Revisiting Open Domain Query Facet Extraction and Generation

Chris Samarinas
University of Massachusetts Amherst
United States
csamarinas@cs.umass.edu

Arkin Dharawat
University of Massachusetts Amherst
United States
adharawat@cs.umass.edu

Hamed Zamani
University of Massachusetts Amherst
United States
zamani@cs.umass.edu

ABSTRACT

Web search queries can often be characterized by various facets. Extracting and generating query facets has various real-world applications, such as displaying facets to users in a search interface, search result diversification, clarifying question generation, and enabling exploratory search. In this work, we revisit the task of query facet extraction and generation and study various formulations of this task, including facet extraction as sequence labeling, facet generation as autoregressive text generation or extreme multi-label classification. We conduct extensive experiments and demonstrate that these approaches lead to complementary sets of facets. We also explored various aggregation approaches based on relevance and diversity to combine the facet sets produced by different formulations of the task. The approaches presented in this paper outperform state-of-the-art baselines in terms of both precision and recall. We confirm the quality of the proposed methods through manual annotation. Since there is no open-source software for facet extraction and generation, we release a toolkit named Faspect[1], that includes various model implementations for this task.

CCS CONCEPTS

• **Information systems** → **Information extraction**; **Query intent**; • **Computing methodologies** → **Natural language generation**.

ACM Reference Format:
Chris Samarinas, Arkin Dharawat, and Hamed Zamani. 2022. Revisiting Open Domain Query Facet Extraction and Generation. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3539813.3545138

1 INTRODUCTION

Search queries can often be characterized by multiple facets, which are implicit or explicit aspects of the query. Implicit facets are often called latent topics. Explicit facets, on the other hand, are words or phrases that represent query aspects. For example, given the query '*James Webb*', some facets can be '*James Webb satellite*', '*assembly of James Webb*', '*first images from James Webb telescope*', '*launch of James Webb*', and'*James E. Webb*'. In an open domain setting, facets are diverse and cannot be simply extracted using knowledge bases. The facets sometimes do not necessarily correspond to attributes, but they can be considered as query subtopics. For instance for the search query '*starting a business*', some facets would be '*loans*', '*requirements*', '*ideas*', '*tips*', '*cost*'. A plausible solution to facet extraction or generation is to exploit the top retrieved documents returned in response to the query [10, 14].

Query facet identification has many applications. The display of facets for a query can assist the user in refining and specifying the original query and in exploring various subtopics [19]. In a conversational search system, explicit facets can be used for asking clarifying questions [27–29]. Query facets, either implicit or explicit, can also be used for diversifying the search results [5].

Early methods for open domain query facet extraction focused on frequency of terms and phrases in the search engine result list [7, 23, 24]. There also exist some supervised approaches that score each phrase based on some co-occurrence features [13, 14]. Recently, Hashemi et al. [10] showed that neural models can be employed for effective generation of query facets.

In this paper, we revisit this task by providing various formulations of the task and exploring their effectiveness. First, we formulate query facet extraction as a *sequence labeling* task on the top retrieved documents. Second, we formulate query facet generation as an *autoregressive text generation* problem. Third, we cast it as an *extreme multi-label classification* task. Finally, we formulate it as a prompt-based (or conditioned) text generation from pre-trained large language models (i.e., GPT-3). We also complement these formulations by borrowing ideas from simple yet effective unsupervised facet extraction based on term and phrase frequency. We study all these formulations from various precision- and recall-oriented angles. We demonstrate that facet generation models are more effective than facet extraction models. Our analysis demonstrates that these formulations provide complementary information. Following this observation, we explore a number of approaches for aggregating the facets produced by different models. We show that a round-robin approach that diversifies the source of generated facets can lead to significant improvements in terms of recall.

The main contributions of this work include:

(1) Introduction of novel formulations for the facet extraction and generation task driven by the recent advancements in text understanding and generation.
(2) Through offline evaluation, we demonstrate that the models studied in this paper significantly outperform state-of-the-art baselines. We demonstrate that their combination leads to improvement in recall. The manual annotation of the results highlights the high quality of generated facets.
(3) Despite the importance of facet extraction and generation, to the best of our knowledge, there is no open-source toolkit for this task. Thus, another contribution of this work is an

[1]Faspect is available at https://github.com/algoprog/Faspect.

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545138

open-source toolkit, named Faspect, that includes various implementations of facet extraction and generation methods included in this paper. Such resource will smooth the path for researchers and practitioners to use facet extraction and generation in their research.[2]

2 RELATED WORK

The majority of the early work on facet extraction focused on methods that use various external resources. Dakka and Ipeirotis [4] extracted potential facet terms based on entity hierarchies in Wikipedia and WordNet. In a subsequent work, Li et al. [16] developed a faceted retrieval system that displays relevant facets from Wikipedia hyperlinks and categories. Stoica et al. [22] proposed a method that generates hierarchical faceted metadata from textual descriptions of items using hypernym relations in WordNet. Oren et al. [19] developed a faceted interface for semi-structured RDF data. Kohlschtter et al. [12] presented a facet extraction method based on personalized PageRank link analysis and annotated taxonomies. While methods using curated resources can work well in some cases, they cannot scale easily to many domains.

