Utilizing Query Facets for Search Result Navigation

Jan Friedrich, Christoph Lindemann, and Michael Petrifke
Department of Computer Science
University of Leipzig
Augustusplatz 10, 04109 Leipzig, Germany
{friedrich, cl, petrifke}@rvs.informatik.uni-leipzig.de

***Abstract*—Facets provide an efficient way to analyze and navigate the search result space. However, we believe facet selection has been guided by suboptimal facet and facet term properties. In this paper, we present features that rank facets based on their utility to partition the search result documents. Moreover, we propose a computationally inexpensive facet generation algorithm, provide a novel approach to extract term lists from HTML documents, and analyze facet feedback models using BM25F as baseline retrieval model. We show that the proposed approach outperforms existing algorithms and discuss the significantly reduced computational costs.**

Keywords-faceted web search; query facets; interactive feedback

I. INTRODUCTION

For most people, the way they interact with web search engines has not changed significantly in the last decade. They still issue queries manually and review lists of result documents. The most significant and obvious user interface changes were the introduction of verticals (e.g. images, videos, and news), query autocomplete, and question answering (e.g. Google Knowledge Graph). However, most internet users are also acquainted with faceted search: any e-commerce website, any library and most catalogues of any kind employ this technique to provide an accessible and fast way to locate arbitrary objects. We believe that most users would appreciate the utilization of this idea in web search.

However, this is no trivial task. The ultimate goal of Faceted Web Search [6] is to support the user to accomplish his search task. Previous work [2], [4], [7], [11] mainly focused on the idea of using existing taxonomies or on generating facets for an entire corpus offline after indexation. These approaches lack the adaptation to the document result space or the user intent, and are too narrow. We propose web search facets that automatically recognize different subtopics, partition the search result space evenly and exhaustively per subtopic, and still contain only a small number of terms.

Only recently, Dou et al. [3] published first ideas to generate query-specific facets solely using the contents of search result documents. Kong et al. [5] improved their approach and provided a method to assess the search utility of extracted facets [6]. Their evaluation analyzes the search quality in relation to the time units a user consumes to scan facet lists. In contrast, we believe that as long as the facet system obeys some reasonable restrictions on the number of generated facets and the number of terms per facet, only the overall utility is relevant.

In this paper, we present novel facet features and the facet extraction algorithm NAV that ranks facets based on their utility to partition the search result set. This approach reduces the computational effort significantly:

- Term specific features like the collection-level document or term frequency are not required. These are especially expensive for multi-word terms.
- Usage of the simple bag-of-words model.
- Ranking of a small list of facets compared to a large list of terms.
- No re-clustering of potential facet terms into facets.

Our evaluation shows that we generate facets at least on par with the results of Kong et al. [6]. However, their optimal facet feedback model fails in our experiments. As we employ a different baseline retrieval algorithm (BM25F [10] vs. SDM [8]), we recognize the requirement of a facet feedback model specific to the retrieval algorithm. Our proposed BM25F-tailored soft ranking model shows properties similar to the SDM-optimized models of [6].

The remainder of this paper is organized as follows. Section 2 summarizes related work on Faceted Web Search. In Section 3, we provide a primer on query facets as background. Section 4 introduces our novel approach for navigation-focused facets. The associated facet feedback mechanism for BM25F is introduced in Section 5. A comprehensive performance study of the proposed approach is presented in Section 6. Finally, concluding remarks are given.

II. RELATED WORK

Faceted search in the context of web retrieval is a relatively novel field. Most proposals provide solutions for at least one of the following problems: (1) extraction of facets and facet terms that reflect the content of the search result documents (2) ranking of facets and presentation of facets that are appropriate for the query and the user (3) feedback of user selected facet terms to the search engine. Two different strategies exist to solve the first problem: (1a) pre-compute document specific facets and facet terms (1b) dynamically extract the facets from the search result documents.

The most straight forward proposals for problem (1) focused on Wikipedia [7] and verticals (e.g. images [11]). Meta data (Wikipedia categories, keywords and comments) are directly utilized to extract and rank facets. However, to

utilize faceted search in general purpose search engines, facet extraction has to rely on document features generally available (e.g. textual content, link graph). Until recently, most approaches evaluated these document features in the context of external hierarchies (e.g. WordNET [9], Wikipedia categories) [2], [4], [7]. Finally, [3] and [5] introduced the idea to solely rely on the textual content of the search results. They proposed to extract term lists using HTML, visual, and textual patterns and to re-cluster these lists into facets.

