Efficient Search Result Diversification via Query Expansion Using Knowledge Bases

Raoul Rubien
Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
rrubien@know-center.at

Hermann Ziak
Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
hziak@know-center.at

Roman Kern
Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
rkern@know-center.at

***Abstract*—Underspecified search queries can be performed via result list diversification approaches, which are often computationally complex and require longer response times. In this paper, we explore an alternative, and more efficient way to diversify the result list based on query expansion. To that end, we used a knowledge base pseudo-relevance feedback algorithm. We compared our algorithm to IA-Select, a state-of-the-art diversification method, using its intent-aware version of the NDCG (Normalized Discounted Cumulative Gain) metric. The results indicate that our approach can guarantee a similar extent of diversification as IA-Select. In addition, we showed that the supported query language of the underlying search engines plays an important role in the query expansion based on diversification. Therefore, query expansion may be an alternative when result diversification is not feasible, for example in federated search systems where latency and the quantity of handled search results are critical issues.**

I. Introduction

The ubiquitousness and high performance of contemporary Web search engines have shaped the way users interact with information retrieval systems. Since user queries tend to be short (typically just 2 to 3 terms [1]), they are often ambiguous [2], [3], [4]. Such queries as “java” can have different meanings and represent different information needs [5]: coffee, the programming language or the island. Likewise, a query “car” could refer to “buying a car,” “a car wash” or “a car magazine.” Given that the space for displaying a list of search results is limited, one may want to cover as many of the interpretations of the query as possible by the first few search results. The most common approaches to diversifying the result lists and boosting important intentions of the underlying query where proposed in [6], [4], [7], [8]. Drosou and Pitoura [9] surveyed the literature and introduced a classification scheme of the most common diversification techniques: i) content-based, ii) novelty-based and iii) coverage based. All three types are based on restructuring the final result list. The elaboration in [9] describes the diversity in general as follows: Given a set $\mathcal{X}$ of n available query-related items and a restriction k of the number of wanted results, the goal is to select a subset $\mathcal{S}^*$ of k items out of $\mathcal{X}$, that will maximize the diversity and relevance among the items $\mathcal{S}^*$.

The most common type of result list diversification can best be described as a form of greedy approximation: the query is sent to the search engine and the most relevant items are retrieved in the descending order of their relevancy. Next, the result list is reordered, beginning with the most relevant item. Each succeeding item is selected depending on i) the previously selected items, ii) the query or iii) a combination of both. For this method to work effectively, it is necessary to retrieve a large number of relevant items, typically much more than those that are finally displayed to the user. As such, diversification of search results via reordering creates higher requirements for achieving reasonable response times. First, the search process is longer since more top items have to be loaded, and even if the search itself is sufficiently quick, the retrieval of the details to be displayed takes a considerable time. Second, the re-ranking algorithm itself is often computationally complex and contributes to longer response times. The response time is an important factor since it strongly correlates with the perceived users’ satisfaction and acceptance [10], [11]. Brutlag [12] demonstrated that an artificially-introduced delay of 200ms to 400ms has an negative impact on user acceptance. Interestingly, even after the delay was removed the users did not return to their normal search behaviour for some time. Furthermore, the expansion of data continuously outweighs the technological progress and diminishes the speed increase of the computational infrastructure. Therefore, alternative ways to achieve a similar level of diversity are required, without having to rearrange the large result lists. If the search engine is treated as block box that cannot be changed and one does not want to modify the result list, there is only one option left: to modify the query before sending it to the search engine. Automatic query reformulation is a standard technique in the field of information retrieval [**?**]. One form of query reformulation is query expansion, when all terms of the original query are kept and only a number of related terms are added to the query. The relatedness is a critical aspect since adding an unrelated term to the query may introduce a query drift [16]. There are two types of query expansion, depending on how the related terms are computed. The first type, the global query expansion, operates independently from the search engine itself (e.g., related terms are generated via a look-up in a thesaurus). The second type, the local query expansion, takes the search engine in account. For example, the original query is issued to the search engine and then the search results are inspected for related terms. Pseudo relevance feedback is a closely related technique, with the first few hits assumed to be of high relevance. When applying their proposed evaluation framework, Clark et al. [3] discovered that diversity increased when pseudo relevance feedback was applied. Similar insights where reported by Strohmaier et al. [?]. Still two questions remain open, how does query expansion relate to the explicit diversification of search result lists and can the same level of diversification be achieved? To that end, we devised a pseudo relevance feedback-based query expansion approach and compared it with a state of the art result diversification algorithm. If the diversification via query expansion delivers a comparable level of diversification, it would be of great benefit for the developers of search engines and all its users due to lower response times and equally good search results.

