

Dynamic Facet Selection by Maximizing Graded Relevance

Michael Glass* and Md Faisal Mahbub Chowdhury* and Yu Deng* and Ruchi Mahindru
and Nicolas Rodolfo Fauceglia and Alfio Gliozzo and Nandana Mihindukulasooriya
IBM Research, T.J. Watson Research Center, Yorktown Heights, NY, USA
{mrglass, mchowdh, dengy, rmahindr, gliozzo}@us.ibm.com,
{nandana.m,nicolas.fauceglia,Gaetano.Rossiello}@ibm.com

Abstract

Dynamic faceted search (DFS), an interactive query refinement technique, is a form of Human-computer information retrieval (HCIR) approach. It allows users to narrow down search results through facets, where the facets-documents mapping is determined at runtime based on the context of user query instead of pre-indexing the facets statically. In this paper, we propose a new unsupervised approach for dynamic facet generation, namely *optimistic facets*, which attempts to generate the best possible subset of facets, hence maximizing expected Discounted Cumulative Gain (DCG), a measure of ranking quality that uses a graded relevance scale. We also release code to generate a new evaluation dataset. Through empirical results on two datasets, we show that the proposed DFS approach considerably improves the document ranking in the search results.

1 Introduction

Human-computer information retrieval (HCIR) is the study of techniques that takes advantage of human intelligence into the search process. Through a multi-step search process, it facilitates opportunities for human feedback by taking into account the query context. Examples of HCIR approaches include – faceted search, relevance feedback, automatic query reformulation, illustration by tag clouds, etc.

Faceted Search (FS) (Tunkelang, 2009), a form of HCIR, is a prevalent technique in e-commerce where document retrieval systems are augmented with faceted navigation. Facets are terms that present an overview on the variety of data available given the user query, thereby hinting at the most relevant refinement operations for zooming in on the target information need (Ben-yitzhak et al., 2008).

Equal contributions.

Traditional facet generation approaches present several drawbacks. Documents must be pre-tagged with an existing taxonomy, adding overhead in content curation and management. Moreover, such static facets lack contextual matching with documents or queries. Figure 1 shows an example of static/traditional facets.

Dynamic Faceted Search (DFS) overcomes such limitations (Dash et al., 2008). For **Dynamic facets**, the facet to document mapping is determined at run-time based on the context of user query instead of pre-indexing the facets statically. In other words, in an information retrieval (IR) system, there is no exclusive list of terms to be considered for dynamic facets and such facets are not known in advance. There is no pre-existing mapping of facets to the documents (that are indexed in the corresponding IR system). The mapping can only be created at the real-time when the query is submitted followed by generation of such facets based on the search results specific to the given query and are presented to the user along with the relevant documents.

In this paper, we present an approach for generating dynamic facets and selecting the best set of facets to be presented to the user. Hence, allowing the user to select relevant facets (if any) to interactively refine their queries, which in turn improves search results at each facet selection iteration. This interaction can be repeated until the user is satisfied with the results presented or no further refinement is possible.

Below we highlight the major contributions of our work –

- a new state-of-the-art unsupervised approach for dynamic facet generation (see Section 3) evaluated on two datasets (see Section 6), and
- a new benchmark dataset, `Stackoverflow-Technotes` (or,

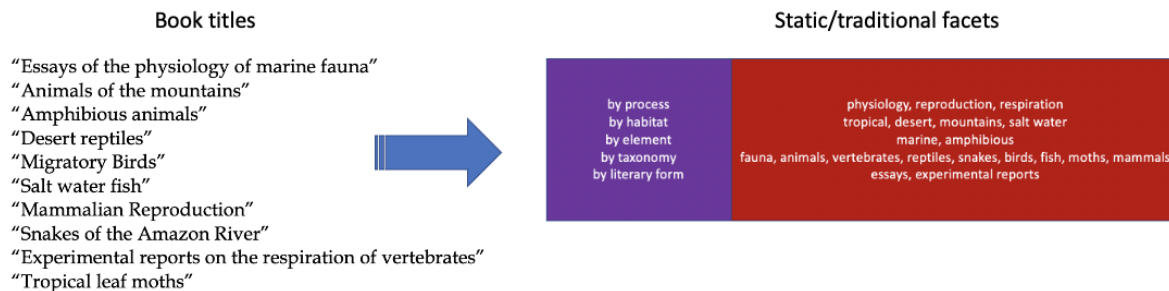


Figure 1: Example of static facets used to organize a set of book titles in a digital library.

simply Stackoverflow) Benchmark.¹ (see Section 5).