Solutions to the facet ranking problem (2) are diverse and often connected to the facet extraction and assignment algorithm. However, they mainly utilize term frequencies [3], [5] or the facet memberships of search result documents [2].

Another open discussion concerns the facet feedback mechanism (3) and the appropriate utility evaluation of extracted facets. Facet feedback is mostly neglected; user studies or clustering metrics compare system-generated with user-selected facets. Recently, [6] and [12] analyzed the inefficiency of Boolean filtering in the context of web search and proposed soft ranking models. Additionally, [6] described the first user model to measure the real search utility. Opposed to [6], we exclusively use features connected to the partition and navigation properties of facet candidates to assess their utility.

III. PRIMER ON FACETED WEB SEARCH

Objects stored and indexed in relational databases and information retrieval systems are represented by a number of attributes. Reference management systems, for example, provide among other information the names of the authors, the publication date, the main topics and the conference name. These attributes enable users not only to query databases in an information retrieval manner but also to navigate the document space. Users specify required values for some of the attributes und filter irrelevant documents. In faceted search the value space of each attribute constitutes a single facet that is named after the attribute. Therefore, if we apply faceted search to the above described reference management system, we create a facet called "author" represented as the set of author names. A user now might select authors he is interested in to reduce the search result space.

Faceted Web Search [6] applies this idea to the web. However, facets in this context are not merely stored properties of the indexed web documents. In contrast, Faceted Web Search dynamically analyzes the search results, independently per query, and extracts a number of abstract facets. These facets $F = \{t_1, t_2, \dots, t_n\}$ are sets of facet terms t that, according to some metric described in the next sections, are efficient in providing users with a tool to refine their search intent.

As facets are already used in other fields, Faceted Web Search has to obey some common expectations. In the context of e-commerce and library systems users are accustomed to utilize facets as Boolean filters. These systems only display objects matching all of the user selected facet entries. Consequently, Faceted Web Search has to behave similarly to facilitate its acceptance in web search. Unfortunately, previous work [6] has shown that Boolean filtering, as applied in above examples, provides no utility at all in this new context. Thus, Faceted Web Search not only requires new algorithms to extract facets but also feedback models to utilize selected facet terms. Additionally, most terms of one facet are expected to be mutually exclusive: only very few facet terms of one facet match the same document (e.g. one publication year, one journal, small number of topics).

Unfortunately, facets potentially distract users from examining the search results. Thus, it is crucial to restrict the number of shown facets and facets terms to a minimum while still providing relevant navigation options for multiple search intents.

In summary, Faceted Web Search combines elements of query subtopic extraction, search result clustering and user feedback models.

IV. FACETED WEB SEARCH

A. Facet Extraction with Meta Patterns

A very successful idea to generate facets for HTML documents is based on the extraction of lists from HTML pages [3], [5]. These approaches utilize (1) lexical patterns (lists in free text) (2) HTML patterns (based on HTML list and table elements, e.g. and <ol>) and (3) visual repeat regions. The resulting raw lists are post-processed and their terms re-clustered to generate the final facets. A ranking algorithm ensures only the most useful facets are presented to the user.

The most recent work on the subject [6] utilized only extraction method (1) and (2). However, modern web design often applies CSS to general HTML tags (e.g. div, p) to create visual lists. Computationally expensive approaches like (3) that try to simulate the human visual perception are able to extract these lists. Fortunately, we found most visual lists, which are relevant for Faceted Web Search, to have the same basic HTML structure: an arbitrary HTML element contains multiple structurally identical children and the text content of each of these children constitutes a single list item. The structural identity of siblings can be assessed by comparing their HTML sub-trees on the basis of element names. We call this approach the *meta pattern*.

Some practical assumptions reduce the number of comparisons significantly: (a) lists have at least three items and (b) the HTML sub-tree of each list item has a maximum tree depth of five. Additionally, an implementation of this algorithm has to take care of some other HTML elements like comments, scripts, and blank elements.