II. Query Expansion

Our query expansion system can be categorized as a variation of a local query expansion system since the search is first conducted with the original, unexpanded query. However, instead of using the target search engine, which is used for the final result, an external knowledge base is searched to find related terms on demand. The results returned from the searched knowledge base are analysed and used as input for creating query expansion candidate terms. Next, the candidate terms are ranked and the top-ranked candidates are added to the query. Finally, the expanded query is applied to generate the final search result with the target search engine, which is then reported back to the user. Each of these steps allows a wide range of configurations and supports a number of parameters for tuning its behaviour.

a) Knowledge Base: In most cases, local query expansion is conducted using the same search engine, which generates the final search result set. In contrast, our system has two different search engines. The first one is exclusively used for query expansion. The returned results are not displayed to the user but utilized to generate the query expansion of the candidate terms. Next, the second search engine is invoked with the expanded query. The results of the expanded query search are then presented to the user as the result to the original query. For the first search engine we opted to query an existing knowledge base, in our case the English version of Wikipedia. We developed a search engine, based on the open source library Apache Lucene[1] to create an offline index of Wikipedia.

a.i) Indexing Strategy: At first, we downloaded the dump of all pages and articles[2], that had a single XML file containing the Wikipedia articles in their native MediaWiki syntax. Then we applied a MediaWiki parser in order to extract the text and all the meta data, including the paragraph information. This paragraph information is crucial for our indexing strategy. We observed that many Wikipedia articles became longer over time, contain more information and cover various aspects of a single concept. For example, an article about a city contains information about its climate, geology, economy, etc. Based on this, we split the Wikipedia articles into respective paragraphs and indexed them individually, following an existing procedure for Web search [?]. If the query matches multiple paragraphs, a single article may show in the result list multiple times. The internal ranking of the search engine should ensure that the best match is ranked the highest. Typically, the first paragraph of each Wikipedia article is a synopsis of the complete article and mentions the most important facts, which prompted us to generate query expansion of the candidate terms, with the first paragraph especially labelled in the index. Finally, each indexed paragraph of a Wikipedia article contains the following facets: i) the title of the Wikipedia article, ii) the title of the paragraph (if available), iii) the text of the paragraph itself.

a.ii) Searching Strategy: When searching the index, all three facets are queried at the same time and the final ranking corresponds to the combination of the matches found within these facets. With regard to the two facets corresponding to the title and paragraph title information, we allowed a partial overlap of the query terms with the facet's content up to 25% of the query terms to be absent from the respective facet. Matches within the initial paragraph were treated as a full match in the paragraph title. Due to the inner working of the ranking algorithm, documents containing a term in every facet were ranked higher and documents with matches in multiple facets were preferred. More specifically, we opted for the divergence from randomness algorithm [13] (DFR) as the ranking method due to the favourable reports in the literature [14]. To approximate the binomial model, we used the Poisson distribution, and since it was unsuitable for infrequent terms, we excluded all terms with a document frequency of less than 3 from the ranking. For smoothing, we chose the Laplace's law of succession (add one smoothing), taking the number of time a query term occurs within a facet into account. For normalization, the average length of the respective facet was compared with the actual length (number of terms). From the ranked search result list, we selected the first 10 hits for the further processing and to generate the query expansion of the candidate terms.