Rest of the paper is structured as follows. Section 2 includes a brief summary of related work with respect to DFS. Section 3 describes our proposed approaches. The next two sections (4 and 5) describes the experimental settings and datasets. In Section 6, we show the empirical results, both quantitative and qualitative. Finally, Section 7 concludes the paper and highlights perspectives for future work.

2 Related Work

A closely related research task of facet generation is to generate alternative queries, also known as query suggestion (Mei et al., 2008). Other related tasks are query substitution (Jones et al., 2006) and query refinement (Kraft and Zien, 2004). The main difference between these tasks and facet generation is that facets are not alternative/substitute/refined queries but rather a way to organize the search results obtained using the original query.

Another related task is query expansion (Xu and Croft, 1996) where the goal is adding related words to a query in order to increase the number of returned documents and improve recall accordingly. In contrast, selection of facets allow to narrow down search results.

There is a considerable amount of work on faceted search (Zheng et al., 2013; Kong, 2016). For brevity, here we focus on DFS only.

DFS can be divided into two categories. First, *DFS on databases* (Basu Roy et al., 2008; Kim et al., 2014; Vandic et al., 2018). Databases have a rich meta-data in the form of tables, attributes, dimensions, etc. DFS on databases focuses on the

¹We provide the codes for automatically creating the dataset using publicly available data, and also to run the simulated automatic evaluation. They can be found here - <https://github.com/IBM/Stackoverflow-Technotes-dataset>.

best possible attributes from the meta-data, to be presented as facets.

Our contributions are in the second category – *DFS on textual data*. An early approach was proposed by Ben-yitzhak et al. (2008), where the generated dynamic facets are constrained by the ability to sum pre-defined Boolean expressions. Dash et al. (2008) proposed an approach, given a keyword as query, to dynamically select a small set of “interesting” attributes and present their aggregation to a user. Their work is focused on evaluating the execution time rather than result re-ranking. Dakka and Ipeirotis (2008) proposed an approach using external resources, namely WordNet and Wikipedia, to generate facets given a query.

Our proposed DFS approach on text generates dynamic facets that are terms (which are not restricted), not just aggregated values, and does not rely on any external resource. Input queries can be natural language texts, not restricted to keywords.

In a recent relevant work, Mihindukulasooriya et al. (2020) proposed an unsupervised DFS approach that exploits different types of word embedding models to extract so called *flat* and *typed facets*. The *typed facets* are organized in hierarchies while the *flat facets* are simply a list of facets without hierarchy. They show empirically both set of facets yield similar results.

3 Proposed Dynamic Facet Generation

Given a ranked set of search results from a traditional search engine, our proposed approach, namely **Optimistic facet set selection**, tracks document ranking changes produced by selecting each candidate facet, and uses this information to select a subset of best possible facets.

We use the following notations in this section:

- $\mathbf{D} = [(d_1, s_1), (d_2, s_2), \dots, (d_n, s_n)]$, where score $s_i \in \mathbb{R}$, is a list of n documents in search

