow q$ holds. Rijsbergen defined IR as being based on uncertain inference [17], i.e. determining the probability $P(d \rightarrow q)$ that document d implies the query. Thus, from a querying point of view, classic DBs can be regarded as a special case of IR.[1] However, in recent years, probabilistic databases [10] also have become a popular research topic in the DB field [16].

I think that the major difference between DB and IR lies in the consideration of the pragmatic level: In the standard architecture of DB application, there is a clear separation between the application system and the database management system, where all pragmatic aspects are delegated to the former; thus, pragmatics is not a core issue in DB research. In contrast, pragmatics plays a prominent role in IT research, which is reflected e.g. in the notion(s) of relevance [5] or in user-oriented evaluations focusing on efficiency and effectiveness. In terms of architecture, our field does not have a DB-like separation between application and core IR system, although such a separation might be useful, e.g. for accessing the same IR system from different end user devices or task contexts.

There are also some aspects where IR still could learn from the DB field:

Multiple steps of inference: In DB terminology, inference steps correspond to the well-known joins, which allow for combining different sources of knowledge (e.g. relational tables). In contrast, most IR applications are restricted to one or two inference steps, which are usually hard-coded in the IR system. Supporting more flexible inference schemes would not only ease the inclusion of knowledge from internal or external sources (like, e.g. ontologies — the numerous IR applications using Wikipedia can be regarded as instances of this approach), it also would become possible to combine the knowledge of different documents (like e.g. d_1: "cigarette smoke contains tar", d_2: "tar causes

[1] As a historic anecdote, Codd's seminal article on relational databases [7] appeared in the section entitled 'information retrieval' — the separation between IR and DB developed only later.

cancer" → "smoking causes cancer").

Expressive query language: Just like SQL, an expressive IR query language would allow for specifying 1) the inference scheme to be used for answering the query, and 2) the document parts or aggregates to be retrieved. In most IR systems, both aspects are hard-coded. thus making it difficult to tailor them for specific applications. In analogy to the DB field, only a minority of end users will formulate their queries using such a language, but the application programmers would benefit from such a descriptive language. One might argue that XQuery-FT[2] already provides the desired features, but its uncertain inference mechanism is restricted to the text search part of the language, thus prohibiting e.g. uncertain joins or aggregates.

Data types and vague predicates: Most IR research focuses on text for which linguistic methods like term normalization or stemming are usually applied. However, both within the text as well as in specific fields, there are character strings where these methods are inappropriate, like e.g. names of persons, institutions or products, times, dates, locations, amounts, technical measurements. For dealing with these entities, the DB notion of data types (and the corresponding type system) should be adopted, along with data type-specific comparison predicates; for IR applications, however, IR needs *vague* predicates [9], for supporting query conditions such as "about a month ago", "I want to pay no more than 200 USD", "in the New York area", "at room temperature", "similar to this photo".

3 IR as an engineering discipline

Many people view IR as an engineering discipline. The title of our major conference "Research and *Development* in Information Retrieval" points to that aspect, and a large fraction of IR researchers (including myself) feel themselves as IR "engineers", i.e. they apply and extend known methods for solving new problems. However, imagine an IR researcher applying his methods in civil engineering: being asked to build a bridge across a river, he would propose to build several bridges of different types, and then wait and see which of the bridges is still standing after a year or so. As funny as this analogy might be — the major Web search engines are tuned this way, by massively collecting query logs and click-through data.

In contrast, a civil engineer planning such a bridge, a mechanical engineer constructing a new machine or an electrical engineer designing a new device—they all build upon a rich portfolio of basic findings and theoretic models, which not only allow them to solve the problem, but also make *predictions* about the result of their efforts (as well as knowing the limits of their methods). In analogy, assume an IR "engineer" being confronted with the task of designing a system for a new collection of documents and a new type of information needs—would she be able to guarantee a certain MAP value or an average search task completion time? Rather not — we are neither able to make reliable predictions, nor do we know the limitations of our methods. As a further consequence, we also do not know how to improve an existing system in order to reach a predefined performance level: e.g., a physician typically needs 30 minutes to perform a literature search for a problematic case – what would it take to reduce this time to 10 minutes?

[2] `http://www.w3.org/TR/xquery-full-text/`

Institutions and companies have large varieties of collections and a broad range of search tasks. The "try and error" method outlined above is not applicable in most of these settings. So we might wonder what is required for constructing an IR system like in other engineering disciplines? I think we need two things that allow us to build upon, namely 1) theoretic models, and 2) solid empirical evidence.

On the theory side, the basis is rather small. E.g., for the case of ad-hoc retrieval, we have the probability ranking principle [14], various models based on it (like the classical probabilistic models, the principle of uncertain inference [17], and language models / divergence from randomness [13, 3]), and the term weighting axioms [8]. These results tell us how to achieve good or even optimum retrieval performance, provided that the underlying assumptions hold. However, we hardly have theories that tell us why certain methods work and others don't. Furthermore, these results do not yet allow us to make any predictions. So there seems to be little benefit from IR theory so far.