b) Candidate Selection & Ranking: For the candidate term purposes, we collected all terms from all matching facets, with the exception of known stop words, for which we used a stop word list. The terms were weighted individually for each facet, taking the score from the search hit into account and following the divergence from randomness weighting method. The final score for each term $s(t)$ is the sum over all its occurrences within the search result and the matching facets.

$$s(t) = \sum_{i \in S} \sum_{f \in F} DFR(boost(f) * score(d_i)) \qquad (1)$$

Where S is the set of all top search results, d_i is the ith result in the list and F is the set of all facets of d_i. The factor *boost* is defined as 0.1 for the paragraph title and 1 for all other facets. Given the score function, ranking of the candidate

[1] https://lucene.apache.org/ (Version 4.10.1)

[2] Date of dump: 2015-02-05

terms was computed, from which the top terms were then used for the query expansion. The computational effort for query expansion can be reduced to the cost of index searching.

c) Query Formulation: Once the related terms were selected, the query formulation began. To that end, the original query was combined with the additional, related query terms. This expanded query was then issued to the target search engine to generate the final search result set, which was the presented to the user without any further modifications. This allows us to study the effect of the query formulation and changes made on this stage on diversity. We wanted to test the assumption that a search system that can only support simple query languages may not be suitable for our purposes as systems with a richer support in the query syntax (e.g., not all search engines support conjunction and disjunction queries or grouping of multiple search terms). We implemented two strategies to combine the original query with the additional query terms.

c.iii) Our preferred query formulation strategy combined all added query terms as a disjunction query. This disjunction query was then added to the original query as a single query clause:

$$OrigQueryTerms\,OR\,(ExpTerm_1\,OR\,...\,OR\,ExpTerm_n)$$

c.iv) Alternatively, we implemented a simpler query formulation strategy, with all additional query terms appended as if the user had entered these terms. We expected this strategy to lead to a query drift rather than query formulation c.iii):

$$OrigQueryTerms\,OR\,ExpTerm_1\,OR\,...\,OR\,ExpTerm_n$$

Due to the underlying search engine, the scoring of c.iii) is different from c.iv) in cases when not all query terms match a document. Assuming that the query terms for document d_i are $T_n \in d_i$, $T'_n \notin d_i$, we denote term scores as s_n and document score as $S_i(query)$ to determine that, with strategy c.iii), the grouped expanded terms are attenuated in relation to the $OrigQueryTerms$ in cases 1), 2) and handled neutrally in 5). Cases 3) and 4) to illustrate strategy c.iv).

1) $S_i(T_1 \vee (T_2 \vee T_3 \vee T'_4) = s_1 + (s_2 + s_3 + 0) * (2/3)$
2) $S_i(T'_1 \vee (T_2 \vee T_3 \vee T_4) = 0 + (s_2 + s_3 + s_4) * (1/2)$
3) $S_i(T_1 \vee T_2 \vee T_3 \vee T'_4) = (s_1 + s_2 + s_3 + 0) * (3/4)$
4) $S_i(T_1 \vee T_2 \vee T_3 \vee T_4) = s_1 + s_2 + s_3 + s_4$
5) $S_i(T_1 \vee (T_2 \vee T_3 \vee T_4)) = s_1 + s_2 + s_3 + s_4$

III. Diversification Evaluation Setup

To evaluation our approach, we conducted three searches for each query q_i from a set of queries Q. First, the query was sent unchanged to the search engine and the result list $R(q_i)$ was accepted without any modifications. At this stage, the search results were expected to be of high relevancy but the diversity was expected to be low. Next, a state-of-the-art search result diversification algorithm was used to create another search result $R_{IA}(q_i)$, which should be a mixture of relevant and diverse results. Finally, we applied our query expansion to the original query and computed $q'_i = QExp(q_i, numExpTerms)$ but did not change the obtained search result list $R_{QE}(q'_i)$ at all. Subsequently, we compared the amount of diversification of the explicit diversification algorithm with our query expansion method.