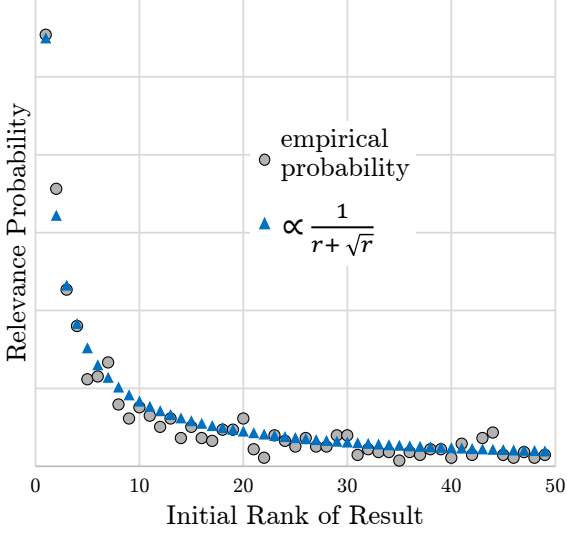


Figure 2: Relevance Probability

results for the initial query, q_0 returned by initial traditional IR component/search engine.

- $\mathcal{C} = \{f_1, f_2, \dots, f_c\}$ is a set of c terms to be considered as facet candidates.
- $F \subset \mathcal{C}$ is a set of k facets generated by the system as output, where k can be set by the user or the interactive search system.

3.1 Facet candidate generation

Given a user query and the respective search results (i.e. documents) from a search engine, we extract the terms from those candidate documents with a frequency above threshold θ_{freq} . Let us limit the expected number of dynamic facets to k . Given a pre-trained word embedding model (for the indexed document collection), cosine similarity, $sim(q_0, t)$, between the query and each term t is computed. Up to the top c terms with a minimum similarity score of θ_{sim} are kept as **facet candidates**.²

3.2 Optimistic Facet Set Selection

Our algorithm is built on two key assumptions:

- **Optimism**: the user will select the best facet: one that attains the best Discounted Cumulative Gain (DCG) (or other graded relevance measure).
- **Relevance Probability**: how likely a document is to be relevant is approximated by its rank in initial search results.

²We set $\theta_{freq} = 3$, $\theta_{sim} = 0.5$, and $c = \max(k^2, 50)$.

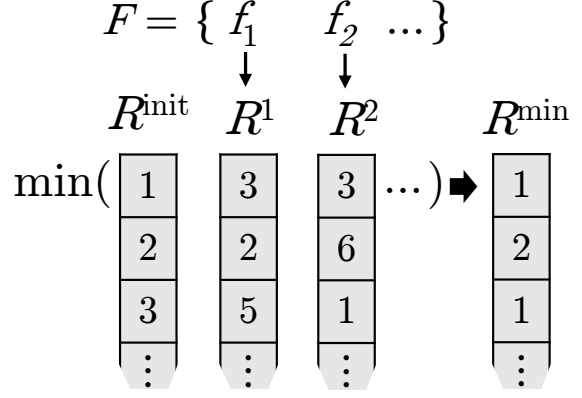


Figure 3: Minimum Rank (R^{min}) for Facet Set

Each candidate facet, f , is associated with some change in the scores of the document results, δ^f , and hence, some new ranking of the document results, R^f . Using the *filter* strategy, δ_i^f is set as $-\infty$ if f does not appear in document d_i , else zero. Experimenting with a strategy of computing the change in BM25 score (Robertson and Zaragoza, 2009) if f is added to the query, resulted in lower performance.

Suppose p_i is the probability of being relevant for the i th ranked document in the initial retrieval. We fit a curve to estimate p_i independent of the query or document results and find this probability to be roughly proportional to the inverse of the rank plus its square root. Figure 2 shows empirical probability of relevance and the curve to fit.

A facet set has a *minimum possible rank* for each document, the lowest rank that can be achieved by selecting any facet in the set, or no facet. We indicate this list of ranks as $R^{min} = [r_1, r_2, \dots, r_n]$ where $r_j = \min(j, \min_{f \in F}(R_j^f))$. The list of ranks R^{min} is closely connected with our optimistic assumption. If, for example, the single relevant document is in initial rank j , then R_j^{min} is the rank it will have after the user sees the initial results and optionally selects the best facet.