This raises the more general question about the value of theoretic IR models. In fact, there are three good reasons why we need them:

1. Theories give us a deeper *insight* into the foundations of our field, thus satisfying the scientific interest.
2. Theoretic models possess general *validity*, thus forming the basis for broad ranges of applications — in contrast to experiments where we just don't know to what extent their results can be generalized.
3. Only theories allow us to make reliable *predictions* — which is important from the engineer's point of view.

The first justification seems to be the dominating one in our field, and thus we 'ban' theory papers to specific conferences on IR theory. I have the impression that many IR researchers think very low of theory, since it seems too far away from the applications they are dealing with. Sometimes when I give a talk in the IR community, and ask the audience who of them knows about the Probability Ranking Principle, only a small fraction of the audience raises their hands.. This looks to me like an electrical engineer who ignores Ohm's law, since he can measure everything he needs. This analogy also illustrates the second and third justification: Ohm's law is generally valid, and it allows the engineer to design electric circuits and make predictions about their behavior. In the same way, IR theory has the potential to form the foundation for IR engineering.

The three claims from above hold in a well-defined application range: this range is implicitly defined by underlying assumptions. These assumptions usually can be verified, by testing whether or not they hold for the specific application under consideration. A traditional engineer is familiar with the underlying assumptions of his methods, and either knows from the application specification that they hold (e.g. electronic devices work reliably only in a certain temperature range), or he performs some tests for verifying them (e.g. a civil engineer would perform a soil examination).

4 The roles of experimentation in IR

These considerations lead us to the role of experimentation. Here I would like to distinguish between two kinds of experiments, namely why- vs. how-experiments:

- *Why*-experiments are based on a solid theoretical model, and they are performed for validating the model assumptions.
- *How*-experiments, in contrast, are based on some ad-hoc model, and they focus on optimizing some output parameter (usually retrieval quality), not bothering much with the underlying assumptions.

4.1 How-experimentation

An interesting insight into this type of experimentation is given by the meta-study [4], which looked at papers proposing specific methods for improving the quality of ad-hoc retrieval, based on some of the official TREC and CLEF collections. Although all of the considered papers claimed some performance improvements over previous work, hardly any of them was actually able to beat the best official run for the corresponding collection—the authors had been cheating by using poor baselines for comparison. Armstrong et al.'s major finding was summarized already in the title "improvements that don't add up"—the methods proposed in the various papers just describe alternative methods for achieving the same performance level, but nothing beyond.

As an extreme form of how-experimentation, we have recently seen a significant increase of experimental results using proprietary data. Thus, these results cannot be validated by others—which is a violation of basic scientific standards. Besides the possibility of cheating (intentionally or unintentionally) without ever being noticed, there is the more important issue that follow up research by others is not possible here: most research directions in our field starts with some innovative idea, which is then followed by many other researchers aiming at improving over the initial results. With proprietary data, this is no longer possible; a conference that mainly presents research of this kind could no longer be called scientific, it would be more an assembly of people telling anecdotes.

4.2 Why-experimentation

This type of experiments is based on a solid theoretical model, and its main purpose is the validation of the model's assumptions. As a simple example, consider the well-known binary independence retrieval model. Its core assumption is that terms are distributed independently in relevant and irrelevant documents. However, I don't know any study which investigated to which extent this assumption holds for a specific test collection. Theoretically, the quality of this model depends on the extent to which this assumption is fulfilled by a collection.

As another example, in [1], we contrasted $tf{\cdot}idf$ parameters of terms with their probability of relevance, illustrating that the standard $tf{\cdot}idf$ weighting scheme and the corresponding term weighting axioms are valid here, but that there are substantial differences between the two test collections considered. The paper [18] follows a similar approach by regarding the perplexities of (word) n-grams in different parts of Web documents and queries for developing appropriate language models.

In fact, this kind of study is rather rare in IR. Presumably, most researchers look at their data only in an informal way, before they come up with some model, and then they only regard its performance. From a scientific point of view, however, we have to "look under the hood", in order to get a better understanding of our field. Moreover, transparent models allow for verifying each assumption used, along with the estimation of the model parameters

involved. Focusing on these issue will also help in identifying weaknesses of current models and formulating better ones.

4.3 Experimentation vs. theory

The general limitations of empirical approaches in computer science have been discussed in [11]. Genova pointed out that there are two kinds of empirical approaches in science, namely verificationism and falsificationism. The former is the optimistic approach, assuming that induction is possible. However, unlike the standard proof method of complete induction known from mathematics, induction in empirics is based on a limited number of observation, and there is no proof for generalization. A popular example of this kind of thinking can be found in sports: "Team A never lost against team B when playing at home — so they also won't lose against B next weekend". Falsificationism is the pessimistic approach being propagated by Karl Popper; from this point of view, experiments can only be used for falsifying conjectures, never for proving anything. From these considerations, it should be clear that purely empirical approaches will not lead to scientific progress — we need theoretic models.