There also exist some unsupervised methods for extracting facets from unstructured and semi-structured text. Dou et al. [7] developed one of the first open domain facet extraction systems, named QDMiner, that discovers query dimensions by aggregating frequent lists within the top web search results using textual patterns. Wang et al. [23] extracted facets by clustering similar text fragments from the top retrieved documents. Based on the clusters and a language model trained from a query log, they scored each facet. Wei et al. [24] built a facet tree based on semantic relations between potential facet words. Deveaud et al. [5] used LDA on retrieved documents to identify latent (implicit) query concepts. Even though this method can be useful for retrieval, it fails to extract explicit facets, because every latent representation is not easily interpretable.

More recently, we have seen the introduction of some supervised methods for facet extraction. For example, Kong and Allan [13] proposed a probabilistic graphical model that learns the likelihood of each candidate term to be a facet term, in addition to the likelihood of two terms being grouped together in a query facet. In their subsequent work [14], the authors proposed a graphical model that optimizes the expected performance measure and uses a performance prediction model that selectively shows facets for some queries. Most recently, Hashemi et al. [10] proposed an encoder-decoder transformer-based model that can learn multiple intent representations for each search query. Thus, their method can be potentially used for facet generation.

Inspired by all the mentioned methods in this section, we study various formulations of extractive (supervised and unsupervised) and generative methods and their combination. Some of the strongest methods mentioned in this section are used as baselines in our experiments.

3 FACET EXTRACTION AND GENERATION

Problem Statement: In this paper, we focus on the extraction and generation of facets from the search engine result page (SERP) for a given query. Let $T = \{(q_1, D_1, F_1), (q_2, D_2, F_2), \cdots, (q_n, D_n, F_n)\}$ be a training set containing n triplets (q_i, D_i, F_i), where q_i is an open domain search query, $D_i = [d_{i1}, d_{i2}, \cdots, d_{ik}]$ denotes the top k documents returned by a retrieval model in response to query q_i, and $F_i = \{f_{i1}, f_{i2}, \cdots, f_{im}\}$ is a set of m ground truth facets associated with query q_i. Each facet f_{ij} may or may not appear in the top retrieved documents D_i. The task is to train a facet extraction, generation, or classification model M_θ parameterized by θ such that for any unseen query q with a result list D, the model $M_\theta(q, D)$ returns an accurate list of facets.

Overview: In the following, we describe five methods for facet extraction or generation. These formulations provide complementary information, thus theoretically we can benefit from aggregating their results. We empirically validate this in our experiments. The first formulation casts the facet extraction problem as a *sequence labeling* task, while the second formulation looks at the facet generation problem as *autoregressive text generation*. The third formulation treats the task as an *extreme multi-label classification* problem. The last formulation looks at all the documents in SERP and selects a number of facets based on a simple frequency-based approach. This results in a simple unsupervised yet effective solution.

3.1 Facet Extraction as Sequence Labeling

We can cast the facet extraction problem as a sequence labeling task, which has been successfully used for entity recognition [9], keyphrase extraction [21], and question answering [26]. To this aim, for every document d_{ij} in the result list returned in response to query q_i, we create a sequence labeling output based on the BIO tagging format. In more detail, we tokenize the document d_{ij} and assign a label `B`, `I`, or `O` to each token. For every token $w_x \in \text{tokenize}(d_{ij})$, we use the following labeling function:

$$y_{ijx} = \begin{cases} \texttt{B} & \text{if } w_x \text{ is the beginning token for a facet in } F_i. \\ \texttt{I} & \text{if } w_x \text{ is a facet token other than the beginning token.} \\ \texttt{O} & \text{otherwise.} \end{cases}$$

What exactly facets are?

Thus, the label `B` is always followed by a number of `I` labels for terms with more than one token. The reason we use three labels instead of two, is to be able to deal with edge cases where two different facets are consecutive words in a document. Note that the labels `B` and `I` are only used if a whole facet text from F_i is mentioned in d_{ij}. Therefore, our first component $M_{\theta_{\text{ext}}}$ classifies each document token to `B`, `I`, or `O`. We use RoBERTa [6, 17] for modeling $M_{\theta_{\text{ext}}}$ and apply an MLP with the output dimensionality of three to each token representation of BERT. We use `[CLS] query tokens [SEP] doc tokens [SEP]` as the BERT input and optimize the following likelihood objective:

$$\theta^*_{\text{ext}} = \arg\min_{\theta_{\text{ext}}} \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{1}{|d_{ij}|} \sum_{x=1}^{|d_{ij}|} -\log p(y_{ijx} | M_{\theta_{\text{ext}}}(q_i, d_{ij})_x) \quad (1)$$

where $\log p(y_{ij} | M_{\theta_{\text{ext}}}(q_i, d_{ij})) = \sum_{x=1}^{|d_{ij}|} \log p(y_{ijx} | M_{\theta_{\text{ext}}}(q_i, d_{ij})_x)$. The probability $p(y_{ijx} | M_{\theta_{\text{ext}}}(q_i, d_{ij})_x$ can be computed by applying a softmax operator to the model's output for the x^{th} token.