Consider the case (a majority in our datasets) where only one document is relevant. Then the expected DCG under the optimistic assumption is given by Equation 2. DCG is a standard metric in IR to measure the overall quality of the search results. DCG depends only on the ranks of the relevant ($rel_i = 1$) documents. Intuitively, we optimize DCG in expectation by providing facets that produce *different* and *likely* rankings for the returned documents.

$$DCG = \sum_{i=1}^n \frac{rel_i}{\log_2(1+i)} \quad (1)$$

$$\mathbb{E}(DCG_F) = \sum_{i=1}^n \frac{p_i}{\log_2(1+R_i^{min})} \quad (2)$$

We select a facet set to approximately optimize $\mathbb{E}(DCG_F)$ using greedy and local search. Both the greedy and local search phases of facet set selection rely on a function to select the facet candidate that will improve $\mathbb{E}(DCG_F)$ the most: $\text{Best}(\mathcal{C}, F, f^*, s^*)$. The greedy phase adds k facet candidates to the facet set, each time adding the facet that maximizes the set score. Local search tries to swap each facet in the facet set for some better facet candidate. This process could repeat until $\mathbb{E}(DCG_F)$ does not improve. Algorithm 1 shows pseudocode for these functions.

Algorithm 1 Greedy/Local Facet Set Selection

<pre> Best(C, F, f*, s*) for f in C - F do s ← E(DCG_{F ∪ {f}}) if s > s* then f* ← f s* ← s end if end for return f*, s* </pre>	<pre> Greedy(C, k) F ← ∅ for i ← 1 through k do f*, s* ← Best(C, F, ∅, 0) F ← F ∪ {f*} end for return F, s* </pre>
---	--

```

LocalSearch(C, F, s*)
  repeat
    s0 ← s*
    for f0 in F do
      F ← F / f0
      f*, s* ← Best(C, F, f0, s*)
      F ← F ∪ {f*}
    end for
  until s* = s0

```

4 Experiments

Evaluation Settings: We use the simulated user based automatic evaluation, called ORACLE, proposed by Mihindukulasooriya et al. (2020). For each iteration of the faceted search, the system presents a list of ranked search results and facets to the ORACLE. It selects the facet which retrieves the target document at the highest rank.

5 Datasets

5.1 TechQA Benchmark

The first dataset is an existing benchmark of real-world user questions in English in the domain of technical customer support, named the TechQA dataset (Castelli et al., 2020). The reason we choose this dataset is - the most recent work,

(Mihindukulasooriya et al., 2020)), that we are aware of for faceted search is evaluated on this dataset. The RoBERTa based state-of-the-art IR approach (Liu et al., 2019) that we use as one of the baselines also used this dataset. The TechQA dataset has **160** answerable questions in the Dev split and is aligned with a corpus of 801,998 publicly available IBM Technotes documents. We evaluate our approaches on these questions while treating the corresponding Technotes documents (containing the answers) as the corpus.

5.2 Proposed Stackoverflow Benchmark

In addition to the TechQA benchmark, we create a new dataset in the technical support domain to verify the generality of our approach. This allows us to evaluate it on a different benchmark containing real-world queries which are often noisy and not curated.

We are releasing the corresponding benchmark generation codes to the research community as part of this work. The dataset contains total **883** queries. It was created from Stackoverflow³ forum threads. We only considered those queries where the accepted answer posts contain link(s) to documents in the Technotes corpus (the same corpus as mentioned in the TechQA Benchmark). Here is how the released codes create this new benchmark:

- **Extraction of Candidate Question Answer (QA) Pairs:** We first identify the set of question posts and corresponding accepted answer posts from the StackOverflow post history dump. Then we extract the *title* and *body* of the identified *question posts* from post history, considering that the post body further elaborates context of the question.
- **Validation of QA Pairs with Result Links:** We retain the QA pairs where desired corpus links have been mentioned in answer posts. This ensures that the questions in the dataset have answer links from the Technotes corpus.
- **Generation of Benchmark Dataset:** We then extract the Technotes IDs from the answer posts to form the benchmark dataset. Figure 4 shows an example of an entry in the dataset, which includes an “*id*” field containing the id of a question post, a “*title*” field about the title of the question post, a “*body*” field which is the body part of the question post, and a “*relevant_docids*”

³<https://stackoverflow.com>

field with a set of Technotes IDs extracted from the corresponding accepted answer post.

