The Limits of Retrieval Effectiveness

Ronan Cummins[1], Mounia Lalmas[2] and Colm O'Riordan[3]

[1]School of Computing Science, University of Glasgow, UK
[2]Yahoo! Research, Barcelona, Spain
[1,3]Dept. of Information Technology, National University of Ireland, Galway, Ireland
ronan.cummins@nuigalway.ie

Abstract. Best match systems in Information Retrieval have long been one of the most predominant models used in both research and practice. It is argued that the effectiveness of these types of systems for the ad hoc task in IR has plateaued. In this short paper, we conduct experiments to find the upper limits of performance of these systems from three different perspectives. Our results on TREC data show that there is much room for improvement in terms of term-weighting and query reformulation in the ad hoc task given an entire information need.

1 Background

Best match systems in IR are the predominant model in both research and industry for developing search engines. From library searches to Internet search, these best match systems aim at returning only relevant documents given a user query. It has been stated in the past few years that the performance of ad hoc retrieval has plateaued or even that the performance of IR systems has failed to improve since 1994 [1]. This is one of the reasons that there has been a shift away from more traditional views of IR, to examine, among others, the querying process. The overall aim of our project (ACQUIRE - AutomatiC Query formUlation in Information REtrieval) is to learn how best to extract good queries given an information need from statistical and linguistic features of the terms and queries. However, for this short paper, we focus on finding the upper bound on performance from three different perspectives.

In a typical search scenario, a user who has an information need (*IN*) in mind, formulates this need into a query (Q). This is similar to the TREC formulation for the ad hoc task, where the topic *description* and *narrative* are a natural language description of the information need (*IN*) and the *title* is a sample query (Q). However, this sample query is only one of a myriad of queries that might be posed for the same information need.

In web search, usually a user poses short queries mainly because they have adapted their own behaviour for use with the system. Ultimately however, a user should to be able to communicate (and search) using an IR system in his/her own natural language and thus, automatic query extraction is an important goal in IR. If web systems provided a much better performance for longer queries, users may adapt their behaviour further. At present, there are many IR domains

in which a user already provides longer type queries. For example, in spoken retrieval, a user may utter a few sentences of an information need [4]. Similarly, in patent search [2], queries are often extracted from a document that has been filed for patent. In this paper, we report on experiments that aim to find the best query (Q) for a given information need (*IN*). The contribution of this paper is three-fold:

- Firstly, we determine the effectiveness of humans at manually extracting queries from an information need (*IN*). It is important to understand how good people are at the task of query generation.
- Secondly, we determine the effectiveness of the *best* possible query that might be extracted given an information need (*IN*).
- Thirdly, we determine the effectiveness of the *best* possible query for each topic (i.e. the universal upper bound of system performance for a topic).

The paper is organised as follows: Section 2 formally outlines the problem of query extraction and draws comparison to that of query term-weighting. We also outline a method for finding near optimal queries given a set of terms. Section 3 presents experiments in three subsections that map to the three different research questions above. Our conclusions are outlined in Section 4.

2 Query Generation

In this section, we describe the task of query formulation and outline a method to find near optimal queries given an *IN*. Research into tasks of query sampling [5], query modification, query re-weighting, query reduction [3] and query extraction, can be thought of in similar ways (usually the query is modelled as a vector of terms, and the weights are modified to change retrieval behaviour).

Given a best match IR system, a user interacts with it by formulating and entering a query for a specific *IN*[1]. More formally, for a *IN* of N terms, there are $2^N - 1$ possible queries that exist (ignoring the empty query). The example below shows all the possible queries for a three term *IN*.

$$\left\{\begin{array}{c|ccccccc} & Q_1 & Q_2 & Q_3 & Q_4 & Q_5 & Q_6 & Q_7 \\ \hline t_1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ t_2 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ t_3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right\}$$

Given that the number of possible queries that can be created for even a relatively small *IN* (e.g. 20 terms) is so large, we can suppose that a user creates sub-optimal queries. Essentially, given an IR system, we can reformulate the IR problem into that of finding the best query (Q) for an *IN*. As already stated, exhaustively finding the optimally performing query is infeasible given even a

[1] This information need may, or may not, be written down but for the purposes of this study we can assume that there is a written description of the *IN*.

small *IN* of 20 or 30 terms for each topic (as we would have to submit 2^N queries to the system). However, by adopting a standard *'greedy'* approach, we can build near optimal queries. A *'greedy'* approach finds the best one term query. Then, the approach finds the best two term query, assuming that it includes the best one term query. This approach requires submitting N^2 queries to the system. The query search space may be deceptive, but this approach finds high performing queries (as our results will show) and can be thought of as a conservative upper bound on performance given an *IN*.