At inference, we get the model output for all the documents in D_i and sort them by frequency. This means that a facet that is generated multiple times from different documents in SERP gets a higher weight.

[2] Faspect is available at https://github.com/algoprog/Faspect.

3.1. Sequence labeling 3.3. multi-label classification 3.5. unsupervised
3.2. ad-hoc generate 3.4. prompt-based facet SERP generation.

3.2 Autoregressive Facet Generation

In the second formulation, we perform abstract facet generation using an autoregressive text generation model. To this aim, for every query q_i, we concatenate the facets in F_i using a separation token. Let y'_i denote this concatenation. We use BART [15], a Transformer-based encoder-decoder model for text generation. We train two variations of this generative model: (1) one variation only takes the query tokens and generates the facets, and (2) another variation that takes the query tokens and the document tokens for all documents in SERP (separated by [SEP]) as input and generates facet tokens one by one. Therefore, we use the following objective:

$$\theta^*_{\text{gen}} = \arg\min_{\theta_{\text{gen}}} \sum_{i=1}^{n} \frac{1}{|y'_i|} \sum_{x=1}^{|y'_i|} -\log p(y'_{ix}|v, y'_{i1}, \cdots, y'_{ix-1}) \quad (2)$$

where v is the BART encoder's output.

During inference, we perform autoregressive text generation with beam search and sampling, conditioning the probability of the next token on the previous generated tokens.

3.3 Facet Generation as Extreme Multi-Label Classification

In the last supervised formulation, we treat the facet generation task as an extreme multi-label text classification problem. The intuition behind this approach is that some facets tend to appear very frequently across different queries. For instance, the *'for sale'* facet can appear for multiple queries related to products, or the facet *'review'* can be relevant for various queries related to movies, books, games, etc. After identifying a set of frequent facet terms F, we can train an extreme multi-label classifier to estimate the probability of relevance of each individual term given a query and relevant documents. Before picking the most frequent facets, we apply some pre-processing, removing prepositions and query tokens from the beginning and end of the facet phrase. For example, the facet term *'world series of lacrosse 2018'* for the query *'world series 2018'*, would be normalized to *'lacrosse'*.

In this work, we use RoBERTa [6, 17] to model $M_{\theta_{\text{mcl}}}$, and get the probability of every facet by applying a linear transformation to the representation of the [CLS] token followed by sigmoid activation. We train the model by optimizing the binary cross-entropy objective:

$$\theta^*_{\text{mcl}} = \arg\min_{\theta_{\text{mcl}}} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_i - 1)\log(1 - y'_{i,j}) - y_i \log y'_{i,j} \quad (3)$$

where $y'_{i,j} = p(y_{ij}|M_{\theta_{\text{mcl}}}(q_i, D_i))$ is the probability of relevance of the facet $f_j \in F$ given the query q_i and the list of documents D_i, and it can be computed by applying a sigmoid operator to the model's output for the j^{th} facet class.

3.4 Facet Generation by Prompting Large Language Models

In this approach, we investigate the few-shot effectiveness of large-scale pre-trained autoregressive language models. We experiment with GPT-3 [1] and generate facets using a task description followed by a small number of examples as seen in Figure 1. Through prompting, we define the number of facets in the beginning of every example output, so that we can have control over the number of facets GPT-3 can generate.

```
Extract aspects for the given queries:

Query: fortnite game
Aspects (4): platform, characters, minimum pc requirements, streamers

Query: University of Massachusetts Amherst
Aspects (5): tuition, acceptance rate, location, admissions, ranking

Query: lungs
Aspects (3): clean, heal, strengthen

Query: Oculus Quest VR headset
Aspects (6): reviews, price, apps, where to buy, discount, models

Query: cancer
Aspects (4): treatment, types, research, definition

Query: tax return
Aspects (5): filing, deadline, online, status, refund
```

Figure 1: Prefix passed to GPT-3 for conditioning the facet generation. The input query is in green and the generated output in blue. The first five queries are provided to let GPT-3 for prompting.