Subsequent to candidate list extraction, cleaning techniques (e.g. stop word removal and removal of non-alphanumeric symbols) are employed. However, our proposed approach does not re-cluster the candidate lists or facet terms. Useful facets, as described in previous papers (e.g. "Lost" actors, Mars rovers), already occur as lists on web pages. Therefore, we consider each candidate list as final facet and ignore overlapping facets at this time. The next subsection presents a ranking algorithm that removes these anomalies.

B. Facet Ranking with NAV

Our approach does not post-process the candidate lists, i.e. the facets, extracted in the previous section. However, during

candidate list extraction, it removes the text sections and HTML sub-trees of the processed documents that constitute a list candidate. The resulting documents may still contain terms of the removed lists in other locations.

This approach is crucial for the following facet ranking algorithm that builds on a simple binary relevance assessment of the facet term $t \in F$ for document d: t is a valid value for d in facet F, if d contains t outside of lists. Utilizing the above method, our approach transforms each search result document d into the bag-of-words representation $d' = \{t'_1, t'_2, \dots, t'_n\}$ containing only potentially relevant terms t'. We call d' the *condensed document representation*. Accordingly, we define the *condensed search result* $D' = \{d'_1, d'_2, \dots, d'_m\}$ that is utilized to calculate the facet ranking features below.

Opposed to [6], we use ranking features of the facets themselves. Focusing on the navigation properties of the generated facets, we refer to our approach from now on as *NAV* and define the following four features: subtopic coverage C_F, partition size equality S_F, reciprocal of the mean number of facet terms per page P_F, and the number of facet terms T_F. The final rank of facet F is the linear combination of above features:

$$R_F = \alpha C_F + \beta S_F + \gamma P_F + \delta T_F \quad (1)$$

We further define $D'_F = \{\, d' \mid d' \in D',\ d' \cap F \neq \emptyset\}$ and $D'_t = \{\, d' \mid d' \in D',\ d' \cap \{t\} \neq \emptyset\}$ as the sets of condensed search result documents containing at least a single term of facet F or one specific term t, respectively.

Subtopic coverage C_F recognizes the fact that the original query might have numerous interpretations, but each facet is only relevant for one of these possible search intents. We approximate the number of sub-intents $\#I$ and calculate a distance measure to the expected number of documents matching at least one of the facet terms of F:

$$\#\mathrm{I(D)} = \log(|\mathrm{D}|) \quad (2)$$

$$C_F = \exp\left(-\frac{\left|\frac{|D|}{\#\mathrm{I(D)}} - |D'_F|\right|}{10}\right) \quad (3)$$

Size equality S_F is a measure of the equality of the D'_t document set sizes with μ_F^S being the mean set size.

$$\mu_F^S = \frac{\sum_{t \in F} |D'_t|}{|F|} \quad (4)$$

$$S_F = 1 - \frac{\sum_{t \in F}\left(\mu_F^S - |D'_t|\right)^2}{\sum_{t \in F} |D'_t|^2} \quad (5)$$

The reciprocal of the mean number of facet terms per page P_F is used to prefer facets whose facet terms' co-occurrence rate is very low:

$$\mu_F^C = \frac{\sum_{d' \in D'_F} |d' \cap F|}{|D'_F|} \quad (6)$$

$$P_F = \frac{1}{\mu_F^C} \quad (7)$$

And finally, T_F is used to prioritize larger facets:

$$T_F = log|F| \quad (8)$$

Subsequent to ranking, overlapping facets, i.e. facets that share at least one term, have to be removed. NAV employs an algorithm that loops over the facets in descending rank order and removes any lower ranked, overlapping facet. Thus, if the algorithm finds slightly different versions of essentially the same facet, the higher-ranked facet remains. On the other hand, the proposed approach might keep facets that share terms with higher ranked, but already removed candidates.

Finally, NAV sorts the facet terms of each facet in alphabetical order. This is the behavior a user might expect. Moreover, the facet generation is not capable of deducing the more specific user intent or subtopic the user is interested in. Therefore, we believe ranking the facet terms in any specific order according to some term importance to be non-beneficial.

V. FACET FEEDBACK

The feedback model defines how user selected facet terms are used to improve the web search result in terms of matching the user intent. We borrow the notation of [6] and use t^u for user selected terms (feedback terms), $F^u = \{t_1^u, t_2^u, .., t_o^u\}$ for the set of feedback terms of facet F (feedback facet) and $\mathcal{F}^u = \{F_1^u, F_2^u, \dots, F_p^u\}$ for the set of non-empty feedback facets.