d) Query Set Construction: For the evaluation purposes, a sufficiently large set of queries diverse domains was required. The query set was extracted from query logs contributed by Seifert et al. [15] and collected in course of the EEXCESS project[3]. The complete data set, including the query log, can be found online[4]. The query log contains queries entered by users and those automatically extracted from web-pages as result of the tasks in [15]. We filtered out duplicates, including near duplicates and non English query terms of the log. Our final query set Q consisted of over 70 manually-selected queries.

e) Measure of Diversity: In order to assess the amount of diversification we employed NDCG-IA (Normalized Discounted Cumulative Gain - Intent Aware) [8]. which is the intent aware version of NDCG. The required "ground truth" in our case was the result list $R_{IA}(q_i)$ from IA-Select, consisting of weighted items. We then computed NDCG-IA@k values of both lists $R_{IA}(q_i)$ and $R_{QE}(q'_i)$ with $R(q_i)$ as the ideally sorted list for each query. Finally, we compared the two values $NDCG\_IA@k(R_{IA}(q_i))$ and $NDCG\_IA@k(R_{QE}(q'_i))$ for all $q_i \in Q$. If the two values fell into the same range for the majority of queries, they were assumed to have achieved the same level of diversification.

f) Explicit Diversification of Search Results: For comparison purposes, we selected the IA-Select (Intent Aware - Select) algorithm for the comparison since it was reported to have good diversification results. This is a greedy algorithm which takes possible intents into account and was proposed by Agrawal et al. [8]. IA-Select requires each query to be assigned to a number of categories out of a classification scheme. To that end, we took the Wikipedia main categories[5] as list of categories and manually assigned each of the query a number of categories and a respective weight. IA-Select also requires the retrieved search result items to be linked to the same categorization scheme. Assigning categories to an item in the result list was originally conducted via classification using the Rocchio classification algorithm. IA-Select conducts a selection and reordering of the search result list, compiling a final search result list of k' (in [8] denoted as k) items. It iteratively selects items from the original result list based on their highest marginal utility. The marginal utility is defined as a product of the relevancy of the item and the overlap of the item's categories with the query's categories based on their respective weights and documents selected so far. Once the item with the highest marginal utility is added to the final search result list, a conditional probability is updated to reflect the inclusion. The item's weighted category assignments are deducted from the fine-grained article categories using a special categorization schema. This is repeated until k' items

[3] http://eexcess.eu/
[4] http://purl.org/eexcess/datasets/umlt
[5] http://en.wikipedia.org/wiki/Category:Main_topic_classifications

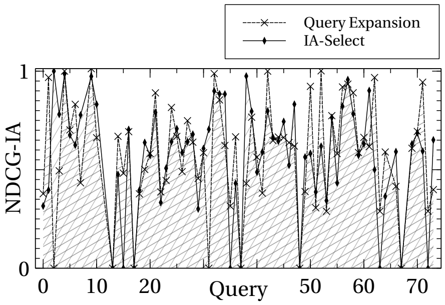

Fig. 1. Raw results of IA-Select and the query expansion algorithm with their respective NDCG-IA@10 values for each query. Although in most cases the NDCG-IA results appear to be similar, in some queries the two values disagree.

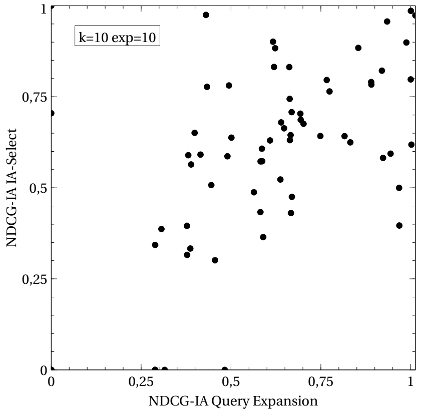

Fig. 2. Scatter plot of NDCG-IA@10 for the two diversification methods. Each occurrence o is defined as $o = (NDCG\_IA@10(R_{QE}(q'_i)), NDCG\_IA@10(R_{IA}(q_i)))$ with $q'_i = QExp(q_i, 10)$. The two distributions appear to be positively correlated.