most related

3.5 Unsupervised Facet Extraction from SERP

To complement the proposed approaches, we use a simple yet effective unsupervised method that extracts frequent ngrams from the returned documents D_i. We filter out the majority of ngrams based on a number of criteria. We skip ngrams that start or end with verbs, prepositions, determiners, or symbols. We also filter out ngrams that end with pronouns. For this purpose we use the part-of-speech tags given by the average perceptron tagger of the NLTK library [18].[3] We sort the remaining ngrams using the following scoring function:

$$s(q_i, f') = (1 + \alpha * \text{overlap}(q_i, f')) \times \text{freq}(f', D_i) \quad (4)$$

where f' is a facet candidate ngram for query q_i, the function $\text{overlap}(q_i, f')$ returns the percentage of words from the query appearing in f', $\text{freq}(f', D_i)$ computes the frequency of ngram f' in the top retrieved documents D_i, and α is a hyperparameter. The intuition behind the introduction of the query overlap in this scoring function is that many facets often appear as prefix or suffix of the query. For example, for the query '*cars*', some candidate ngrams could be '*cars for sale*' or '*used cars*' which are viable candidate facets for this query.

3.6 Facet Lists Aggregation

As we see in our experiments, these aforementioned methods provide complementary information and thus their aggregation is expected to lead to improvements, especially in terms of recall. We explored three aggregation methods: *Learning to Rank*, *MMR diversification* and *Round Robin Diversification*.

[3] https://www.nltk.org

Table 1: Evaluation of facet extraction and generation methods. The superscript * denotes statistically significant improvements compared to all the baselines in terms of two-tailed paired t-test with Bonferroni correction with 99% confidence.

Model	Term Overlap			Exact Match			Set BLEU				Set BERT-Score		
	Prec.	Recall	F1	Prec.	Recall	F1	1-gram	2-gram	3-gram	4-gram	Prec.	Recall	F1
QDist [25]	0.1275	0.1108	0.1121	0.0084	0.0103	0.0087	0.2042	0.1752	0.1578	0.1439	0.5185	0.5114	0.5108
QFI [14]	0.1525	0.1840	0.1606	0.0137	0.0162	0.0155	0.2141	0.1845	0.1597	0.1561	0.5113	0.5174	0.5156
QFJ [14]	0.1504	0.1853	0.1584	0.0143	0.0151	0.0144	0.2144	0.1879	0.1591	0.1540	0.5174	0.5208	0.5185
QDMiner [8]	0.1629	0.1879	0.1690	0.0207	0.0278	0.0223	0.2225	0.1970	0.1680	0.1605	0.5118	0.5170	0.5124
NMIR [10]	0.1856	0.1965	0.1905	0.0354	0.0388	0.0370	0.2524	0.2139	0.1906	0.1741	0.5311	0.5368	0.5344
Facet Generation (query+docs)	0.2014*	**0.3084***	**0.2361***	0.0417*	0.0655*	0.0496*	0.2972*	0.2339*	0.2030	**0.1855***	0.5361*	**0.5412***	**0.5382***
Facet Generation (query)	0.1816	0.2861*	0.2161*	0.0304*	0.0458*	0.0355*	**0.2993***	**0.2355***	**0.2032**	0.1852*	0.5319	0.5401*	0.5356
Sequence Labeling	**0.2075***	0.2424*	0.2131*	**0.0588***	**0.0878***	**0.0678***	0.2236	0.1637	0.1380	0.1260	0.5389*	0.5252	0.5314
Extreme Facet Classification	0.0608	0.0626	0.0594	0.0254	0.0373	0.0294	0.1227	0.0638	0.0386	0.0305	**0.5455***	0.5182	0.5310
Unsupervised Facet Extraction	0.1748	0.2465*	0.1971	0.0149	0.0287	0.0192	0.2915	0.2203	0.1851	0.1669	0.5260	0.5362	0.5307
GPT-3 (few-shot prompting)	0.0948	0.1396	0.1063	0.0249	0.0354	0.0283	0.1928	0.1185	0.0837	0.0724	0.5373*	0.5201	0.5280

Facet Relevance Ranking We use a bi-encoder model [11] to assign a score to each candidate facet for each query and re-rank them based on their score in descending order. We compute the facet relevance score using the dot product of the query and facet representations: $\text{sim}(q_i, f_i) = E(q_i) \cdot E(f_i)$. For the embedding function E, we use the average token embedding of BERT pre-trained on multiple text similarity tasks [20].[4] To find the optimal parameters for the embedding function E, we minimize the following cross-entropy loss for every positive query-facet pair (q_i, f_i^+) in the MIMICS dataset:

$$\mathcal{L}(q_i, f_i^+, \{f_{i,j}^-\}_{j=1}^{B-1}) = -\log \frac{e^{sim(q_i, f_i^+)}}{e^{sim(q_i, f_i^+)} + \sum_{j=1}^{B-1} e^{sim(q_i, f_{i,j}^-)}} \quad (5)$$

where B is the training batch size, and $\{f_{i,j}^-\}_{j=1}^{B-1}$ the set of in-batch negative examples.