A. Feedback Models for BM25F

Feedback models utilize user selected facet terms to adapt the search result to the user intent. Kong et al. [6] examined different feedback models: Boolean filtering and soft ranking. While the first model removes search results that do not contain each feedback term, or at least one feedback term per feedback facet, soft ranking combines the original document IR score S with a score based on the document-to-facet match S_E. They found Boolean filtering to be too restrictive to be useful in Faceted Web Search, thus we focus on soft ranking:

$$S'_E(d, q, \mathcal{F}^u) = \lambda S(d, q) + (1 - \lambda) S_E(d, \mathcal{F}^u) \quad (9)$$

They further examined two different implementations of S_E, which performed very similarly in the experimental evaluation. Therefore, we utilize the term expansion model ST in the following sections:

$$S_{ST}(d, \mathcal{F}^u) = \frac{1}{N} \sum_{F^u \in \mathcal{F}^u} \sum_{t^u \in F^u} S(d, t^u) \quad (10)$$

BM25F [10] is utilized as baseline retrieval model for $S(d, q)$ as well as for $S(d, t^u)$ in all of our experiments. Preliminary examinations in preparation of this paper revealed a poor performance of ST in this context. Therefore, we introduce the third expansion model TT. In contrast to ST, TT sums the feedback term scores up:

$$S_{TT}(d, \mathcal{F}^u) = \sum_{F^u \in \mathcal{F}^u} \sum_{t^u \in F^u} S(d, t^u) \quad (11)$$

B. Evaluation Model

We believe the impact on search result quality is the most important evaluation measure. Therefore, we focus on the extrinsic evaluation of the top ranked facets, feed user selected facet terms back to the soft ranking models of the last section, and compare the search quality.

We assume the perfect user, who correctly deduces the importance of each facet term solely on the basis of the search result page and his search intent. He incrementally adds the most helpful facet term, reviews its impact on the ranked documents, and then chooses the next term. By simulating this user, we are able to measure the utility of the facet extraction algorithms in terms of macro averaged nDCG@10 and nDCG@20 values.

Kong et al. [6] called the facet terms that improve search result quality significantly *oracle terms*. In contrast to our approach, that incrementally adds one oracle term, they classified multiple oracle terms based on their impact on the original search result. Moreover, they applied them in the order of their intent-unaware facet and facet term score.

VI. EVALUATION

A. Setup

We evaluate the facet extraction algorithm on the ClueWeb09[1] Category B dataset. The index is stemmed using the English Porter[2] stemmer and the PCFG parser of CoreNLP[3] is utilized to extract lexical candidates. Our experimental search engine system uses BM25F [10] as baseline retrieval model, but we additionally enforce Boolean AND retrieval. We set $\lambda = 0.5$. TREC 2011 diversity task queries relevance judgments [1] provide the basis for the macro-averaged nDCG scores.

Each TREC 2011 diversity task query contains subtopics and relevance judgements on the subtopic level. We use the original query to generate the facets, but identify oracle terms on subtopic level. As a result, the set of feedback terms is different for each subtopic. We then calculate nDCG@k per subtopic and average those results to assign an nDCG@k score to each query. The average of the query-level measurements is reported as macro-averaged nDCG@k in the following evaluations.

We use the graphical model of Kong et al. [5] as baseline, more precisely their QF-I algorithm, as it outperformed QF-J in most of their experiments. We optimize the facet term and term pair weights as well as the QF-I clustering parameters to maximize nDCG@20. During optimization, the oracle term is selected from the best ranked facet. The reported term weights from [5] divided by 10 produce optimal results (Table I). We apply their term pair weights unaltered. Our optimal QF-I clustering parameters are $\omega_{min} = 0.6$ and $d_{max} = 0.1$. The weights of NAV are optimized accordingly and we present the employed weights in Table II.

Surprisingly, facets extracted by lexical patterns always degrade search quality in our experiments, so we do not report their detailed results in the following.