From the example three term *IN* above, it should be noted that the problem of query extraction can also be viewed as a query term-weighting problem. For example, given the *IN*, a discrete weighting of 1 or 0 for the query terms (i.e. query term-weights) can be applied to the terms. Equally the problem might be viewed as a binary classification problem, classifying a term in the *IN* to be used in the query, or not used in the query. Nevertheless, reformulating the problem in this user centered way, allows us to view the problem as a classic AI search problem and allows us to specify the difficulty of the problem.

Given this user or query focused view of the IR problem, we now address the following questions; (1) How good are users at formulating queries given an *IN*? (2) What is the performance of the best possible query that could be extracted from the *IN*? and (3) What is the performance of the best possible query that could be presented to the system? The experiments that are outlined in the rest of this short paper attempt to answer these questions.

3 Experiments

In this section, we compare three approaches to generating queries. For the experiments in this paper we use the FT, AP, WSJ, and FR TREC sub-collections. These collections have different sizes and properties making our results more general. For the *IN*, we use the *description* and *narrative*. On average the *IN* contains 20 to 35 unique terms. The system we used for all our experiments is an implementation of the $BM25$ ranking function[2].

3.1 Manual Extraction

For the first experiment, we investigate the effectiveness of humans at manually extracting queries from a given *IN*. We gave the topic *descriptions* and *narratives* to a number of people in the broad area of IR (i.e. experts) who were asked to manually extract a good query (Q). The *title* was presented to them as a example query. Furthermore, we used the *title* of the information need as another pseudo expert. Table 1 presents the effectiveness of each set of manually extracted queries. Most users performed significantly ($\downarrow$ denotes 0.05 confidence level for Wilcoxon test) worse than simply entering in the entire *IN* (i.e. *description* and *narrative*) into the system. The *Max_User* label is the performance of

[2] We also ran experiments using a dirichlet prior language model and obtained very similar results.

the best of the four user generated queries for each topic. From this we can see that users can often choose queries that surpass the effectiveness of the entire *IN*. This experiments tells us that human extracted queries are, on average, less effective than simply entering the entire *IN* into the system[3], but could surpass the effectiveness of the *IN* in a best case scenario. A repeated-measures ANOVA performed on average precision of the queries across the four users showed no significant difference (on all but the FR collection[4]), telling us that the four users users are likely to perform similarly.

Table 1. MAP for Manual Query Extraction Task

	FT	FR	AP	WSJ
#Docs	210,158	55,630	242,918	130,837
#Topics	188 (251-450)	91 (251-450)	149 (051-200)	150 (051-200)
desc+narr	0.2529	0.2930	0.2098	0.3018
User_1 (title)	0.2281	0.2841	0.1632 ↓	0.2223 ↓
User_2	0.2482	0.2673	0.1773 ↓	0.2496 ↓
User_3	0.2212 ↓	0.2226 ↓	0.1833 ↓	0.2501 ↓
User_4	0.2302 ↓	0.2152 ↓	0.1888 ↓	0.2674 ↓
Avg_User	0.2319	0.2473	0.1782	0.2473
Max_User	0.3173	0.3572	0.2311	0.3163

3.2 Optimal Extraction

In this section, we present the results from the experiment that aims to find a conservative upper bound (i.e. near optimal query) on the performance given an *IN* (i.e. only using terms from the *description* and *narrative*). Table 2 shows the performance of the best query (Opt) using the *greedy* approach outlined in Section 2. It can be seen that if we could extract the best query from the *IN*, we could double the effectiveness (i.e. MAP) compared to the average user. This informs us that there is a lot more that might be achieved using the *IN* given. This might be useful in scenarios where a user poses a longer query or in situations where the *IN* is available. Also shown in Table 2 is the average length of the optimal queries found using our approach. We can see that the optimal queries are short compared to the entire *IN*. This result has implications for term-weighting schemes for longer type queries. This is because the extraction task can also be viewed as a query term-weighting problem. By taking account of terms already in the query, term dependent term-weighting scheme may be a fruitful avenue of research.

3.3 System Limit

In the final experiment of this short paper, we aim to find the upper bound on the performance of the system (i.e. is a MAP of 1 possible for a set of topics?). To

[3] We do acknowledge that entering the entire *IN* into the system is an added effort for the user for only a marginal extra benefit.

[4] This difference was not present using the language model as the IR system.