MMR diversification: In the second aggregation method, we use a popular diversification approach, named *Maximal Marginal Relevance (MMR)* [2]. The intuition is that different models may generate redundant facets and a relevance ranking model looks at facet independently. Thus, it may select redundant facets and diversification can potentially resolve this issue. Our MMR approach selects facets one by one and scores each facet as:

$$\arg \max_{f_i \in R-S} \left[\lambda \text{sim}(q, f_i) - (1-\lambda) \max_{f_j \in S} \text{sim}(f_i, f_j) \right] \quad (6)$$

where R the list of extracted facets for the given query q, and S the set of already selected facets. λ is a hyper-parameters. For the similarity function, we use the same model used in relevance ranking (see above).

Round Robin Diversification: In the round-robin based approach, we iterate over the four lists of facets generated by different models, and alternatively select the facet with the highest score from each list until we generate the desired number of facets. This approach basically diversify the result list based on the facets generated by different models.

4 EXPERIMENTS

4.1 Dataset

Following Hashemi et al. [10], in our experiments, we used the MIMICS dataset [28].[5] MIMICS contains web search queries sampled from the Bing query logs, and for each query, it provides up to 5 facets and the returned result snippets. MIMICS consists of three subsets. We used the largest subset, MIMICS-Click, that contains over 400K queries, for training, and MIMICS-Manual, which contains 2832 queries, for evaluation. For the retrieved documents, MIMICS contains the list of web pages returned by the Bing's web search API. Similar to [10], we use the returned snippets as document text in our experiments.

4.2 Evaluation Metrics

To evaluate our approaches, we follow Hashemi et al. [10] and use four sets of metrics: (1) precision, recall, and F1 of *term overlap* between the produced and the ground truth facets, (2) precision, recall, and F1 of *exact match* between the produced and the ground truth facet sets, (3) the *set BLEU* and (4) the *set BERT-Score* between the mentioned sets. For the exact definition of metrics, we refer the reader to [10]. We perform evaluation on MIMICS-Manual, using the top five extracted facets from each model.

4.3 Experimental Setup

For the sequence labeling model, we fine-tuned RoBERTa-base, with maximum sequence length 400 tokens, using Adam optimizer, batch size 64 and initial learning rate 5×10^{-5} for 5 epochs. We experimented with 2 variations of the training set; one set that includes snippets with no mentioned facets as negative examples, and one that contains only snippets with at least one facet mention. Training the model using the additional negative examples, leads to higher precision, but when training without negatives, the model achieves significantly higher recall, which is the metric we want to maximize before the ranking step.

For the sequence generation models, we fine-tuned BART-base, with maximum sequence length 470 tokens, using Adam optimizer, batch size 32 and initial learning rate 5×10^{-5} for 5 epochs. For generating sequences, we used nucleus sampling, keeping the top

[4] Model weights from https://hf.co/sentence-transformers/all-mpnet-base-v2

[5] MIMICS is publicly available at https://github.com/microsoft/MIMICS.

Table 2: Examples of the top three predicted facets by each model.

Model	Query				
	sam houston state university	heart attack	how to write an essay	edd	planting grass seed
Facet Gen. (q + d)	bookstore, jobs, tuition	in men, in women, in children	expository essay, narrative essay, descriptive essay	social security, for unemployment, eddn social security assistance	outdoors, indoors, planting rye grass seed
Facet Gen. (query)	student portal, sam houston asu jobs, tuition	after drinking, after running, after walking	expository essay, informative essay, narrative essay	songs, in law, in russian	indoors, planting bermuda grass seed, outdoor
Seq. Labeling	education, campus, tour	causes, treatment, muscle	thesis, high school, persuasive	employment development department, development department, edd degree	bluegrass, green, flower
Classifier	weather, hotels, zip code	men, women, kids	facts, quotes, english	usa, insurance, facts	usa, canada, sale
Unsupervised	campus, wikipedia, houston state university sam	heart muscle, symptoms, mayo clinic	your essay, writing an essay, narrative essay	services the edd, english on the edd, edd degree	how to plant grass seed, when to plant grass seed, steps for planting grass seed
GPT-3	tuition, acceptance rate, location	symptoms, treatment, prevention	brainstorm, thesis, outline	eligibility, application process, benefits	best time, how to, when to
Ground truth	bookstore, jobs, tuition	aspirin, blood test, medical term	comparative, descriptive	educationis doctor or doctor, electronic data discovery, employment development department	temperature, tips, tools

not ideal for ... retrieval

tokens with accumulated probability 0.8 at every generation step, and with temperature 0.7.

For the multi-label facet classification model, we experimented with the following frequency thresholds for picking classes: {500, 1000, 2000}. Eventually, we kept the facets with minimum frequency 2000 after pre-processing, resulting in 787 classes. We fine-tuned RoBERTa-base using Adam optimizer, batch size 32 and initial learning rate 5×10^{-5} for 2 epochs. We experimented with various classification thresholds in {0.005, 0.01, 0.05, 0.1, 0.5} and found 0.05 to give the best metrics.