[1] http://www.lemurproject.org/clueweb09.php

[2] http://snowball.tartarus.org

[3] http://nlp.stanford.edu/software/corenlp.shtml

TABLE I. QF-I TERM FEATURES AND WEIGHTS

Feature	Weight
listTF.listIDF	0.26424
listSF	0.21374
wDF	-0.10754
TF.clueIDF	0.10115
SF	0.06873

TABLE II. NAV FEATURES AND WEIGHTS

Feature	Weight
Subtopic coverage	-1.5
Partition size equality	0.7
Mean number facet terms	1.0
Number facet terms	0.3

B. Single Term Feedback

First, we compare the impact of one feedback term on search quality for numerous facet generation methods and parameters. QF-I and NAV post-process lists extracted from the top-20 or top-50 search results using HTML patterns and the proposed meta pattern. The oracle term is chosen from the highest ranked facet. As both soft ranking expansion models calculate the same score in this scenario, they are not considered at the moment. Table III summarizes the results.

NAV achieves considerable higher scores than QF-I, independent of the list extraction algorithm or the number of parsed search results. Moreover, in two scenarios QF-I struggles to find useful oracle terms at all, as the macro-averaged nDCG@10 score is lower than the score of the original search results. Surprisingly, the utilization of the meta pattern impairs search quality in most experiments.

Table IV shows the results, in case we choose the oracle terms from the top-3 facets. Using the optimal candidate extraction algorithm and the right number of top-k search results, QF-I and NAV perform very similarly. QF-I achieves a minimal higher nDCG@10 score while NAV's nDCG@20 score is insignificantly increased. Both algorithms benefit from lists extracted by meta patterns. However, NAV requires more documents than QF-I to achieve comparable results, while QF-I's performance is reduced by increasing the number of documents. We believe this is in the nature of our chosen NAV features and the dataset. TREC diversity task queries are underspecified on purpose to retrieve documents on different topics. The features of NAV however require recurring topics

TABLE III. SINGLE TERM FEEDBACK PERFORMANCE USING TOP-1 FACETS

Facet Ranking	Candidate List Extraction	Parsed Docs	nDCG @10	nDCG @20
No facets			0.0672	0.0759
QF-I	HTML	20	0.0699	0.0805
QF-I	HTML	50	0.0662	0.0798
QF-I	HTML + Meta	20	0.0673	0.0788
QF-I	HTML + Meta	50	0.0649	0.0763
NAV	HTML	20	0.0736	0.0877
NAV	HTML	50	0.0704	0.0839
NAV	HTML + Meta	20	0.0721	0.0858
NAV	HTML + Meta	50	0.0705	0.0778

TABLE IV. SINGLE TERM FEEDBACK PERFORMANCE USING TOP-3 FACETS

Facet Ranking	Candidate List Extraction	Parsed Docs	nDCG @10	nDCG @20
No facets			0.0672	0.0759
QF-I	HTML	20	0.0824	0.0919
QF-I	HTML	50	0.0737	0.0915
QF-I	HTML + Meta	20	0.0919	0.0954
QF-I	HTML + Meta	50	0.0780	0.0911
NAV	HTML	20	0.0808	0.0929
NAV	HTML	50	0.0857	0.0932
NAV	HTML + Meta	20	0.0800	0.0915
NAV	HTML + Meta	50	0.0911	0.0960

TABLE V. MEAN NUMBER OF FACET TERMS OF THE TOP-3 FACETS

Facet Ranking	Candidate List Extraction	Parsed Docs	# Terms per Facet
QF-I	HTML	20	6.29
QF-I	HTML	50	7.69
QF-I	HTML + Meta	20	6.63
QF-I	HTML + Meta	50	8.26
NAV	HTML	20	7.51
NAV	HTML	50	6.94
NAV	HTML + Meta	20	7.62
NAV	HTML + Meta	50	7.27

and the top-20 documents might just be too diverse.
Table V completes the single feedback term analysis. It shows the mean number of facet terms considering only the top-3 facets. The results support our discussion of NAV benefitting from an increased number of documents. Using only 20 documents, the number of facet terms is the most dominant feature and larger facets are preferred. With an increasing number of documents the other features can be assessed correctly so NAV extracts smaller facets, containing more useful terms.