Table 2. Optimal Performance (MAP) for Query Extraction Task

	FT	FR	AP	WSJ
desc+narr	0.2529	0.2930	0.2098	0.3018
Avg_Length	(23)	(24)	(32)	(32)
Avg_User	0.2319	0.2473	0.1782	0.2473
Avg_Length	(3.9)	(3.8)	(4.7)	(4.7)
Opt	0.4832	0.5838	0.3776	0.4712
Avg_Length	(4.5)	(3.85)	(6.4)	(6.3)

find the upper bound on performance for individual topics, terms are extracted from the relevant documents. Again we use the same greedy approach outlined in Section 2 to find high performing queries. However, because there is a larger number of terms (i.e. those extracted from relevant documents), we only find optimal queries up to length of 10 terms to illustrate the general trend. Figure 1 shows the performance of the best queries found for each query length for a set of topics. The key labelled "SYSTEM LIMIT" is the conservative upper bound for the system for a set of topics. The other curves (labelled "IN LIMIT") show the performance of the optimal queries extracted from the *IN* (as per section 3.2). Firstly, we can see that perfect IR performance (i.e. *MAP* of 1) is achievable on one of the collections for a set of topics using queries of only five terms. Although, this collection is the smallest collection, our results would tend to suggest that near perfect retrievability is possible using best match systems. This further enhances the view that we might better improve IR effectiveness by concentrating on modifying the input to these systems.

Figure 1 also outlines the performance of the best query of each length extracted from the *IN*. We can see that the performance peaks at about six terms and decreases afterwards. This curve will decrease to the same performance of the entire *IN* once all terms are selected for use in the query. This evidence might help explain why users often simply submit short queries (i.e. short queries can be powerful).

4 Conclusion and Future Work

Overall, we have found that although people are good at extracting terms from an *IN*, entering the entire *IN* into a system is better. The average person achieves over 80% performance by entering a few terms compared to typing in the entire *IN*. Interestingly, we have found that the upper bound on the performance for query extraction is more than twice that of the average person, and close to twice the performance of the entire *IN*. Furthermore, we have found that given a fixed number of terms (as a person may formulate for an *IN*), optimal performance is achieved by only entering a small number of those query terms. Typically, given 20-30 terms that describe an information need, the optimal queries lie in the range of three to six terms. Finally, we show that best match systems are very powerful, as if the near perfect query is entered, these systems can achieve near perfect retrieval for small to medium sized databases.

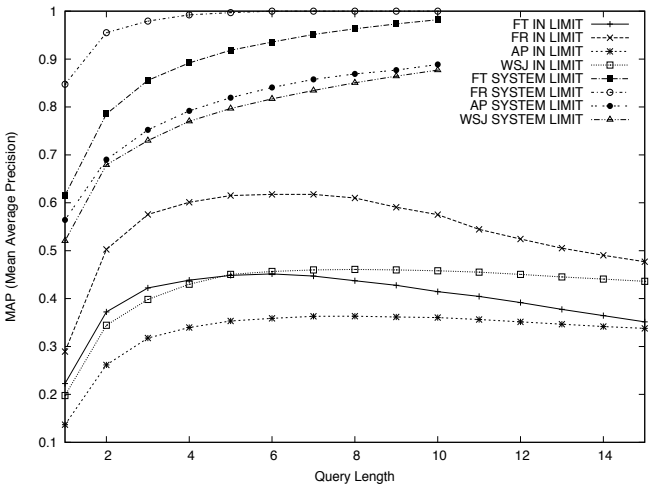

Fig. 1. Upper Limit (MAP) for System

Future work includes applying machine learning algorithms to learn the best query extraction methods given an information need. We plan to release the data and features gathered so that the task of query extraction can become a standard machine learning task for others in the community to research.

Acknowledgments Ronan Cummins is funded by the Irish Research Council (IRCSET), co-funded by Marie Curie Actions under FP7. The authors are grateful to the annotators from Galway and Glasgow who helped in the query extraction process. This paper was written when Mounia Lalmas was a Microsoft Research/RAEng Research Professor at the University of Glasgow.

References

1. Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *CIKM '09*, pages 601–610, New York, NY, USA, 2009. ACM.
2. Erik Graf, Leif Azzopardi, and Keith van Rijsbergen. Automatically generating queries for prior art search. In *CLEF*, pages 480–490, 2009.
3. Giridhar Kumaran and Vitor R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR*, pages 564–571, 2009.
4. Andrei Popescu-Belis, Jonathan Kilgour, Peter Poller, Alexandre Nanchen, Erik Boertjes, and Joost de Wit. Automatic content linking: speech-based just-in-time retrieval for multimedia archives. In *SIGIR '10*, pages 703–703, New York, NY, USA, 2010. ACM.
5. Linjun Yang, Li Wang, Bo Geng, and Xian-Sheng Hua. Query sampling for ranking learning in web search. In *SIGIR '09*, pages 754–755, New York, NY, USA, 2009. ACM.