4.4 Results

Table 1 summarizes the performance of different facet extraction and generation approaches. Each model generates five facets per query, since there are up to five facets per query in the ground truth.[6] The facet generation and sequence labeling models generally perform better than the other methods including the baselines. The improvements over the baselines is statistically significant in nearly all cases. The sequence labeling approach beats facet generation models in terms of exact match metrics since it is a facet extraction model. However, Set BERT-Score that measures semantic similarities gives higher weight to facet generation models. Facet generation models generally perform well for term overlap and Set BLEU (a phrase level matching metric). The extreme multi-label facet classification model seems to have the lowest term overlap and Set BLEU performance, but this is expected because the model is trained to predict only the 787 most frequent facets observed in the training set. The GPT-3 model also performs poorly, but it should be noted that it is a few-shot learning model and it only sees five random examples from the training data as prompt. Due to limited resources, we could not fine-tune GPT-3 on the whole training set.

(In-context learning?)

In order to better understand the behavior of the supervised models, we perform a quantitative comparison of their outputs by calculating their overlap coefficient on token level. For two sets of facets A and B, the overlap coefficient is given by $(A \cap B)/\min(|A|, |B|))$. The results are presented in Figure 2. We discover that the output of the models varies significantly. The facets extracted from the sequence labeling model seem to have very low overlap with the ones generated by the facet generation models, with just 24-28% token level overlap. The two facet generation models, even though their only difference is the inclusion of documents in the input, still produce significantly different outputs, with only 59% overlap. This is an indication that many facets can only be generated using information from the top retrieved documents.

[6]Note that the results in Table 1 are different from the ones reported in [10]. We confirm that their results are on a held out data from the MIMICS-Click dataset. We report our results on MIMICS-Manual since it is manually annotated by trained experts, but we confirm that we also observe improvements on the same held out dataset used in [10]. We have used the same implementations as the authors of the original paper.

Table 3: Comparison of term overlap and exact match recall for the various extraction and generation models and their aggregation.

	Exact Match				Term Overlap			
Models	**R@5**	**R@10**	**R@20**	**R@30**	**R@5**	**R@10**	**R@20**	**R@30**
Facet Generation (query+docs)	0.0655	**0.1219**	0.1711	0.1711	**0.3084**	0.3768	0.4190	0.4214
Facet Generation (query)	0.0458	0.0854	0.1258	0.1293	0.2861	0.3413	0.3816	0.3842
Sequence Labeling	**0.0882**	0.1061	0.1061	-	0.2429	0.2817	0.2819	-
Extreme Facet Classification	0.0501	0.0812	0.1037	0.1093	0.1508	0.2185	0.2723	0.2920
Unsupervised Facet Extraction	0.0287	0.0351	0.0431	0.0508	0.2465	0.2793	0.3232	0.3528
GPT-3 (few-shot prompting)	0.0354	0.0473	0.0527	-	0.1396	0.1815	0.1954	-
Relevance Ranking	0.0326	0.0525	0.0931	0.1296	0.2797	0.3303	0.4135	0.4736
MMR Diversification	0.0306	0.0423	0.0669	0.0949	0.2840	0.3414	0.4196	0.4712
Round Robin Diversification	0.0646	0.1160	**0.1781**	**0.2195**	0.3047	**0.3867**	**0.4724**	**0.5230**

Table 4: Ablation study of aggregation methods. In each row, one facet extraction or generation model is removed from the aggregation input. The removal order is based on their term overlap F1 performance in Table 1.

	Exact Match				Term Overlap			
Models	**R@5**	**R@10**	**R@20**	**R@30**	**R@5**	**R@10**	**R@20**	**R@30**
All	0.0646	0.1160	0.1781	**0.2195**	0.3047	0.3867	**0.4724**	**0.5230**
- GPT-3 (few-shot prompting)	0.0972	0.1511	**0.1921**	0.1993	0.3406	0.4088	0.4697	0.4922
- Unsupervised Facet Extraction	**0.1029**	**0.1667**	0.1833	0.1836	0.3441	0.4158	0.4370	0.4370
- Extreme Multi-Label Classification	0.0994	0.1595	0.1716	0.1716	**0.3516**	**0.4177**	0.4320	0.4320
- Sequence Labeling	0.0780	0.0866	0.0866	0.0866	0.3309	0.3411	0.3411	0.3411
- Facet Generation (query)	0.0655	0.1219	0.1711	0.1711	0.3091	0.3091	0.3091	0.3091