Contrasting experimental and theoretical approaches in our field, the former focus on answering how-questions, they target at good experimental results (for the collections considered), the outcomes have the potential for some further improvements (in limited settings), but the findings are mostly short-lived. Theoretical approaches, on the other hand, deal with the why–questions, they have a high explanatory power and form the basis for a broad variety of approaches; their results are usually long-standing.

Genova' final conclusion sounds a bit contradictory (which it isn't, since it refers to two different levels of science): *There is empirical evidence that the most important contributions to our field are not the experimental ones.*

In the field of IR, we have seen a vast amount of papers presenting numerous variants of existing models and methods in recent years, along with experimental results. This is partly a consequence of favoring experimental over theoretic work in our field. However, in the light of our general theme of IR as engineering science, this development was not helpful. I want to stress this point also with regard to the three claims in favor of IR theory:

Insight/Explanation We are not able to give fully satisfying explanations of the outcomes of the experiments performed. Since we are mainly focusing on retrieval quality, too little attention is paid to the different factors affecting the results, like e. g. the representation used, the various assumptions underlying the retrieval model, the method used for estimating the model parameters, or the specific properties of the test collections used; often, good or bad performance is attributed to 'the model', without investigating its various components or the influence of the other factors (or their combination).

Validity We do not know to what extent these findings can be generalized to other settings: For a single collection, it is impossible to tell how far we can generalize from this single point of observation. Even with multiple collections, given that each one represents a point in a high-dimensional parameter space (and we even don't know what the crucial parameters are), we can hardly define the area where interpolation is reasonable—plus the problem that these are stochastic experiments anyway, and we would need more observations to get a statistically significant result.

Predictions Experimental results alone give us no clue on how to make predictions for new situations (document collections, tasks, relevance criteria), since issues like e.g. scalability or comparability require appropriate theoretic foundations.

5 Systematic evidence

In order to learn more from experimentation and to be able to generate hypotheses, we need more systematic experimentation. The goal is to work towards evidence-based IR. For that, we need to investigate a large variety of test collections and consider a large number of controlled variables. In order to make results from different studies comparable, we also need standardized evaluation procedures.

5.1 Experimental variables

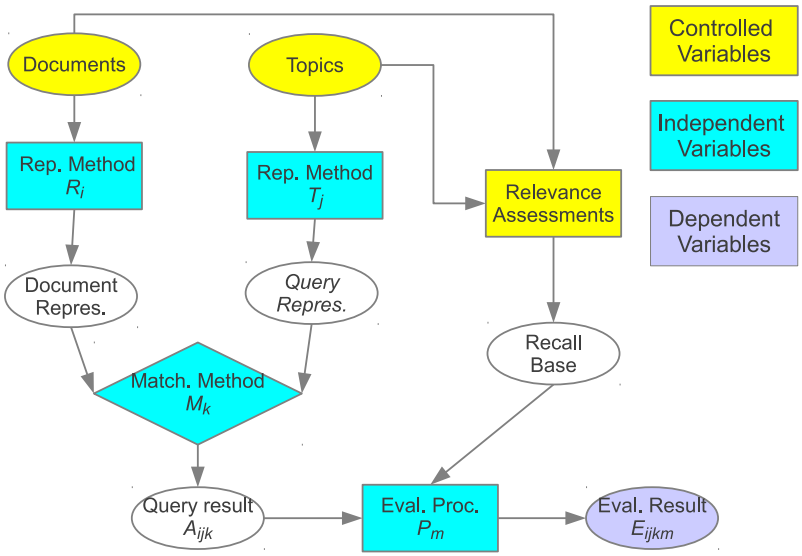

Figure 1: Experimental variables in ad-hoc retrieval

Figure 1 illustrates the standard experimental setting for ad-hoc retrieval: we focus on the representation methods for documents and queries, the matching method and the evaluation procedure as independent variables, and regard their effect on the dependent variable, the evaluation result. Since experiments are performed for a few test collections, they represent different instances of the three controlled variables documents, topics and relevance assessments.