are added to the final search result list, resulting in a balanced list of both relevant and diverse search results. For the purposes of item categorization, we did not follow Agrawal et al. [8] but rather opted for an alternative implementation based on DBPedia[6] and the Wikipedia category graph[7]. This way we could reuse the Wikipedia index that was previously used for pseudo relevance feedback. Since the Wikipedia category graph is very extensive, we applied a mapping scheme. Each of the categories was mapped to the main Wikipedia categories as follows: For each search result, we retrieved its (fine grained) categories. For each of these categories, we took a sub graph of its nearest 1000 neighbours, including the category itself. Within the sub-graph, we computed the shortest path from the category to each of the main categories. For each of the detected paths, we applied a spreading activation scheme to assign weights to the main categories. Finally, the weights of the main categories were normalized.

IV. Results

We show the results of our evaluation for a number of configurations of our query expansion technique. One of the most important parameters is the number of query terms to add to the query - exp. Another parameter is related to the diversification measure, NDCG-IA: the number of search results to compare with the "ground truth." We only report the results for the first $k = 10$ for each list. In Fig.1, raw results for all queries are presented for both algorithms: the explicit diversification method (IA-Select) and our query expansion technique (for $exp = 10$). In Fig.2, the same results are depicted as scatter plot, which shows that there is a certain extent of correlation between the two sets of results. In table I, we report the results of a number of correlation measures, including those for a varying number of terms added to the query. The Pearson's r, Spearman's rho and Kendall's tau correlation coefficients of the NDCG-IA values are also shown in Fig.3 over a range of different values of exp. The highest Pearson correlation is at $exp = 20$ term expansion with $r = 0.60$. Taking all correlation measures into account, the best configuration was the one with the number of query expansion terms equal to 10. Generally, the amount of diversification between the explicit diversification and query expansion appear to be similar for many queries. For these queries, diversity can be introduced via our approach as effectively as via the reference method. Higher correlation values are unlikely due to differences in the nature of the two algorithms: IA-Select is restricted to items from the original result list, while the search result list with the expanded query may contain many additional results. Although the Pearson's r correlation for 20 expansion terms was the highest, we believe that selecting only 10 additional terms to reduce the risk of a query drift [16] is appropriate for this type of technique. During our final evaluation run, we compared the two strategies of query formulation. In contrast to table I, table II shows lower correlation values for the number $exp = 10$ and strategy c.iv) (query expansion without grouping). Considering the correlation coefficients in Fig.3 only, a drop at $exp = 15$ can clearly be observed, which indicates that the query formulation strategy and the amount of expanded terms exp have a significant impact on the final results. Furthermore, the results demonstrate that search engines supporting a richer set of query operators are better suited for query expansion techniques, particularly with regard to diversity of the search results.

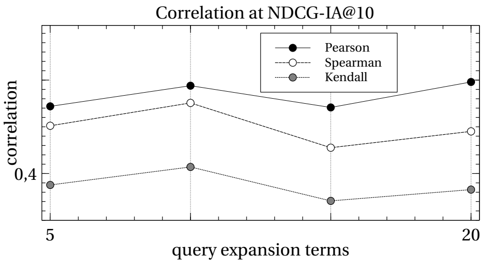

Fig. 3. Correlation values of the NDCG-IA@10 for the query expansion and IA-Select approach in relation to the count of the query expansion terms. The highest Pearson correlation occurs at term expansion $numExpTerms = 20$ with $r = 0.60$. Altogether the coefficients indicate a better correlation for the query expansion with 10 terms.