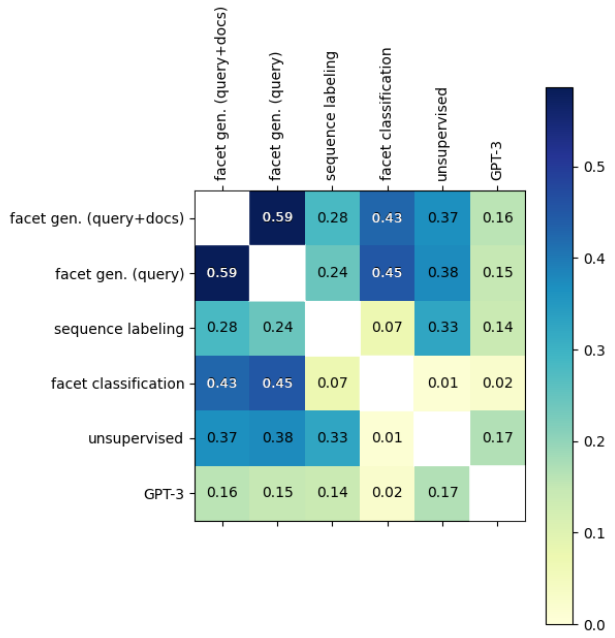

Figure 2: Average overlap coefficient between the outputs of the various supervised models on token level.

Recall-Focused Models. Given Figure 2, we know that the models produce different facets, thus we evaluate three facet aggregation methods mentioned in Section 3.6 to improve recall (see the last three rows in Table 3). Interestingly, we observe that the most effective method that maximizes recall is the round robin diversification approach. For a deeper understanding of models' performance in terms of term overlap recall, we report their performances for varying number of extracted facets in Table 3. We observe that the maximum potential recall increases by a big margin when combining results using round robin diversification. Note that in Table 3 there are no reported metrics for Recall@30 for the Sequence Labeling and GPT-3 models. The reason is that the Sequence Labeling model was not able to extract 30 or more facets for any query in the evaluation set, and GPT-3 was guided to output only up to 15 facets.

Ablation Study. We performed an ablation study for the facet aggregation method by removing methods one by one from the aggregation set. The aggregation method is the round-robin approach which is the best performing model. The removal order is decided based on the term overlap F1 in Table 1. The ablation study results are reported in Table 4. It has been shown that inclusion of all models in facet aggregation, including the poor performing few-shot learning and unsupervised approaches, leads to recall improvement for deeper ranking cut-offs. However, the poor performing models have negative impacts on Recall@5 and Recall@10.

Case Study. Finally, Table 2 reports some examples of facets generated by the six models. We can see that in many cases, some

facets can only be extracted by one of the models, confirming our observation of complementary information across different models.

Human Evaluation. In order to have a better understanding of how well the various models work in a real-world setting, we did some human evaluation. Evaluation on MIMICS-Manual does not reflect well the actual performance, because the list of annotated ground truth facets for each query is not very comprehensive, making it difficult to get a good estimate of precision and recall. We sampled randomly 50 queries from MIMICS-Manual, and followed the standard pooling approach for annotating the results of all the models. For each facet, we asked three expert annotators (not the authors of this work) to annotate the facets as Relevant (label 2), Partially Relevant (label 1), and Irrelevant (label 0). Partially relevant are the facets which are relevant but not extracted properly (e.g. they have grammatical mistakes, they end with preposition, etc). The annotators agreement for Relevant, Partially Relevant, and Irrelevant labels are 67.97%, 78.29%, and 84.39%, respectively. This gives us an overall annotation agreement of 64.72%. In case of disagreement, we use majority voting.

We report the following metrics: normalized cumulative gain (nCG) and normalized discounted cumulative gain (nDCG). The reason for the inclusion of nCG is that it does not have an order discount and it is a recall-oriented metrics and has been used in TREC Deep Learning Tracks [3]. The results from the human evaluation can be seen in table 5. Note that the purpose of this experiment is to confirm the quality of the results produced by our methods through human annotation. We observe that the performance of the models is quite high. The facet generation model that uses query and documents still shows superior performance. GPT-3 is very close to the best supervised model in terms of nCG and nDCG. The facet classification model seems to be very selective.

Based on these results, we conclude that relying on automatic evaluation for the facet extraction task can lead to misleading results when trying to estimate the real-world performance of a model. Curated datasets such as MIMICS can only be used for relative comparison of models due to their incomplete annotations.

Table 5: Human evaluation of the proposed methods for 50 random queries sampled from the MIMICS-Manual dataset.

Model	nCG	nDCG
Facet Generation (query+docs)	**0.8786**	**0.8759**
Facet Generation (query)	0.7840	0.7893
Sequence Labeling	0.7513	0.7499
Extreme Facet Classification	0.3893	0.3889
Unsupervised Facet Extraction	0.7260	0.7211
GPT-3 (few-shot prompting)	0.8780	0.8729

5 CONCLUSIONS

In this paper, we presented and analyzed multiple formulations for extraction and generation of explicit facet terms from search results. We showed quantitatively that the different formulations lead to complementary facet sets. We also studied various facet aggregation methods and demonstrated that a round-robin diversification approach would lead to significant recall improvements. Our models outperform the previous state-of-the-art model [10]. We released an open-source toolkit, named Faspect, for facet extraction and generation that includes all formulations studied in this paper.