C. *Multi Term Feedback*

Finally, we analyze the multi term behavior of the two soft ranking expansion models in connection with QF-I, NAV and BM25F as baseline retrieval model. QF-I and NAV utilize the optimal configurations we learned above. In contrast to the last experiment, we choose oracle terms from the top-5 facets. Fig. 1 plots our findings.

Obviously, ST is not able to utilize more than one QF-I or NAV facet term to improve the search result. This is in contrast to the findings of Kong et al. [6]. However, TT makes efficient use of up to three feedback terms. These findings are in line with previous work. Additionally, the TT expansion model exhibits for two feedback terms selected from NAV facets a significantly higher nDCG@20 score than for two terms selected from QF-I facets. We conclude that different baseline retrieval models might require their own specific soft ranking feedback model.

VII. CONCLUSION

In this paper we introduced the navigation focused, query-specific facet extraction model NAV in the context of Faceted Web Search. We analyzed its extrinsic utility in comparison

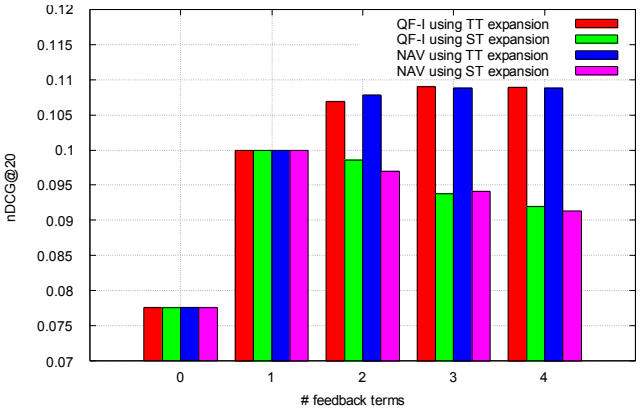

Figure 1. nDCG@20 results for multi term feedback using top-5 facets

to QF-I and discussed its computational benefits. Furthermore, we proposed a soft ranking facet feedback model that utilizes the extracted facets if BM25F is the baseline retrieval model. Finally, we provided a simplified algorithm of the repeat region patterns employed by Dou et al.

The conducted experiments show, that facets generated by NAV, compared to QF-I facets, provide at least the same extrinsic utility. This is especially relevant, as NAV reduces the computational effort significantly. Unfortunately, we find the previously proposed soft ranking expansion models not compatible with BM25F. However, our TT expansion model is able to reproduce the utility reported by Kong et al. We conclude that each baseline retrieval model might require its specific soft ranking expansion model. Finally, our meta pattern HTML extraction algorithm yields lists that improve facet extraction significantly.

REFERENCES

[1] C. L. Clarke, N. Craswell, I. Soboroff, and E. M. Voorhees, "Overview of the TREC 2011 Web Track," TREC, 2011.

[2] W. Dakka and P. G. Ipeirotis, "Automatic extraction of useful facet hierarchies from text databases," Proc. ICDE, 2008, pp. 466–475.

[3] Z. Dou, S. Hu, Y. Luo, R. Song, and J.-R. Wen, "Finding dimensions for queries," Proc. CIKM, 2011, pp. 1311-1320.

[4] C. Kohlschütter and W. Nejdl, "Using link analysis to identify aspects in Faceted Web Search," SIGIR Faceted Search Workshop, 2006, pp. 55-59.

[5] W. Kong and J. Allan, "Extracting query facets from search results," Proc. SIGIR, 2013, pp. 93-102.

[6] W. Kong and J. Allan, "Extending faceted search to the general web," Proc. CIKM, 2014, pp. 839–848.

[7] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia," Proc. WWW, 2010, pp. 651-660.

[8] D. Metzler and W. B. Croft, "A Markov random field model for term dependencies," Proc. SIGIR, 2005, pp. 472-479.

[9] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38 (11), pp. 39–41, 1995.

[10] S. Robertson, "The Probabilistic Relevance Framework: BM25 and beyond," Foundations and Trends in Information Retrieval, vol. 3 (4), pp. 333–389, 2010.

[11] P. Yee, K. Swearingen, K. Li, and M. Hearst, "Faceted metadata for image search and browsing," Proc. CHI, 2003, pp. 401–408.

[12] L. Zhang and Y. Zhang, "Interactive retrieval based on faceted feedback," Proc. SIGIR, 2010, pp. 363-370.