The procedure described above is generic and can be replicated for other forums and corpora with similar characteristics.

6 Results

We implemented the *flat facets* proposed by Mihindukulasooriya et al. (2020) to compare with our results on both datasets. We use BM25 (Robertson and Zaragoza, 2009) as IR baseline for the *Stackoverflow* benchmark. For the *TechQA* dataset, we use the state-of-the-art IR approach of Zhang et al. (2020) built using RoBERTa (Liu et al., 2019) as baseline. Zhang et al. (2020) generously shared with us their system’s output for the *TechQA-DR* (i.e. document retrieval) task mentioned in their paper. We feed this output as input in our system as well as our implementation of Mihindukulasooriya et al. (2020) to extract facets from corresponding search results.

For a given query, we consider maximum 50 search results retrieved by the IR baseline. Then, the ORACLE accepts only up to 5 facets generated from a DFS approach, and chose only one facet (i.e. a single interaction with the DFS system) as a filter. If a corresponding search result does not containing this facet, it is discarded which changes ranks of some of the remaining search results.

6.1 Quantitative Evaluation

We use three standard evaluation metrics: *Discounted Cumulative Gain (DCG)*, *Mean Reciprocal Rank (MRR)*, and *Hits@K*. For *Hits@K*, we share the absolute number of queries where the expected document is ranked within top-K results.

Table 1 empirically compares our DFS approach against other systems. As evident from the results, *optimistic DFS* demonstrated remarkable edge over the DFS approach of Mihindukulasooriya et al. (2020) on both of the datasets in every single metric. Furthermore, our approach significantly improves the results of the underlying strong IR baselines in both datasets.

6.2 Qualitative Evaluation

For the qualitative evaluation, we selected a sample set of 22 random queries from the *Stackoverflow* dataset. We asked a Subject Matter Expert (SME), who is a customer support agent in the field, to

manually inspect the facets (produced by *optimistic DFS*) for each selected query.

According to the SME, a facet is considered useful, if it is contextually related but not already mentioned in the user’s (short) query (i.e. the ‘title’ in Figure 4) and either appears in (i) the fully specified query, aka ‘post’ (i.e. the ‘body’ in Figure 4), or (ii) in the target document.

Table 2 shows sample subset of “*User Query*”, their corresponding “*Top 5 Dynamically Generated Facets*”, “*Additional Relevant Facets Present in Post*” that the system could have considered to rank higher to place in the top 5, and “*SME Recommended Facets*” that the system should have presented (even though they are not seen in the post), as they are relevant for the corresponding user query. The values in the last two columns are provided by the SME.

The SME marked the dynamically generated facets into four following categories:

- “Facets seen in Post” (highlighted in *italic font*) – facets seen in the post body and our algorithm also generated e.g. ‘*ClearCase Remote Client (CCRC)*’;
- “Facets seen in Post and relevant for query” (highlighted in ***bold italic font***) – relevant facets seen in the post body and our algorithm also generated e.g. ‘***ClearCase Remote Client***’;
- “Facets unseen in Post” (highlighted in underline) – facets unseen in the post body that our algorithm also generated e.g. ‘Rational ClearCase SCM Adapter’, ‘rad’, ‘source control’;
- “Facets unseen in Post and relevant for query” (highlighted in **bold underline**) – relevant facets unseen in the post and our algorithm also generated e.g. ‘**dynamic views**’.

In summary, 22 randomly chosen queries and respective 5 facets per query generated from *Optimistic DFS* were evaluated by the SME. On average, our system generated 89% “Facets unseen in Post”, out of which 25% are relevant for queries. Among the 11% “Facets seen in Post”, 82% of them are found to be relevant for queries.