From an engineering point of view, one would be most interested in the effect of the controlled variables on the outcome, since different applications vary with respect to these variables. So we might wonder how we can characterize these variables. For documents, potentially relevant factors include language, length, collection size, vocabulary size, domain, genre and structure, whereas topics can be characterized e.g. by length, linguistic structure, application domain, user expertise, task complexity; in a similar way, relevance can also be characterized by several attributes [5]. However, it is obvious that these lists are not

complete, and we would like to know which other variables are also important. Overall, these aspects span a high-dimensional parameter space, in which each test collection represents a single observation point, for which we have some experimental results. Each new application represents another point in this space, and we would like to make some predictions about the behavior of specific IR methods at this point. Given that the number of public test collections is not much larger than the number of dimensions of the parameter space, we see that we are far from IR engineering in the sense of being able to make predictions. In order to draw any useful conclusions from experimental results, we need standardized test environments and test procedures for being able to aggregate knowledge from larger numbers of experimental studies. This experience calls for more standardized experimentation, so that experimental results can be compared immediately.

For getting closer to this point, we have to collect results from standardized experiments. The site `evaluatir.org` aims at this task, by providing the possibility to submit experimental results and allow for comparison with other results for the same document collection [4]; however, besides the creators of the site, hardly anyone else has submitted any results so far. We need to build a massive collection of experimental results of this kind for populating the parameter space mentioned above. Only with this data, we are able to form hypotheses about the influence of the various test collection parameters on retrieval performance, which in turn can be the starting points for building appropriate theoretic models — which will finally allow us to make predictons about new collections.

5.2 Grand IR theory vs. empirical science

Just like Stephen Robertson in his Salton Award keynote in 2000 [15], I finally come to the question whether we will ever get to a grand IR theory (forming the very basis of IR engineering). I think that this can be only a long-term goal — which we might never reach completely. However, we should aim at strengthening the theoretic foundations of our field. For quite some time, however, empirical evidence will play a major role in our field. So I see two major research directions:

- On the theoretical side, we have to work on proofs that replace the empirical and heuristic approaches. Experiments are used mainly for validating the underlying assumptions of models.
- At the empirical side, we should aim at collecting broad empirical evidence. This is only useful if we strictly control experimental conditions, and also repeat experiments with other collections/tasks. Our focus here should be on the variables affecting performance, for which more meta-studies have to be performed.

6 Conclusion and Outlook

Empirical research is necessary, but it must be accompanied by strong theoretic models. IR evaluations should focus more on answering the *why* questions, and less on *how* to achieve good performance for the test collections at hand. Only this kind of research will put an IR engineer into the position to judge whether or not a specific model is applicable in a given situation, and then use it for making predictions about the properties of the system she is designing. As concrete first steps, I propose the following:

1. Theoretic research of the why–type should be encouraged, e. g. by having a separate conference track for these papers.
2. Rigid evaluation standards should be defined and enforced by corresponding reviewing guidelines.
3. Repositories for standardized benchmarks (like evaluatir.org) should be set up and maintained by our community.

The discussion in this paper has focused on classical, system oriented IR approaches, but it also holds for user-oriented approaches ([12, p. 105] calls for engineering in information seeking) as well as for new research areas in IR [2].

As final statement, I would like to cite a recent letter from the ACM president [6]: "We must develop better tools and much deeper understanding of the systems we invent and a far greater ability to make predictions about the behavior of these complex, connected, and interacting systems."

References

[1] N. Abdulmutalib and N. Fuhr. Language models, smoothing, and idf weighting. In *Proc. of the "Information Retrieval 2010" Workshop at LWA 2010, Kassel, Germany*, pages 169–174, 2010.

[2] J. Allan, W. B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, 46(1):2–32, 2012.

[3] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, 2002.

[4] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, and J. J. Lin, editors, *CIKM*, pages 601–610. ACM, 2009.

[5] P. Borlund. The concept of relevance in ir. *Journal of the American Society for Information Science and Technology*, 54(10):913–925, 2003.

[6] V. G. Cerf. Letter from the ACM president: Where is the science in computer science? *Communications of the ACM*, 55(10):5, 2012.

[7] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[8] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*, pages 480–487, New York, 2005. ACM.

[9] N. Fuhr. A probabilistic framework for vague queries and imprecise information in databases. In *Proceedings of the 16th International Conference on Very Large Databases*, pages 696–707, Los Altos, California, 1990. Morgan Kaufman.

[10] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 14(1):32–66, 1997.

[11] G. Génova. Is computer science truly scientific? *Commun. ACM*, 53(7):37–39, 2010.

[12] P. Ingwersen and K. Järvelin. *The turn: integration of information seeking and retrieval in context*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[13] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press.

[14] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[15] S. E. Robertson. Salton award lecture: On theoretical argument in information retrieval. *SIGIR Forum*, 34(1):1–10, 2000.

[16] D. Suciu, D. Olteanu, C. Ré, and C. Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[17] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485, 1986.

[18] K. Wang, X. Li, and J. Gao. Multi-style language model for web scale information retrieval. In F. Crestani, S. Marchand-Maillet, H.-H. Chen, E. N. Efthimiadis, and J. Savoy, editors, *SIGIR*, pages 467–474. ACM, 2010.