[6] http://www.dbpedia.org

[7] http://data.dws.informatik.uni-mannheim.de/dbpedia/2014/en/skos_categories_en.nt.bz2

Expansion Terms #	Pearson's r	Spearman's rho	Kendall's tau
20	**0.60**	0.49	0.37
15	0.54	0.46	0.34
10	0.59	**0.55**	**0.41**
5	0.54	0.50	0.38

TABLE I
CORRELATIONS RESULTS FOR QUERY EXPANSION STRATEGY C.III) WITH NDCG-IA@10 FOR THREE CORRELATION MEASURES, WITH THE HIGHEST CORRELATION FOR EACH OF THE MEASURES HIGHLIGHTED. THE NUMBERS INDICATES THAT 10 EXPANSION TERMS ARE GENERALLY ASSOCIATED WITH THE BEST PERFORMANCE.

Expansion Terms #	Pearson's r	Spearman's rho	Kendall's tau
10	0.46	0.42	0.30

TABLE II
CORRELATIONS RESULTS FOR QUERY EXPANSION STRATEGY C.IV) WITHOUT EXPANSION TERM GROUPING. THERE IS A PRONOUNCED DROP IN THE DIVERSIFICATION PERFORMANCE FOR THE SIMPLER QUERY FORMULATION STRATEGY.

V. CONCLUSION

Based on the results of our evaluation we can attest that for the majority of queries query expansion based on pseudo relevance feedback may be an efficient alternative in terms of explicit result diversification. One important parameter for query expansion is the number of terms to be added to the original query. In that regard, since we only discovered minor differences when comparing results with 10 and with 20 additional query terms, we recommend to use 10 terms. This would also reduce both the risk of a topic drift and the computational effort. Furthermore, we showed that the supported query language of the underlying system is an important factor for the diversification quality.

VI. FUTURE WORK

In the future, we would like to work on alternative ways of query formulation and investigate the effect of weighting query terms in the expansion process, the *boost* factor and the amount of documents contributing to the query expansion terms based on preliminary testing.

ACKNOWLEDGMENT

The presented work was developed within the EEXCESS project funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 600601. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG. We would like to thank Mark Kröll for his help and constructive comments.

REFERENCES

[1] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Comput. Surv.*, vol. 44, no. 1, pp. 1:1–1:50, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2071389.2071390

[2] F. Radlinski and S. Dumais, "Improving personalized web search using result diversification," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 691–692.

[3] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 659–666. [Online]. Available: http://doi.acm.org/10.1145/1390334.1390446

[4] S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 381–390. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526761

[5] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 881–890.

[6] A. Jain, P. Sarda, and J. R. Haritsa, "Providing diversity in k-nearest neighbor query results," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2004, pp. 404–413.

[7] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia, "Efficient computation of diverse query results," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 2008, pp. 228–236.

[8] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, ser. WSDM '09. New York, NY, USA: ACM, 2009, pp. 5–14. [Online]. Available: http://doi.acm.org/10.1145/1498759.1498766

[9] M. Drosou and E. Pitoura, "Search result diversification," *ACM SIGMOD Record*, vol. 39, no. 1, pp. 41–47, 2010.

[10] J. A. Hoxmeier and C. DiCesare, "System response time and user satisfaction: An experimental study of browser-based applications," *AMCIS 2000 Proceedings*, p. 347, 2000.

[11] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968, pp. 267–277.

[12] J. Brutlag, "Speed matters for google web search, june 2009."

[13] 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)*, vol. 20, no. 4, pp. 357–389, 2002.

[14] G. Amati, C. Carpineto, and G. Romano, "Comparing weighting models for monolingual information retrieval," in *Comparative Evaluation of Multilingual Information Access Systems*. Springer, 2004, pp. 310–318.

[15] C. Seifert, J. Schlotterer, and M. Granitzer, "Towards a feature-rich data set for personalized access to long-tail content."

[16] A. M. Lam-Adesina and G. J. Jones, "Applying summarization techniques for term selection in relevance feedback," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 1–9.

[17] D. Harman, "Relevance feedback and other query modification techniques." 1992.

[18] P. Ogilvie and J. Callan, "The effectiveness of query expansion for distributed information retrieval," in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, ser. CIKM '01. New York, NY, USA: ACM, 2001, pp. 183–190. [Online]. Available: http://doi.acm.org/10.1145/502585.502617