Metric	TechQA dataset				Stackoverflow dataset		
	<i>RoBERTa baseline</i>	<i>Zhang et al. (2020)</i>	<i>Flat DFS</i>	<i>Optimistic DFS</i>	<i>BM25</i>	<i>Flat DFS</i>	<i>Optimistic DFS</i>
DCG	0.76	0.82	0.84	0.91	0.18	0.24	0.29
MRR	0.69	0.77	0.80	0.89	0.13	0.20	0.26
Hits@1	92 (57.5%)	109 (68.1%)	116 (72.5%)	138 (86.3%)	75 (8.5%)	144 (16.3%)	205 (23.2%)
Hits@5	133 (83.1%)	141 (88.1%)	143 (89.4%)	150 (93.8%)	153 (17.3%)	228 (25.8%)	260 (29.4%)
Hits@10	137 (85.6%)	149 (93.1%)	151 (94.4%)	153 (95.6%)	200 (22.7%)	261 (29.6%)	293 (33.2%)

Table 1: DFS evaluation results using simulated user. “Flat DFS” refers to a DFS approach proposed by Mihindukulasooriya et al. (2020). “RoBERTa baseline” is the baseline for (our IR baseline) Zhang et al. (2020).

User Query	Top 5 Dynamically Generated Facets	Additional Relevant Facets Present in Post	Recommended Facets by SME
What are the differences between scm adapter and CCRC eclipse plugin?	dynamic views , <u>rad</u> , <u>source control</u> , <u>Rational ClearCase SCM Adapter</u> , ClearCase Remote Client	Eclipse SCM adapter	CCRC plugin, ClearCase perspective, Eclipse workspace, ucm
CLEARCASE XPN not parsed as variable in clearcase command	cleartool man , <u>cleartool mktrtype</u> , <u>command line</u> , <u>extended pathname</u> , <u>text file</u>	linux	cleartool find
CCRC ClearCase Remote Client - Error 'Config spec for view. . . needs to be synchronized	<u>ClearCase Remote Client (CCRC)</u> , <u>IBM Rational ClearCase Remote Client</u> , ucm , <u>vob</u> , <u>web view</u>	synchronize with stream, CCRC Version: 7.1.1	
Unable to undo rebase stream	<u>integration view</u> , rebase operation , <u>recommended baseline</u> , <u>target view</u> , vob		cleartool rebase
Web Service Auto Generated Files	<u>WSDL file</u> , ear , <u>roundtrip</u> , <u>web services</u> , <u>xsd</u>	SEI, ser, deser, helper files, BOUNTY EDIT	
What’s the easiest way to detect ”evil twins” in Rational ClearCase?	<u>ClearCase MultiSite</u> , <u>IBM Rational ClearCase</u> , <u>clearfsimport</u> , <u>file element</u> , ccrc	vob, ClearCase 7.1	cleartool, cleartool find
How to rename member baseline? Is it acceptable practice?	new baseline , <u>project</u> , <u>p_vob</u> , <u>rebase</u> , <u>rmb1</u>	cleartool rname, UCM, clearcase	lbtype, label type
Cleartool command: Get symlink path and target	VOB symlink , <u>cleartool rmelem command</u> , <u>config spec</u> , <u>global path</u> , pathname		

Table 2: Qualitative evaluation of *Optimistic DFS* output on the *Stackoverflow* dataset.

```

{
  "id": "57357897",
  "title": "Cannot locate webjars with SpringBoot application running on WebSphere",
  "body": "I have been trying to host Spring Boot application on WebSphere server.\n\nVersions :\n\n - SpringBoot 2.1.6.RELEASE\n - WebSphere Liberty 19.0.0.6 with JavaEE 8\n - webjars-locator 0.36\n\nThe app was successfully deployed on standalone Tomcat 9.0.22 and webjars could be resolved there. However for WebSphere I receive an error when trying to access webjars from `html` page ... ..",
  "relevant_docids": [
    "swg1PM99378"
  ]
}

```

Figure 4: Question Answer Pair Example

7 Conclusion

In this paper, we propose *Optimistic facet set selection*, a new unsupervised approach for dynamic facet generation for interactive search. It outperforms existing state of the art on two publicly available benchmarks, one of which we are releasing as part of this work.