6 FUTURE WORK

In this work, we demonstrated that extracting and aggregating facets from multiple models can improve facet recall significantly in the MIMICS dataset. It remains an open question how a single end-to-end model could achieve comparable performance. Multi-task learning could be a potential direction towards this goal.

We described three methods for ranking a list of extracted facets. However, the supervised models, relevance ranking and MMR, did not seem to improve the ranking of the extracted facets, even though they demonstrated good performance when ranking the ground truth facets among other randomly sampled facets. Further analysis is required to determine whether the reason behind the poor performance could be the high percentage of relevant extracted facets that do not appear in the ground truth.

In the future, we intend to extend this work by employing the studied facet generation models for clarifying question generation and search result diversification.

7 ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by Amazon through an Alexa Prize Grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). https://arxiv.org/abs/2005.14165

[2] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 335–336. https://doi.org/10.1145/290941.291025

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *TREC*.

[4] Wisam Dakka and Panagiotis G. Ipeirotis. 2008. Automatic Extraction of Useful Facet Hierarchies from Text Databases. *2008 IEEE 24th International Conference on Data Engineering* (2008), 466–475.

[5] Romain Deveaud, Eric SanJuan, and Patrice Bellot. 2014. Accurate and effective latent concept modeling for ad hoc information retrieval. *Document Numérique* 17 (2014), 61–84.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Zhicheng Dou, Sha Hu, Yulong Luo, Ruihua Song, and Ji-Rong Wen. 2011. Finding dimensions for queries. In *CIKM '11*.

[8] Zhicheng Dou, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song. 2016. Automatically Mining Facets for Queries from Their Search Results. *IEEE Transactions on Knowledge and Data Engineering* 28, 2 (2016), 385–397. https://doi.org/10.1109/TKDE.2015.2475735

[9] Kai Hakala and Sampo Pyysalo. 2019. Biomedical Named Entity Recognition with Multilingual BERT. In *EMNLP*.

[10] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[11] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Gregory S. Corrado, László Lukács, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart Reply: Automated Response Suggestion for Email. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[12] Christian Kohlschtter, de, Paul-Alexandru Chirita, and Wolfgang Nejdl. 2006. Prototype Demonstration: Using Link Analysis to Identify Aspects in Faceted Web Search.

[13] Weize Kong and James Allan. 2013. Extracting Query Facets from Search Results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland) *(SIGIR '13)*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/2484028.2484097

[14] Weize Kong and James Allan. 2016. Precision-Oriented Query Facet Extraction. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016).

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *CoRR* abs/1910.13461 (2019). arXiv:1910.13461 http://arxiv.org/abs/1910.13461

[16] Chengkai Li, Ning Yan, Senjuti Basu Roy, Lekhendro Lisham, and Gautam Das. 2010. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In *WWW '10*.

[17] 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* abs/1907.11692 (2019).

[18] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1* (Philadelphia, Pennsylvania) *(ETMTNLP '02)*. Association for Computational Linguistics, USA, 63–70. https://doi.org/10.3115/1118108.1118117

[19] Eyal Oren, Renaud Delbru, and S. Decker. 2006. Extending Faceted Navigation for RDF Data. In *SEMWEB*.

[20] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[21] Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase Extraction as Sequence Labeling Using Contextualized Embeddings. In *Advances in Information Retrieval*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer International Publishing, Cham, 328–335.

[22] Emilia Stoica, Marti A. Hearst, and Megan Richardson. 2007. Automating Creation of Hierarchical Faceted Metadata Structures. In *NAACL*.

[23] Qinglei Wang, Ya nan Qian, Ruihua Song, Zhicheng Dou, Fan Zhang, Tetsuya Sakai, and Qinghua Zheng. 2013. Mining subtopics from text fragments for a web query. *Information Retrieval* 16 (2013), 484–503.

[24] Xiao Wei, Xiangfeng Luo, and Qing Li. 2012. Automatic Facet Extraction Based on Multidimensional Semantic Index. *2012 Eighth International Conference on Semantics, Knowledge and Grids* (2012), 64–71.

[25] Xiaobing Xue and W. Bruce Croft. 2013. Modeling Reformulation Using Query Distributions. *ACM Trans. Inf. Syst.* 31, 2, Article 6 (may 2013), 34 pages. https://doi.org/10.1145/2457465.2457466

[26] Wonjin Yoon, Richard Jackson, Jaewoo Kang, and Aron Lagerberg. 2021. Sequence tagging for biomedical extractive question answering. *arXiv preprint arXiv:2104.07535* (2021).

[27] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. *Generating Clarifying Questions for Information Retrieval.* Association for Computing Machinery, New York, NY, USA, 418–428. https://doi.org/10.1145/3366423.3380126

[28] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. MIMICS: A Large-Scale Data Collection for Search Clarification. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020).

[29] Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational Information Seeking. In *arxiv*.