We believe this new dataset will be useful for the research community for training and evaluating interactive models. Currently, our proposed approach does not have an active learning component and does not explicitly learn from the user feedback (e.g. fine-tuning an NLP model). However, we think our approach will serve as a strong baseline for the future interactive search approaches.

In future, we plan to investigate the following –

- how to leverage the proposed algorithm to generate facets automatically grouped by types.
- how dynamic facets can be generated using language models as Knowledge Bases.

Our vision is to transform the interactive search experience into a learnable knowledge discovery process.

References

- Senjuti Basu Roy, Haidong Wang, Gautam Das, Ullas Nambiar, and Mukesh Mohania. 2008. [Minimum-effort driven dynamic faceted search in structured databases](#). In *Proceedings of the International Conference on Information and Knowledge Management (CIKM 2008)*, pages 13–22.
- Ori Ben-yitzhak, Nadav Golb, Nadav Har’el, Ronny Lempel, Andreas Neumann, Shila Ofek-koifman, Dafna Sheinwald, Eugene Shekita, Benjamin Sznaider, and Sivan Yogev. 2008. [Beyond basic faceted search](#). In *Proceedings of the international conference on Web search and web data mining (WSDM 2008)*, pages 33–44.
- Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. [The TechQA dataset](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.
- Wisam Dakka and Panos Ipeirotis. 2008. [Automatic Extraction of Useful Facet Hierarchies from Text Databases](#). In *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE 2008)*, pages 466 – 475.
- Debabrata Dash, Jun Rao, Nimrod Megiddo, Anastasia Ailamaki, and Guy M. Lohman. 2008. [Dynamic Faceted Search for Discovery-driven Analysis](#). In *Proceedings of the International Conference on Information and Knowledge Management (CIKM 2008)*, pages 3–12.
- Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. [Generating query substitutions](#). In *Proceedings of the 15th International Conference on World Wide Web (WWW ’06)*.
- Hak-Jin Kim, Yongjun Zhu, Wooju Kim, and Taimao Sun. 2014. [Dynamic faceted navigation in decision making using semantic web technology](#). In *Decision Support Systems*, volume 61, pages 59 – 68.
- Weize Kong. 2016. [Extending Faceted Search to the Open-Domain Web](#). Ph.D. thesis, College of Information and Computer Sciences, University of Massachusetts Amherst, MA, USA.
- Reiner Kraft and Jason Zien. 2004. [Mining anchor text for query refinement](#). In *Proceedings of the 13th International Conference on World Wide Web (WWW ’04)*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.

- Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query suggestion using hitting time. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08)*.
- Nandana Mihindukulasooriya, Ruchi Mahindru, Md Faisal Mahub Chowdhury, Yu Deng, Nicolas Rodolfo Fauceglia, Gaetano Rossiello, Sarthak Dash, Alfio Gliozzo, and Shu Tao. 2020. Dynamic faceted search for technical support exploiting induced knowledge. In *International Semantic Web Conference*, pages 683–699. Springer, Cham.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.
- Daniel Tunkelang. 2009. [Faceted Search](#). In *Synthesis Lectures on Information Concepts, Retrieval, and Services*, volume 1, pages 1–80. Morgan & Claypool Publishers.
- Damir Vandic, Steven S. Aanen, Flavius Frasincar, and Uzay Kaymak. 2018. [Dynamic Facet Ordering for Faceted Product Search Engines](#). In *IEEE Transactions on Knowledge and Data Engineering*, volume 29, pages 1004 – 1016.
- Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)*.
- Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avi Sil, and Todd Ward. 2020. [Multi-stage pre-training for low-resource domain adaptation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5461–5468, Online. Association for Computational Linguistics.
- Bweijunl Zheng, Wei Zhang, and Xiaoyu Fu Boqin Feng. 2013. A survey of faceted search. *Journal of Web engineering*, 12(1&2):041–064.